Dpto. Matemáticas, Estadística y Computación

Universidad de Cantabria

TESIS DOCTORAL

Aplicaciones Estadísticas de las Proyecciones Aleatorias

Presentada por Alicia Nieto Reyes para optar al grado de doctora por la Universidad de Cantabria

Dirigida por D. Juan Antonio Cuesta Albertos

Febrero 2010

Title in English:

Statistical Applications of Random Projections

With the exception of the cover, the acknowledgments and a brief summary, this thesis is written in English in order to achieve the "Doctorado Europeo".

Agradecimientos

Esta tesis nunca habria podido ser realizada sin la paciencia de dos personas, Juan y Aitor. Al primero le estoy muy agradecido por estar siempre en su despacho y nunca poner una mala cara a mis continuas interrupciones. ¡No se como todavía sigue en pie el flexo de su mesa! También por sus buenas ideas, lo correcto de su persona y haber sabido ponerse en mi situación en muchas ocasiones. Al segundo porque, como él ya sabe, se ha ganado el cielo y "man lernt nie aus!"

Así mismo me gustaría agradecer a Fabrice Gamboa porque sin él no se habría realizado el test de Gaussianidad que constituye el Capítulo 6. Además, junto a Regina Liu y a Bernard Bercu, me gustaría agradecerles por lo agradable que su hospitalidad hizo las estancias que he realizado. Sin ellas este doctorado no habría sido lo mismo. También querría agradecer a la gente de Valladolid por los buenos ratos pasados cuando hemos coincidido.

La realidad es que tanto en lo personal como académico, todo lo vivido me ha hecho llegar hasta aqui. Por ello me gustaria agradecer a Marilux porque me hizo decidir estudiar matemáticas, a Jesús Gago por sugerir un segundo año Erasmus y probar con la Universidad de Cantabria y a Nick por estar ahí en la época inglesa. Como toda tesis, ésta ha implicado largas jornadas en el despacho, por el tiempo pasado en él me gustaría agradecer a Domingo, Claudiu, Mar y a Gema y por los descansos a la gente del café. Finalmente, a mis padres y hermanas por su paciencia con que sea una "descastá".

Contents

List of Tables

List of Figures

Chapter 1

Resumen en castellano

Dado un conjunto de datos, o una distribución, en un espacio de dimensión mayor a uno, las proyecciones aleatorias consisten en proyectar los datos, o calcular la marginal de la distribución, en un subespacio de menor dimensión que ha sido elegido de forma aleatoria. En el caso en que el subespacio en el que proyectamos tenga dimensión uno, la llamamos proyección aleatoria unidimensional.

En el libro Vempala [81] está escrito *"Random projection is useful in many settings. (...) A natural setting is when the input data is in high-dimensional space, and it is possible to preserve essential properties for the data (for the particular problem at hand) while reducing dimensionality."* Este libro contiene aplicaciones del Lema de Johnson y Lindenstrauss, Johnson y Lindenstrauss [45]. Dicho lema asegura que las proyecciones aleatorias aproximadamente preservan las distancias con un grado alto de probabilidad si el subespacio en el que proyectamos ha sido elegido con la distribución uniforme. Este resultado está extendido a la distribución gaussiana estándar en Frankl y Maehara [32].

Sin embargo, nuestro interés en las proyecciones aleatorias viene de otra propiedad que éstas preservan: la distribución. En Cuesta-Albertos et al. [15] se demuestra que una proyección aleatoria unidimensional basta para distinguir entre dos distribuciones siempre

y cuando se encuentren en un espacio de Hilbert separable y que los momentos de una de ellas satisfagan una condición determinada. Esto es, dadas dos distribuciones y una marginal aleatoria unidimensional de éstas, tenemos que casi seguro las distribuciones son diferentes/iguales si y sólo si las marginales son diferentes/iguales. Una extensión de este resultado a espacios de Banach se encuentra en Cuevas y Fraiman [24].

Esta propiedad hace que las proyecciones aleatorias sean una herramienta importante en la estadística multidimensional y funcional, ya que las proyecciones aleatorias nos permiten reducir la dimensión a uno, donde podemos aplicar técnicas unidimensionales, a la vez que obtenemos una conclusión que es válida en el espacio de partida. Es decir, en vez de aplicar una técnica determinada en un espacio de dimensión mayor que uno podemos hacer una proyección y aplicar la técnica en dimensión uno y si se cumplen determinadas condiciones de regularidad, Cuesta-Albertos et al. [15] nos permite inferir conclusiones en el espacio de partida. Por lo tanto, esta herramienta nos da facilidad debido a que la técnica en un espacio de dimensión mayor que uno es, en caso de que exista, más complicada que su homóloga unidimensional.

Podríamos pensar que esta manera de actuar se basa en una idea algo radical: es suficiente con sólo una proyección aleatoria. Pero, por un lado, realmente esto no es algo tan nuevo puesto que en el prólogo de Vempala [81], C. H. Papadimitriou escribe *"This book is about the radical idea that even a random projection is often useful."* De todas formas, a lo largo de esta tesis se muestra como a veces, en la práctica, puede ser útil tomar más de una proyección aleatoria. Por otro lado, hemos dicho que es suficiente con proyectar en un espacio de dimensión uno y aplicar ahí las técnicas unidimensionales. Obviamente, la sustitución de cada uno de los datos por un número real (los datos proyectados) es un proceso que implica pérdida de información. Sin embargo, en Hand [39] podemos leer *"...simple methods typically yield performance almost as good as more sophisticated methods to the extent that the difference in performance may be swamped by other sources*

of uncertainty...” El trabajo de Hand está relacionada con técnicas de clasificación, pero esta idea podemos aplicarla también aquí en el sentido de que la pérdida de información que sufrimos no es tan relevante cuando se une a otros problemas que suelen aparecer en los datos reales.

Merece la pena mencionar que Cuesta-Albertos et al. [15] ha creado cierto interés en las proyecciones aleatorias a pesar de ser un artículo relativamente reciente. Por ejemplo, algunas aplicaciones estadísticas en las que ha sido utilizado son los siguientes:

- Análisis de la varianza de varias vías para datos funcionales, Cuesta-Albertos y Febrero-Bande [13].
- Identificación no paramétrica de la distribución de heterogeneidad en modelos económicos, Fox y Gandhi [30].
- Tests no parametricos, Cuesta-Albertos et al. [11].
- Profundidad y estadística dual, Cuevas y Fraiman [24].
- Detección de valores atípicos en datos funcionales, Febrero-Bande et al. [27].
- Estimación robusta y clasificación de datos funcionales, Cuevas et al. [23].
- Tests de bondad de ajuste, Cuesta-Albertos et al. [12, 16]. Además, el test propuesto en Cuesta-Albertos et al. [16] se utiliza en Opazo et al. [68]. Por otro lado, en Bugni et al. [10], los autores toman como referencia Cuesta-Albertos et al. [12] para comparar el test de ajuste que proponen.
- Finalmente en Cuesta-Albertos et al. [14] se clasifican datos de sonoridad del habla y para ello se utiliza un test de Kolmogorov-Smirnov para datos funcionales que está desarrollado en Cuesta-Albertos et al. [15].

En esta tesis trabajamos con proyecciones aleatorias unidimensionales. Por lo tanto, cuando hablemos en lo que sigue de proyecciones aleatorias estaremos refiriéndonos a

proyecciones aleatorias unidimensionales, a no ser que digamos lo contrario.

En esta memoria presentamos dos nuevas aplicaciones de las proyecciones aleatorias. La primera es una definición nueva de profundidad que, además, es una aproximación a la conocida profundidad de Tukey, Tukey [80], y la segunda es un test de gaussianidad para procesos estrictamente estacionarios. En lo que sigue vamos a describir estas dos aplicaciones y hacer un resumen, sin demostraciones, de los resultados que hemos obtenido en esta tesis.

Profundidad de Tukey aleatoria.

Esta aplicación consiste en definir una profundidad multidimensional que es conceptualmente simple y fácil de calcular y que puede ser aplicada a datos funcionales proporcionando resultados comparables a los obtenidos con profundidades más complicadas.

El objetivo de las profundidades es ordenar un conjunto dado de forma que si un dato se mueve hacia el centro de la nube de datos, su profundidad aumenta, y si el dato se mueve hacia el exterior, su profundidad disminuye. Más general, dada una distribución de probabilidad P definida en un espacio multidimensional (o incluso infinitodimensional) $\mathcal{X}$, una profundidad trata de ordenar los puntos de $\mathcal{X}$ desde el "centro (de P)" a la parte "exterior (de P)". Obviamente, este problema incluye conjuntos de datos si tenemos en cuenta que P puede ser la distribución empírica asociada al conjunto de datos. Así que en lo que sigue nos referiremos a la profundidad asociada a una distribución de probabilidad P.

En el caso unidimensional es razonable ordenar los puntos de $\mathbb{R}$ utilizando el orden inducido por la función

$$x \to D_1(x, P) := \min\{P(-\infty, x], P[x, \infty)\}. \tag{1.1}$$

De esta forma los puntos están ordenados siguiendo el orden decreciente de los valores absolutos de la diferencia entre sus percentiles y 50, y el punto más profundo es la mediana de P.

En el caso multidimensional no existe una forma tan evidente de ordenar los puntos. Por ello, se han definido diferentes profundidades multidimensionales (véase, por ejemplo, Liu et al. [53, 54]). Una de ellas, en la cual estamos particularmente interesados, es la *profundidad de Tukey (o del semiespacio)*, Tukey [80]. En Zuo y Serfling [83] se muestra que esta profundidad se comporta muy bien en comparación con varios competidores. La profundidad de Tukey de un punto $x \in \mathbb{R}^p$ con respecto a una probabilidad P, $D_T(x, P)$, es la probabilidad mínima que puede alcanzarse en los semiespacios cerrados que contienen a x. De forma más precisa

$$D_T(x, P) = \inf\{D_1(\Pi_v(x), P \circ \Pi_v^{-1}) : v \in \mathbb{R}^p\}, \ x \in \mathbb{R}^p, \tag{1.2}$$

donde Π_v denota la proyección de $\mathbb{R}^p$ en el subespacio unidimensional generado por v y $P \circ \Pi_v^{-1}$ denota la marginal de P en ese subespacio. Se han propuesto otras definiciones de profundidad basadas en la consideración de todas las posibles proyecciones unidimensionales (véase, por ejemplo, Zuo [84]). Consideramos que lo que sigue se puede aplicar a todas ellas pero hemos elegido la profundidad de Tukey para fijar ideas.

El mayor inconveniente de la profundidad de Tukey es que requiere mucho tiempo de computación, debido a que necesita del cálculo de todas las profundidades unidimensionales. Este tiempo computacional es más o menos razonable si $p = 2$, pero se vuelve prohibitivo incluso para $p = 8$, véase Mosler y Hoberg [65, pág. 54]. Ha habido algunas propuestas para reducirlo. Por ejemplo, en Zuo [85, pág. 2234] se sugiere tomar proyecciones seleccionadas de forma aleatoria. Por otro lado, en Cuevas et al. [23] se define una profundidad aleatoria consistente en, dado un punto $x \in \mathbb{R}^p$, elegir un número finito de vectores $v_1, ..., v_k \in \mathbb{R}^p$ de forma aleatoria y tomar como profundidad de x el promedio de $D_1(\Pi_{v_i}(x), P \circ \Pi_{v_i}^{-1})$, $i = 1, ..., k$. Nuestra idea se asemeja más a la sugerencia hecha en Zuo

[85]: sustituimos el ínfimo en 1.2 por un mínimo sobre un número finito de proyecciones aleatorias. De esta forma obtenemos una aproximación a la profundidad de Tukey que llamamos *profundidad de Tukey aleatoria.* Definámosla formalmente.

Definition 1.0.1. *Sean P y ν distribuciones de Borel en $\mathbb{R}^P$ absolutamente continuas y $v_1, ..., v_k$ vectores aleatorios independientes e idénticamente distribuidos con distribución ν. La profundidad de Tukey aleatoria de $x \in \mathbb{R}^p$ con respecto a P basada en k vectores aleatorios elegidos con ν es*

$$D_{T,k,\nu}(x, P) := \min\{D_1(\Pi_{v_i}(x), P \circ \Pi_{v_i}^{-1}) : i = 1, ..., k\}, \;\; x \in \mathbb{R}^p.$$

En el Capítulo 4 de esta tesis se estudian las propiedades de la profundidad de Tukey aleatoria. Una propiedad importante es que puede ser extendida a espacios funcionales. Por lo tanto, denotemos por $\mathcal{X}$ indistintamente a $\mathbb{R}^p$ o a un espacio de Hilbert separable y por $\mathcal{P}$ la clase de distribuciones de Borel en $\mathcal{X}$. A continuación se describen los cuatro resultados más importantes de dicho capítulo. En el primero se analiza si la profundidad de Tukey aleatoria satisface la definición de profundidad estadística propuesta en Liu [52] y formalizada en Zuo y Serfling [83]. En dichos articulos se habla de profundidades multidimensionales, sin embargo nosotros escribimos aqui dicha definición en términos de $\mathcal{X}$. En esta definición, si X es un vector aleatorio, P_X denota su distribución.

Definition 1.0.2. *La función acotada y no negativa $D(\cdot, \cdot) : \mathcal{X} \times \mathcal{P} \longrightarrow \mathbb{R}$ es una profundidad estadística si se satisfacen las siguientes propiedades:*

1. *$D(x + b, P_{X+b}) = D(x, P_X)$, para cualquier vector aleatorio X con valores en $\mathcal{X}$ y cualquier $b \in \mathcal{X}$.*

2. *$D(\theta, P) = \sup_{x \in \mathcal{X}} D(x, P)$ para cualquier $P \in \mathcal{P}$ que tenga centro de simetría en θ.*

3. *Para cualquier $P \in \mathcal{P}$ con punto más profundo θ y cualquier $\alpha \in [0, 1]$, $D(x, P) \leq D(\theta + \alpha(x - \theta), P)$.*

4. *$D(x, P) \to 0$ cuando $\|x\| \to \infty$, para cada $P \in \mathcal{P}$.*

Con respecto al punto *1.* de la Definición 1.0.2, es obvio que

$$D_{T,k,A\nu}(Ax, P_{AX}) = D_{T,k,\nu}(x, P_X),$$

pero no es difícil encontrar ejemplos tal que $D_{T,k,\nu}(Ax, P_{AX}) \neq D_{T,k,\nu}(x, P_X)$, ver Nota 4.1.3.1. El punto *2.* de la definición recoge la idea de que si una distribución tiene un único centro de simetria (con respecto a alguna noción de simetria) la profundidad debe alcanzar su máximo en este centro. Existen diferentes nociones de simetría, como la central, la angular y la del semiespacio. Como la simetría central, implica la angular que a su vez implica la del semiespacio, identificaremos el centro P con el centro de la simetría del semiespacio. Con respecto al punto *4.*, en la Sección 4.1.2 se presenta un contraejemplo para el caso funcional.

Theorem 1.0.3. *La profundidad de Tukey aleatoria es acotada, no-negativa y satisface* $D_{T,k,\nu}(x + b, P_{X+b}) = D_{T,k,\nu}(x, P_X)$, *para cualquier vector aleatorio* X *con valores en* $\mathcal{X}$ *y cualquier* $b \in \mathcal{X}$, *al igual que los puntos 2. y 3. de la Definición 1.0.2.*

Además en el caso en que $\mathcal{X}$ *sea* $\mathbb{R}^p$, *si* $\|x\| \to \infty$ *con* $x \in \mathbb{R}^p$, $P \in \mathcal{P}$ *y* $k > 0$, *entonces* $D_{T,k,\nu}(x, P)$ *converge a cero en probabilidad.*

Nótese que la aleatoriedad sólo afecta al punto *4.*. Esto sucede porque podría ocurrir que los k vectores estuviesen en el mismo hiperplano. En este caso el punto *4.* no se satisfacería para las sucesiones de puntos ortogonales al hiperplano, con norma tendiendo al infinito, si $D_{T,k,\nu}(0, P) > 0$.

En el siguiente resultado se demuestra que la versión muestral de la profundidad de Tukey aleatoria converge a la correspondiente poblacional.

Theorem 1.0.4. *Sean* $v_1, ..., v_k \in \mathcal{X}$, $P \in \mathcal{P}$ *y* $\{P_n\}$ *una sucesión de distribuciones empíricas calculadas en una muestra aleatoria de* P *que es independiente de los vectores* $v_1, ..., v_k$.

Entonces, condicionalmente a $v_1, ..., v_k$, *tenemos*

$$\sup_{x \in \mathbb{R}^p} |D_{T,k,\nu}(x, P_n) - D_{T,k,\nu}(x, P)| \to 0, \quad \textit{casi seguro [P].}$$

Es curioso que, a pesar del gran interés que hay en torno a las profundidades en general, y a la profundidad de Tukey en particular, no tenemos conocimiento de muchos resultados que prueben que una profundidad determina su distribución correspondiente. De hecho, con respecto a la profundidad de Tukey sólo conocemos un resultado, Koshevoy [47], donde se prueba que si P y Q son dos distribuciones en $\mathbb{R}^p$ teniendo ambas soporte finito y sus profundidades de Tukey coinciden, entonces $P = Q$. Una demostración alternativa del resultado de Koshevoy se puede ver en Hassairi y Regaieg [41]. En la última sección del Capítulo 4 se demuestra que la profundidad de Tukey aleatoria caracteriza las distribuciones discretas. En este resultado $\mathbb{S}^{p-1}$ representa la esfera unidad en $\mathbb{R}^p$.

Theorem 1.0.5. *Sean P y Q dos medidas de probabilidad tal que el soporte de P es a lo sumo numerable. Sean $v_1, ..., v_k \in \mathbb{S}^{p-1}$ vectores aleatorios idénticamente distribuidos con distribución ν, absolutamente continua con respecto a la medida geométrica en $\mathbb{S}^{p-1}$, definida en el espacio probabilístico (Ω, σ, κ). Sea*

$$\Omega_0 := \{\omega \in \Omega : D_{T,k,\nu}(x, P) = D_{T,k,\nu}(x, Q), \quad \textit{para todo } x \in \mathbb{R}^p\}.$$

Entonces $\kappa(\Omega_0) \in \{0, 1\}$, y $\kappa(\Omega_0) = 1$ si y sólo si $P = Q$.

El Capítulo 4 finaliza con una generalización del resultado principal en Koshevoy [47], que se incluye a continuación.

Theorem 1.0.6. *Sean P y Q dos medidas de probabilidad tal que P es discreta y para cualquier $x \in \mathbb{R}^p$, $D_T(x, P) = D_T(x, Q)$. Entonces $P = Q$.*

Este resultado generaliza al de Koshevoy porque aqui se necesita que sólo una de las dos probabilidades sea discreta (el soporte puede ser numerable) y en Koshevoy se necesita que las dos probabilidades tengan soporte finito.

El Capítulo 5 contiene algunas aplicaciones de la profundidad de Tukey aleatoria. Comienza analizando el número de proyecciones aleatorias necesarias para que la profundidad de Tukey aleatoria sea una buena aproximación de la profundidad de Tukey. En

principio este número depende del tipo de aplicación de la profundidad en la que estamos interesados, así como de la dimensión del espacio subyacente y del tamaño muestral que estamos utilizando. Sin embargo, las simulaciones llevadas a cabo en la Sección 5.1 sugieren que un máximo de 250 proyecciones aleatorias son suficientes para satisfacer una amplia gama de casos. Esta sección termina con una comparación entre el tiempo necesario para calcular la profundidad de Tukey aleatoria y el requerido para la de Mahalanobis, que de acuerdo con la Tabla 1 de Mosler y Hoberg [65] es una de las profundidades más rápidas de calcular.

Con respecto a las aplicaciones, en la Sección 5.2 tenemos una aplicación de la profundidad de Tukey aleatoria multidimensional consistente en reproducir el estudio de simulación realizado en Liu y Singh [57], donde los autores aplican medidas de profundidad para construir un test de las diferencias en homogeneidad entre distribuciones. Nuestro objetivo principal con esta aplicaión es mostrar que la profundidad de Tukey aleatoria proporciona resultados similares a los obtenidos en la práctica con la profundidad de Tukey. Usamos el bootstrap para elegir el número de proyecciones necesarias para calcular la profundidad de Tukey aleatoria.

Como hemos dicho antes, la profundidad de Tukey aleatoria puede ser extendida a espacios funcionales, a pesar de que no satisface todas las propiedades en la definición de profundidad estadística. En la Sección 5.3 presentamos una aplicación de la profundidad de Tukey aleatoria en espacios funcionales que consiste en un problema bien conocido de clasificación supervisada donde se clasifica un individuo como masculino o femenino según su curva de crecimiento.

En esta sección comenzamos por comparar los resultados obtenidos con la profundidad de Tukey aleatoria con los que se obtienen con las profundidades propuestas en López-Pintado y Romo [61]. Para ello, repetimos el estudio realizado en López-Pintado y Romo [60] reemplazando la profundidad utilizada por estos autores por la profundidad de

Tukey aleatoria. A continuación, analizamos las mejoras en clasificación que se obtienen al aplicar los métodos propuestos en Cuevas et al. [23] y Li et al. [49]. Estos resultados se comparan con métodos de clasificación basados en los k vecinos más próximos (k-NN) y en núcleos. Finalmente, como los datos son sólo 31-dimensionales, comparamos también con el método de "random forest". Los resultados aparecen en las Tablas 5.8, 5.9 y 5.10.

En esta sección se utiliza la validación cruzada para elegir el número de proyecciones.

Test de gaussianidad para procesos estacionarios.

En una amplia gama de situaciones en Estadística, se trata con datos consistentes en una serie temporal real, esto es, en una serie de observaciones reales $X_1, \ldots, X_n$ que están secuenciadas en el tiempo o espacio. Un modelo común es asumir que estas observaciones son una realización de un proceso estacionario de segundo orden $\mathbf{X} := (X_t)_{t\in\mathbb{Z}}$ (véase, por ejemplo, Hannan [40] y Gershenfeld [36]). Esto significa que la variable aleatoria X_t es, para cualquier $t \in \mathbb{Z}$, de cuadrado integrable y que la media y la matriz de covarianzas del proceso son invariantes por cualquier translación en el tiempo. Es decir, para cualquier $t, s \in \mathbb{Z}$, $\mathbb{E}(X_t)$ no depende de t y $\mathbb{E}(X_t X_s)$ sólo depende de la distancia entre t y s.

Un marco todavía más popular que el estacionario de segundo orden, es el caso gaussiano, donde adicionalmente se asume que todas las distribuciones marginales finito-dimensionales del proceso $(X_t)_{t\in\mathbb{Z}}$ son gaussianas. En este caso, como la distribución multidimensional gaussiana sólo depende de los momentos de orden uno y dos, el proceso también es estrictamente estacionario; lo que significa que todas las distribuciones marginales de dimensión finita del proceso son invariantes en ley si se realiza una translación en el tiempo:

$$(X_1, \cdots, X_n) \overset{\mathcal{L}}{=} (X_{t+1}, \cdots, X_{t+n}),\ t \in \mathbb{Z}, n \in \mathbb{N}.$$

Para facilitar la lectura hablaremos de procesos estacionarios cuado nos refiramos a los

estrictamente estacionarios.

La razón de que los procesos estacionarios gaussianos sean muy populares es que comparten muchas propiedades útiles que conciernen a sus estadísticos o predicción (véase, por ejemplo, Azencott y Dacunha-Castelle [4] y Stein [77]). Por lo tanto, un tema importante es la disponibilidad de procedimientos estadísticos que permitan evaluar la gaussianidad del proceso en estudio. En las últimas tres décadas se han desarrollado muchos trabajos para construir dichos prcedimientos. Por ejemplo, en Epps [26] encontramos un procedimiento basado en el análisis de la función caracteristica empírica, en Lobato y Velasco [58] otro basado en el test de asimetría-curtosis, en Moulines y Choukri [66] uno basado en ambos y en Subba y Gabr [78] otro basado en la función de densidad biespectral. Una desventaja importante de estos procedimientos es que sólo evaluan si las marginales hasta cierto orden del proceso son gausianas (¡por lo que conocemos sólo hasta orden uno!). Obviamente, esto da lugar a tests al nivel adecuado para el problema, pero estos tests tienen potencia nominal bajo algunas alternativas no gaussianas como los procesos estrictamente estacionarios no gaussianos que tienen marginales unidimensionales gaussianas; e incluso con marginales unidimensionales no gausianas pero que la característica de la variable que analiza el test toma el mismo valor que si la marginal fuese gaussiana.

En esta tesis se propone un procedimiento para evaluar si un proceso estrictamente estacionario es gaussiano. Este procedimiento es consistente, rechazando todas las alternativas estrictamente estacionarios que cumplan ciertas condiciones de regularidad. El procedimiento es una combinación del método de las proyecciones aleatorias (véase Cuesta-Albertos et al. [12] y Cuesta-Albertos et al. [15]) y los métodos clásicos, discutidos antes, que permiten evaluar si la marginal unidimensional de un proceso estrictamente estacionario es gaussiana.

En cuanto al método de las proyecciones aleatorias, seguimos la misma filosofía que

se propuso en Cuesta-Albertos et al. [15], donde, a grosso modo, se demuestra que (sólo) una proyección aleatoria es suficiente para caracterizar una distribución de probabilidad. Aquí se utilizarán los resultados obtenidos en Cuesta-Albertos et al. [12], donde el principal resultado de Cuesta-Albertos et al. [15] se generaliza para obtener tests de bondad de ajuste para familias de distribuciones, particularmente para familias gaussianas.

Por lo tanto, dado un proceso estrictamente estacionario, $(X_t)_{t\in\mathbb{Z}}$, estamos interesados en la construcción de un test para la hipótesis nula $H_0 : (X_t)_{t\in\mathbb{Z}}$ es gaussiano. Resulta que H_0 es válida si, y sólo si, $(X_t)_{t\leq 0}$ es un vector gaussiano. Así que, utilizando el método de las proyecciones aleatorias, Cuesta-Albertos et al. [12], esto es, a grandes rasgos, equivalente a realizar una proyección aleatoria (unidimensional) de $(X_t)_{t\leq 0}$ y ver si es gaussiana.

Para utilizar el procedimiento propuesto en Cuesta-Albertos et al. [12] necesitamos un espacio de Hilbert apropiado (para una descripción ver la Sección 6.1). Sea $< \cdot, \cdot >$ su producto escalar. Denotemos $Y_t :=< \mathbf{h}, (X_j)_{j\leq t} >$, donde $\mathbf{h}$ es un vector aleatorio en el espacio elegido de norma uno (para la elección de $\mathbf{h}$ ver Capítulo 6).

Los resultados de Cuesta-Albertos et al. [12] convierten el problema en el de comprobar que la marginal unidimensional de $(Y_t)_{t\in\mathbb{Z}}$ es gaussiana, lo cual puede realizarse con los tests clásicos. Para que podamos aplicar al proceso $(Y_t)_{t\in\mathbb{Z}}$ estos tests, necesitamos que $(Y_t)_{t\in\mathbb{Z}}$ herede de $(X_t)_{t\in\mathbb{Z}}$ las condiciones que el test requiera. Esta cuestión se analiza en la siguiente proposición. Para ello denotemos $\gamma_X(t) := \mathbb{E}[(X_0 - E[X_0])(Y_t - E[X_0])]$, con $t \in \mathbb{Z}$, la autocovarianza de orden t y, denotando por $\mu_{Y|\mathbf{h}}$ la esperanza condicional de Y_0 dado $\mathbf{h}$, definimos $\gamma_{Y|\mathbf{h}}(t) := \mathbb{E}[(Y_0 - \mu_{Y|\mathbf{h}})(Y_t - \mu_{Y|\mathbf{h}})|\mathbf{h}]$.

Proposition 1.0.7. *Sea $(X_t)_{t\in\mathbb{Z}}$ un proceso ergódico y estrictamente estacionario tal que $\mathbb{E}[|X_0|] < \infty$ y $\sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t)| < \infty$, con $\zeta \geq 0$. Entonces, condicionalmente a $\mathbf{h}$, el proceso $(Y_t)_{t\in\mathbb{Z}}$ es ergódico y estrictamente estacionario. Adicionalmente, $\mathbb{E}[|Y_0||\mathbf{h}]$ y*

$\sum_{t=0}^{\infty} t^{\zeta}|\gamma_{Y|\mathbf{h}}(t)|$ *son finitas.*

En principio, la normalidad del proceso $(Y_t)_{t\in\mathbb{Z}}$ puede analizarse con cualquiera de los test mencionados. Aquí, para fijar ideas, hemos seleccionado los propuestos en Epps [26] y en Lobato y Velasco [58]. El test de Epps analiza si la función característica de la marginal unidimensional de un proceso estacionario coincide con la función característica de una distribución gaussiana en un conjunto finito de puntos previamente fijados, $\lambda_1, ..., \lambda_N$. Sin embargo, en esta tesis los puntos empleados en el test de Epps también son elegidos aleatoriamente, lo que proporciona la consistencia de todo el procedimiento. En el siguiente teorema se muestra como aplicar el procedimiento de las proyecciones aleatorias utilizando el test de Epps.

En este resultado denotamos por $\hat{\mu}_{Y|\mathbf{h}}$ y $\hat{\gamma}_{Y|\mathbf{h}}$ a las versiones muestrales de $\mu_{Y|\mathbf{h}}$ y $\gamma_{Y|\mathbf{h}}$. Además se utiliza una función $Q_n(\cdot,\cdot,\lambda)$ cuya definición puede verse en Sección 3.3. Las condiciones de regularidad impuestas al proceso $(Y_t)_{t\in\mathbb{Z}}$ aparecen en la Sección 6.1.5.

Theorem 1.0.8. *Sea $(X_t)_{t\in\mathbb{Z}}$ un proceso ergódico y estrictamente estacionario satisfaciendo $\mathbb{E}[|X_0|] < \infty$ y $\sum_{t\in\mathbb{Z}} |t|^{\zeta}|\gamma_{\mathbf{Y}}(t)| < \infty$ para algún $\zeta > 0$. Tomemos λ con una determinada distribución P_λ y $\boldsymbol{h}$ independiente de λ con una $P_{\mathbf{H}}$.*

Adicionalmente asumamos que, condicionalmente a $\mathbf{h}$, $(Y_t)_{t\in\mathbb{Z}}$ satisface ciertas condiciones de regularidad. Sea $Q_n(\cdot,\cdot,\lambda)$ una forma cuadratica definida a partir de la función característica de $(Y_t)_{t\in\mathbb{Z}}$, (μ_n,γ_n) el minimizador de la forma cuadática más cercano a $(\hat{\mu}_{Y|\mathbf{h}},\hat{\gamma}_{Y|\mathbf{h}})$ y $A := \{(\lambda,h) : nQ_n(\mu_n,\gamma_n,\lambda) \to_d$ una distribución no degenerada$\}$.

Entonces, $(X_t)_{t\in\mathbb{Z}}$ es gaussiano si, y solo si, $(P_\lambda \otimes P_{\mathbf{H}})[A] > 0$.

En el siguiente teorema se muestra como aplicar el procedimiento de las proyecciones aleatorias utilizando el test de asimetría-curtosis de Lobato y Velasco. Es importante decir que hemos probado la consistencia del test bajo hipótesis diferentes que en Lobato y Velasco [58], ver Sección 3.3.3. Para dicho teorema utilizamos el estadístico

$$G_Y = \frac{n\hat{\mu}_3^2}{6|\hat{F}_3|} + \frac{n(\hat{\mu}_4 - 3\hat{\mu}_2^2)^2}{24|\hat{F}_4|},$$

con

$$\hat{F}_k = 2\sum_{t=1}^{\tau_n} \hat{\gamma}(t)(\hat{\gamma}(t) + \hat{\gamma}(\tau_n + 1 - t))^{k-1} + \hat{\gamma}^k,$$

donde $\tau_n < cn^{\beta_0}$ para $\beta_0 = 1 - 2/\alpha$, c es una constante positiva y $2 < \alpha < 4$.

Theorem 1.0.9. *Sea $(X_t)_{t\in\mathbb{Z}}$ un proceso ergódico y estrictamente estacionario tal que $\mathbb{E}[|X_0|] < \infty$ y $\sum_{t=0}^{\infty} |\gamma_X(t)| < \infty$. Tenemos*

1. *Si $(X_t)_{t\in\mathbb{Z}}$ es un proceso gaussiano, entonces $G_Y \longrightarrow_d \chi^2_2$.*
2. *Si $(X_t - \mu_X)_{t\in\mathbb{Z}}$ se puede escribir con la representación dada en Kavalieris [46] y $\mathbb{E}[X_0^4] < \infty$, entonces, condicionalmente a $\mathbf{h}$, G_Y diverge casi seguro a infinito cuando $\mu_3 \neq 0$ o $\mu_4 \neq 3\mu_2^2$.*

Finalmente quedan varios problemas prácticos que resolver. En primer lugar tenemos que, de acuerdo con la teoría, una única proyección es suficiente. Sin embargo, las razones de pérdida de información (y, por tanto, de potencia) que apuntábamos anteriormente, sugieren la conveniencia de tomar más de una dirección al azar.

En segundo lugar sucede que los tests de Epps y Lobato y Velasco tienen problemas de potencia (ver Sección 6.3) que, en cierto sentido, van en direcciones opuestas. Finalmente, sucede que la distribución utilizada para elegir $\mathbf{h}$ también afecta a la potencia del test (ver Sección 6.2.2), de forma positiva o negativa, dependiendo de la alternativa a la que nos enfrentemos. Una manera de buscar una solución neutral a estos dos últimos problemas es utilizar una combinación de todos los procedimientos: se trataría de elegir varias proyecciones aleatorias (varias con cada tipo de distribución) y aplicar cada uno de los dos tests a una mitad de las proyecciones.

Esta manera de actuar plantea el problema de cómo resumir toda la información disponible en un único número. Para este objetivo se han propuesto varias metodologías como son el uso de la desigualdad de Bonferroni, el bootstrap o el "False Discovery Rate". Nosotros hemos optado por esta última (ver Benjamini y Hochberg [5] y Benjamini y

Yekutieli [6]) que es razonablemente sencilla; ya que, como es bien conocido, Bonferroni es demasiado conservadora y el bootstrap es demasiado exigente desde un punto de vista computacional, aparte de la dificultad teórica de decidir en un problema determinado si el bootstrap es aplicable o no.

Los cálculos necesarios para esta tesis se ha realizado utilizando MatLab, con la excepción de los necesarios para el método "random forests" que se han llevado a cabo con el software asociado a Witten y Frank [82] que se puede descargar de http://www.cs.waikato.ac.nz/ml/weka. Los códigos desarrollados para MatLab se encuentran en el Apéndice A.

Algunos resultados de esta tesis ya han sido publicados. En lo referente a las profundidades, las Secciones 4.1, 5.1 y 5.2 aparecen en Cuesta-Albertos y Nieto-Reyes [19], donde se introduce la profundidad de Tukey aleatoria. La Sección 4.2 está publicada en Cuesta-Albertos y Nieto-Reyes [20] y una idea de lo que hemos desarrollado en la Sección 5.3 aparece en Cuesta-Albertos y Nieto-Reyes [21]. Por último, en lo que concierne al test de gaussianidad, en Cuesta-Albertos et al. [17] y Cuesta-Albertos et al. [18] podemos encontrar los resultados del Capítulo 6. Merece la pena decir que Cuesta-Albertos y Nieto-Reyes [19] ha sido utilizado ya por otros autores, concretamente en Li et al. [49] y en Shapira et al. [76].

Además, estos resultados han sido presentados en diferentes congresos, conferencias y encuentros. En particular, los resultados relativos a las profundidades fueron presentados en el XXX Congreso Nacional de Estadística e Investigación Operativa (SEIO) celebrado en Valladolid (España), el IV Encuentro de Estadística Matemática BoSanTouVal celebrado en Castro Urdiales (España) y en el I Workshop Internacional de Estadística Funcional y Operatorial (IWFOS) celebrado en Toulouse (Francia). Los resultados relativos al test de gaussianidad han sido presentados en el V Encuentro de Estadística Matemática

BoSanTouVal celebrado en Le Teich (Francia), en la I Reunión conjunta SMM-RSME celebrado en Oaxaca (México), en el XVI Encuentro Europeo de Jóvenes Estadísticos celebrado en Bucarest (Rumania) y en el XI Congreso Latinoamericano de Probabilidad y Estadística Matemática (CLAPEM) celebrado en Naiguatá (Venezuela).

Chapter 2

Introduction

A random projection consists in projecting a given data set, or in computing the marginal of a distribution, on a randomly chosen lower dimensional subspace. If the subspace onto which we project has dimension one, we call it a one-dimensional random projection.

Random projections are well-known due to the Johnson and Lindenstrauss' Lemma, Johnson and Lindenstrauss [45], which states that random projections approximately preserve pairwise distances with high probability when the subspace, on which we project, is chosen following a uniform distribution. An extension for standard Gaussian distributions can be found in Frankl and Maehara [32]. Many applications of this Lemma appear in the book Vempala [81]. In this book we can read *"Random projection is useful in many settings. (...) A natural setting is when the input data is in high-dimensional space, and it is possible to preserve essential properties for the data (for the particular problem at hand) while reducing dimensionality."* In this book, the property of preserving distances is used in many branches of Statistics.

However, our interest in random projections comes from another property they preserve: the distribution. In Cuesta-Albertos et al. [15], it is proved that only a one-dimensional randomly chosen projection is enough to distinguish between two distribu-

tions that are in a separable Hilbert space if one of them satisfies a condition on their moments. More precisely, given two distributions and a randomly chosen one-dimensional marginal of them we have that almost surely, the two distributions are different/equal if and only if the two marginals are different/equal. This result is extended to Banach spaces in Cuevas and Fraiman [24].

This property makes random projections an important tool for multidimensional and functional statistics, since in some problems they allow the dimension to be reduced to one, where we can apply the one-dimensional techniques, while obtaining conclusions in the larger space. Let us explain this in more detail: if we need to apply a particular technique in a space with a dimension larger than one, we make a projection and apply the technique in dimension one and if some regularity conditions are satisfied (Cuesta-Albertos et al. [15]), conclusions can be drawn from the distribution in the larger space. Therefore, this tool reduces the difficulty of the problem as the technique in a dimension larger than one, should it exist, is usually more involved than its one-dimensional homologue.

It may be thought that this way of proceeding is based in a rather radical idea: that only one random projection is enough. On one hand, this is not so novel as stated in the foreword of Vempala [81], written by C. H. Papadimitriou, *"This book is about the radical idea that even a random projection is often useful."* Obviously, substituting each of the data by a real number (the projected data) is a process that implies a lack of information. However, in the abstract of Hand [39] it is stated that *"...simple methods typically yield performance almost as good as more sophisticated methods to the extent that the difference in performance may be swamped by other sources of uncertainty..."* Hand's work is related to classification techniques, but his thoughts apply also to the idea that the loss of information we suffer is not relevant when considered a long side the other problems that usually appear in real data.

It is worth mentioning that Cuesta-Albertos et al. [15], despite of being recent, has already triggered some interest in random projections and their applications to several statistical problems. For instance, some problems in which it has been used are:

- Multiway ANOVA for functional data, Cuesta-Albertos and Febrero-Bande [13].
- Nonparametric identification of the distribution of heterogeneity in economic models, Fox and Gandhi [30].
- Non-parametric tests for spherical and compositional data, Cuesta-Albertos et al. [11].
- Depth measures and dual statistics, Cuevas and Fraiman [24].
- Outliers detection for functional data, Febrero-Bande et al. [27].
- Robust estimation and classification for functional data, Cuevas et al. [23].
- Goodness-of-fit tests, Cuesta-Albertos et al. [12, 16]. In addition, the test proposed in Cuesta-Albertos et al. [16] is used in Opazo et al. [68]. Furthermore, in Bugni et al. [10], the authors use Cuesta-Albertos et al. [12] as the reference to compare the goodness-of-fit test they propose.
- Finally, in Cuesta-Albertos et al. [14], some speech sonority data are classified using a Kolmogorov-Smirnov test for functional data proposed in Cuesta-Albertos et al. [15].

Through this thesis we work with one-dimensional random projections, i.e., the given data set will be in dimension one after being projected on the randomly chosen space. Therefore, in what follows we understand one-dimensional random projections as random projections unless otherwise stated.

In this work, we present two applications of the random projections. The first one is a new definition of depth that approximates the well-known Tukey depth, Tukey [80], and the second is a test of Gaussianity for strictly stationary processes. Let us introduce these two applications.

The random Tukey depth.

In recent times, depths have received some attention from statistical researchers (see, as an example, the book Liu et al. [54]). Depths are intended to order a given set in the following way: if a datum is moved toward the center of the data cloud, then its depth increases, and if the datum is moved toward the outside, then its depth decreases. More generally, given a probability distribution P defined in a multidimensional (or even infinite-dimensional) space $\mathcal{X}$, a depth tries to order the points in $\mathcal{X}$ from the "center (of P)" to the "outer (of P)". Obviously, this problem includes data sets if we consider P as the empirical distribution associated to the data set at hand. Thus, in what follows, we will always refer to the depth associated to a probability distribution P.

The first application we present here consists in defining a conceptually simple and easy to compute multidimensional depth that can be applied to functional problems and that provides results comparable to those obtained with more involved depths.

Note that in the one-dimensional case, it is reasonable to order the points using the order induced by the function

$$x \to D_1(x, P) := \min\{P(-\infty, x], P[x, \infty)\}. \tag{2.1}$$

Thus, the points are ordered following the decreasing order of the absolute values of the difference between their percentiles and 50, and the deepest points are the medians of P.

In the multidimensional case does not exist a so clear function to order points. There-

fore, several multidimensional depths have been proposed (see, for instance, Liu et al. [53, 54]) but here we are mainly interested in the *Tukey (or halfspace) depth* (see Tukey [80]). According to Zuo and Serfling [83], this depth behaves very well in comparison with various competitors. If $x \in \mathbb{R}^p$, then the Tukey depth of x with respect to P, $D_T(x, P)$, is the minimal probability which can be attained in the closed halfspaces containing x. I.e., $D_T(x, P)$ is the infimum of all possible one-dimensional depths of the one-dimensional projections of x, where those depths are computed with respect to the corresponding (one-dimensional) marginals of P. A more precise definition can be found in Section 3.1 of Chapter 3. Some other depths based on the consideration of all possible one-dimensional projections have been proposed (see, for instance, Zuo [84]). We consider that what follows could be applied to all of them, but we have chosen the Tukey depth for the sake of clarity.

The most significant drawback of the Tukey depth is the required computational time, since, as previously stated, it involves the computation of all possible one-dimensional depths. This time is more or less reasonable if $p = 2$, but it becomes prohibitive even for $p = 8$, see Mosler and Hoberg [65, p. 54]. To reduce the time, in Zuo [85, p. 2234] it is proposed that their values be approximated using randomly selected projections. Furthermore, in Cuevas et al. [23], a random depth is defined. There, given a point $x \in \mathbb{R}^p$, the authors propose to choose a finite number of vectors $v_1, ..., v_k \in \mathbb{R}^p$ at random and take as depth of x the mean of the values $D_1(\Pi_{v_i}(x), P \circ \Pi_{v_i}^{-1})$, $i = 1, ..., k$, where D_1 was defined in 2.1 and Π_v denotes the projection of $\mathbb{R}^p$ on the one dimensional subspace generated by v. Thus, $P \circ \Pi_v^{-1}$ is the marginal of P on this subspace. Our approach is closer to the suggestion in Zuo [85]: we simply replace the infimum in the definition of the Tukey depth by a minimum over a finite number of randomly chosen projections, obtaining a random approximation to the Tukey depth, which we call the *random Tukey depth.*

It is curious that, in spite of the great interest in depths in general and the Tukey

depth in particular, we are not aware of many results proving that a depth determines its corresponding distribution. In fact, with respect to the Tukey depth, we only know one result, Koshevoy [47]. Here, it is proved that if P and Q are two distributions defined on $\mathbb{R}^p$, both of them with finite support, and their Tukey depths coincide, then $P = Q$. An alternative proof to the Koshevoy result can be found in Hassairi and Regaieg [41]. We generalize this result and prove that the Tukey depth characterizes discrete distributions. We also prove it for the random Tukey depth.

Test of Gaussianity for stationary processes.

In many concrete situations the statistician observes a finite path $X_1, \ldots, X_n$ of a real temporal phenomena. A common modeling is to assume that the observation is a finite path of a second order weakly stationary process $\mathbf{X} := (X_t)_{t\in\mathbb{Z}}$ (see, for example, Gershenfeld [36]). This means that the random variable X_t is, for any $t \in \mathbb{Z}$, square integrable and that the mean and the covariance structure of the process is invariant by any translation on the time index. That is, for any $t, s \in \mathbb{Z}$, $\mathbb{E}(X_t)$ does not depend on t and $\mathbb{E}(X_t X_s)$ only depends on the distance between t and s. A more popular frame is the Gaussian case where the additional Gaussianity assumption on all finite marginal distributions of the process $(X_t)_{t\in\mathbb{Z}}$ is added. In this case, as the multidimensional Gaussian distribution only depends on moments of order one and two, the process is also strongly stationary. This means that the law of all finite dimensional marginal distributions is invariant if the time is shifted:

$$(X_1, \cdots, X_n) \stackrel{\mathcal{L}}{=} (X_{t+1}, \cdots, X_{t+n}),\ t \in \mathbb{Z}, n \in \mathbb{N}.$$

To facilitate the reading, from now on stationary means strictly stationary.

Gaussian stationary processes are very popular because they share a large number of nice properties concerning their statistics or prediction (see, for example, Azencott and Dacunha-Castelle [4] or Stein [77]). Hence, an important topic in the field of stationary

process is the implementation of a statistical procedure that allows Gaussianity to be assessed. In the last three decades, many studies have been developed to build such a procedure. For example, Epps [26] proposes a test based on the analysis of the empirical characteristic function. Lobato and Velasco [58] present another one based on the skewness and kurtosis test (also called Jarque-Bera test). Moulines and Choukri [66] introduce a test based on both, empirical characteristic function and skewness and kurtosis. In Subba and Gabr [78] we can find another test based on the bispectral density function. An important drawback of these tests is that they only consider a finite order marginal of the process (as far as we know the order one marginal!). Obviously, this provides tests at the right level for the intended problem; but these tests could be at the nominal power against some non-Gaussian alternatives. For example, the case of a stationary non-Gaussian process having one-dimensional Gaussian marginal or, even, with non-Gaussian one-dimensional marginal but with the right value of the selected characteristic.

In this thesis, we propose a procedure to assess that a strictly stationary process is Gaussian. Our test is consistent against every strictly stationary alternative satisfying some regularity assumptions. The procedure is a combination of the random projection method (see Cuesta-Albertos et al. [12] and Cuesta-Albertos et al. [15]) and classical methods that allow to assess that the one-dimensional marginal of a stationary process is Gaussian (see the previous discussion).

Regarding the random projection method, we follow the same methodology as the one proposed in Cuesta-Albertos et al. [15]. In particular, we employ the results of Cuesta-Albertos et al. [12] where the main result of Cuesta-Albertos et al. [15] is generalized to obtain goodness-of-fit tests for families of distributions, and in particular for Gaussian families.

Therefore, given a strictly stationary process, $(X_t)_{t\in\mathbb{Z}}$, we are interested in constructing

a test for the null hypothesis $H_0 : (X_t)_{t\in\mathbb{Z}}$ is Gaussian. Note that H_0 holds if, and only if, $(X_t)_{t\leq 0}$ is Gaussian. Thus, using the random projection method (Cuesta-Albertos et al. [12]), this is, roughly speaking, equivalent to stating that a (one-dimensional) randomly chosen projection of $(X_t)_{t\leq 0}$ is Gaussian. This idea allows the problem to be translated into another one consisting of checking when the one-dimensional marginal of a random transformation of $(X_t)_{t\in\mathbb{Z}}$ is Gaussian. This can be tested using a widely used procedure. Here, we will employ the Epps and Lobato and Velasco skewness-kurtosis tests. We also use a combination of them as a way to alleviate some problems that those tests present.

Furthermore, the Epps test checks whether the characteristic function of the one-dimensional marginal of a strictly stationary process coincides with that of a Gaussian distribution. This check is performed on a fixed finite set of points. As a consequence, it cannot be consistent against every possible non-Gaussian alternative with non-Gaussian marginal. However, in our work, the points employed in the Epps test will be also drawn at random. This will provide consistency to the whole test. Regarding the Lobato and Velasco skewness-kurtosis test we will prove the consistency of the test under different hypotheses than those in Lobato and Velasco [58].

The structure of the thesis is as follows. In Chapter 3 we include some already well-known definitions and results that will be used in later chapters. These are not original and are included for the sake of completeness. In Chapter 4 we introduce the notion of the random Tukey depth. Moreover, in Subsection 4.1.1 (Theorem 4.1.3) we show that, with the exception of the invariance under non-singular linear transformations, this approximation satisfies the definition of depth given in Zuo and Serfling [83] (but with the convergence in the last property being in probability). Subsection 4.1.1 closes with Theorem 4.1.5, which proves the consistency of the random Tukey depth. In Subsection 4.1.2 we extend the results of Subsection 4.1.1 to infinite-dimensional functional spaces. The proofs of these results are included in Subsection 4.1.3.

Section 4.2 is devoted to proving that the random Tukey and the Tukey depths characterize discrete distributions. To be more precise, Theorem 4.2.6 states that if P is a discrete distribution (with finite or denumerable support) defined on $\mathbb{R}^p$ and Q is a Borel distribution on $\mathbb{R}^p$ such that the functions $D_T(\cdot, P)$ and $D_T(\cdot, Q)$ coincide, then $P = Q$. Thus, this result is slightly more general than Koshevoy's theorem in the sense that it only requires one distribution to be discrete and also includes denumerably supported distributions. The result is proved, at first, for the random Tukey depth (Theorem 4.2.3) and then, a simple extension allows the Tukey depth to be covered.

Chapter 5 contains some applications of the random Tukey depth. First of all, in Section 5.1 we address the main difficulty of the random Tukey depth as an approximation to the Tukey depth, which is to find the number of random projections required to obtain a good approximation. This number could depend on the kind of application of the depth in which we are interested as well as on the dimension of the underlying space and the size of the sample we are using. However, the simulations carried out in Section 5.1 suggest that a maximum of 250 randomly chosen projections are enough to satisfy a wide range of cases. Section 5.1 ends with a comparison of the time needed to compute the random Tukey depth and that required for the Mahalanobis depth. In Section 5.2 we deal with an application of the multidimensional random Tukey depth. It consists in reproducing the simulation study carried out in Liu and Singh [57], where the authors apply depth measures to test differences in homogeneity among distributions. Our main objective is to show that the random Tukey depth provides results which are similar to those obtained in practice with the Tukey depth.

One of the main advantages of the random Tukey depth is that it can be extended to infinite-dimensional functional spaces, despite the definition of depth not being fully satisfied in this case (see Subsection 4.1.2). Thus, in Section 5.3, we apply the random

Tukey depth to a well known supervised classification problem where we are required to classify an individual as female or male, based on its growth curve. We do this using the procedures proposed in López-Pintado and Romo [60] and compare the results of the random Tukey depth with those of the depths used there. In addition, we compare our results with another two classification methods based on curves (see Cuevas et al. [23] and Li et al. [49]) and with classification methods based on the k-nearest neighbors (k-NN) and kernels. Moreover, taking into account that, in fact, the data are only 31-dimensional, we also compare it with the random forests method.

Chapter 6 outlines the Gaussianity test for stationary processes. In Section 6.1 we introduce our procedure and analyze its asymptotic behavior. Section 6.2 contains some details on the practical application of the method and Section 6.3 includes the results of the simulations and the application to two real data sets. The thesis ends with a discussion.

The computations carried out for this thesis were done using MatLab codes with the exception of the ones for the random forests. The latter were done with a software associated with Witten and Frank [82] that can be downloaded from
http://www.cs.waikato.ac.nz/ml/weka. The MatLab computational codes can be found in Appendix A.

Some results of this thesis have already been published. Regarding depths, Sections 4.1, 5.1 and 5.2 appear in Cuesta-Albertos and Nieto-Reyes [19], where the random Tukey depth is introduced. Section 4.2 is published in Cuesta-Albertos and Nieto-Reyes [20] and an idea of what we have developed in Section 5.3 is announced in Cuesta-Albertos and Nieto-Reyes [21]. Finally, regarding the test of Gaussianity, in Cuesta-Albertos et al. [17] and Cuesta-Albertos et al. [18] we can find the results of Chapter 6. It is worth noting that Cuesta-Albertos and Nieto-Reyes [19] has already been used in Li et al. [49] and

Shapira et al. [76].

Furthermore, these results have been presented in different congresses, conferences and meetings. Particularly, the results concerning data depth have been presented at the 30th National Congress in Statistics and Operational Research (SEIO) held in Valladolid (Spain), the 4th Meeting of Mathematical Statistics BoSanTouVal in Castro Urdiales (Spain) and the 1st International Workshop on Functional and Operatorial Statistics (IWFOS) in Toulouse (France). Regarding the results concerning the test of Gaussianity, these have been presented at the 5th Meeting of Mathematical Statistics BoSanTouVal held in Le Teich (France), the 1st joint Meeting SMM-RSME in Oaxaca (Mexico), the 16th European Young Statisticians Meeting in Bucharest (Romania) and the XI Latin American Congress of Probability and Mathematical Statistics (CLAPEM) held in Naiguatá (Venezuela).

Chapter 3

Preliminary results

For the convenience of the reader, the aim of this chapter is to present some non-original definitions and results, without proofs, that are relevant to the following chapter, in order to make this thesis self-contained. This chapter is divided into four main parts. The first one is devoted to depths; the second to the random projection method; the third to Gaussianity tests for stationary processes; finally, we pay some attention to the problem of multiple testing.

3.1 Definition of data depth

Here we state the definition of statistical depth. This definition consists of four key properties desirable for depths. They are affine invariance, maximality at center, monotonicity relative to deepest point and vanishing at infinity. These properties were first proposed in Liu [52]. There, the simplicial depth was defined and the properties were used to justify it as a data depth. Subsequently these were adapted in Zuo and Serfling [83] as the key properties required for any general depth function.

Concerning the notation, $\mathcal{P}$ denotes the class of distributions on the Borel sets of $\mathbb{R}^p$ and P_X the distribution of a general random vector X.

Definition 3.1.1. *The bounded and nonnegative mapping* $D(\cdot,\cdot) : \mathbb{R}^p \times \mathcal{P} \longrightarrow \mathbb{R}$ *is called a* statistical depth function *if it satisfies the following properties:*

1. $D(Ax + b, P_{AX+b}) = D(x, P_X)$ *holds for any* $\mathbb{R}^p$*-valued random vector* X*, any* $p \times p$ *nonsingular matrix* A *and any* $b \in \mathbb{R}^p$*.*

2. $D(\theta, P) = \sup_{x \in \mathbb{R}^p} D(x, P)$ *holds for any* $P \in \mathcal{P}$ *having a center of symmetry* θ*.*

3. *For any* $P \in \mathcal{P}$ *having deepest point* θ*,* $D(x, P) \leq D(\theta + \alpha(x - \theta), P)$ *holds for* $\alpha \in [0, 1]$*.*

4. $D(x, P) \to 0$ *as* $\|x\| \to \infty$*, for each* $P \in \mathcal{P}$*.*

Let us state some depths that are used throughout the thesis. Given $x \in \mathbb{R}^p$ and $P \in \mathcal{P}$, let us define

1. The *Tukey depth*, which was proposed in Tukey [80], $D_T(x, P)$:
   Given $v \in \mathbb{R}^p$, let Π_v be the projection of $\mathbb{R}^p$ on the one dimensional subspace generated by v and so $P \circ \Pi_v^{-1}$ the marginal of P on this subspace. Therefore, using (2.1) we have that

$$D_T(x, P) = \inf\{D_1(\Pi_v(x), P \circ \Pi_v^{-1}) : v \in \mathbb{R}^p\}, \ x \in \mathbb{R}^p. \tag{3.1}$$

2. The *Mahalanobis depth*, which was introduced in Mahalanobis [62], $D_M(x, P)$:
   If the mean, μ, and dispersion matrix, Σ, of P exist and Σ is not singular, we have that

$$D_M(x, P) := \frac{1}{1 + (x - \mu)^t \Sigma^{-1} (x - \mu)}. \tag{3.2}$$

It is not our purpose to provide an in-depth explanation of the various well-known definitions of multidimensional depths. For that, see Liu et al. [53] or Parelius [69].

Figure 3.1 illustrates the behavior of the Tukey (left-hand side) and Mahalanobis (right-hand side) depths. The plots represent the depths of a sample of size 200 drawn with a 2-dimensional standard Gaussian distribution. The scale of colors goes from dark red (high depth) to dark blue (low depth) through oranges, yellows and greens.

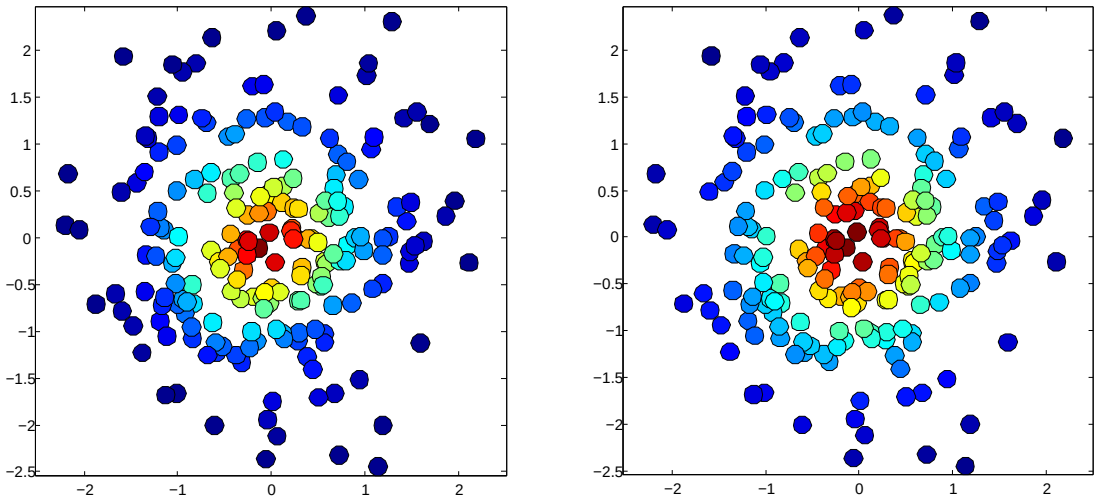

Figure 3.1: Tukey depth (left-hand side) and Mahalanobis depth (right-hand side) of a sample of size 200 drawn with a 2-dimensional standard Gaussian distribution.

Next, we state some definitions of depth in functional spaces we use throughout the thesis. They were introduced in López-Pintado and Romo [61] and used in López-Pintado and Romo [60]. Let $X_1(t), ..., X_n(t)$ be a set of real functions in $L^2[0,1]$. Let us state some notation that is needed for the definition of these depths.

- The graph of a function X defined on $[0,1]$ is $G(X) = \{(t, X(t)) \in \mathbb{R}^2 : t \in [0,1]\}$.

- The band determined by J curves, $X_{i_1}, X_{i_2}, ..., X_{i_J}$, from the sample $X_1, ..., X_n$ is

$$V(X_{i_1}, X_{i_2}, ..., X_{i_J}) = \{(t, Y) \in [0,1] \times \mathbb{R} : \min_{r=1,...,J} X_{i_r}(t) \leq Y \leq \max_{r=1,...,J} X_{i_r}(t)\}.$$

Then, the *band depth*, determined by J different curves, of a curve $X \in L^2[0,1]$ is

$$DSJ(X) := \sum_{j=2}^{J} \begin{pmatrix} n \\ j \end{pmatrix}^{-1} \sum_{1 \le i_1 < \ldots < i_j \le n} I(G(X) \subset V(X_{i_1}, X_{i_2}, ..., X_{i_j}))\}, \ J \ge 2, \quad (3.3)$$

where $I(\cdot)$ denotes the indicator function. The *generalized band depth* is

$$DGS(X) := \begin{pmatrix} n \\ 2 \end{pmatrix}^{-1} \sum_{1 \le i_1 < i_2 \le n} \lambda(\{t \in [0,1] : \min_{r=i_1,i_2} X_r(t) \le X(t) \le \max_{r=i_1,i_2} X_r(t)\}), \quad (3.4)$$

where λ is the Lebesgue measure on the interval $[0,1]$.

Thus, the band depth of X is the proportion of bands that contain X and the generalized band depth is the proportion of time that X is inside the bands with size two.

It is worth noting that although there are many definitions of depth in multidimensional spaces, this does not occur in the functional ones. However, some well-known definitions of functional depth can be found in Fraiman and Muniz [31], Cuevas et al. [23] and Cuevas and Fraiman [24].

3.2 The random projection method

This thesis is based on Corollary 3.2 and Theorem 4.1 in Cuesta-Albertos et al. [15] which are stated bellow.

Corollary 3.2.1 (Cuesta-Albertos, Fraiman y Ransford (2007))**.** *Let P,Q be Borel probability measures on $\mathbb{R}^d$, where $d \ge 2$. Assume that:*

- *the absolute moments $m_n := \int ||x||^n dP(x)$ are finite and satisfy $\sum_{n \ge 1} m_n^{-1/n} = \infty$;*
- *the set $\{v \in \mathbb{R}^p : P \circ \Pi_v^{-1} = Q \circ \Pi_v^{-1}\}$ is of positive Lebesgue measure in $\mathbb{R}^d$.*

Then $P = Q$.

Theorem 3.2.2 (Cuesta-Albertos, Fraiman y Ransford (2007))**.** *Let $\mathbb{H}$ be a separable Hilbert space, and let μ be a non-degenerate Gaussian measure on $\mathbb{H}$. Let P,Q be Borel probability measures on $\mathbb{H}$. Assume that:*

- *the absolute moments $m_n := \int ||x||^n dP(x)$ are finite and satisfy $\sum_{n\geq 1} m_n^{-1/n} = \infty$;*
- *the set $\{v \in \mathbb{H} : P \circ \Pi_v^{-1} = Q \circ \Pi_v^{-1}\}$ is of positive μ-measure.*

Then $P = Q$.

In addition, in this thesis we use the following characterization of Gaussian distributions in Hilbert spaces which comes from Cuesta-Albertos et al. [12]. It is based on the use of *dissipative distributions* which are defined next. However, let us introduce first some notation.

$\mathbb{H}$ denotes a separable Hilbert space with the inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$. $\{v_n\}_{n=1}^{\infty}$ denotes a generic orthonormal basis of $\mathbb{H}$ and V_n the n-dimensional subspace spanned by $\{v_1, \ldots, v_n\}$. For any subspace, $V \subset \mathbb{H}$ we write $V^{\perp}$ for its orthogonal complement. If $\mathbf{D}$ is an $\mathbb{H}$-valued random element, then $\mathbf{D}_V$ denotes the projection of $\mathbf{D}$ on the subspace V of $\mathbb{H}$.

Definition 3.2.3. *Let $\boldsymbol{D}$ be an $\mathbb{H}$-valued random element defined on the probability space $(\Omega, \sigma, I\!P)$. We will say that its distribution is dissipative if there exists an orthonormal basis $(v_n)_{n=1}^{\infty}$ of $\mathbb{H}$, such that*

1. *$I\!P\left(\boldsymbol{D}_{V_n^{\perp}} = 0\right) = 0$, for all $n \geq 2$.*
2. *The conditional distribution of $\boldsymbol{D}_{V_n}$ given $\boldsymbol{D}_{V_n^{\perp}}$ is absolutely continuous with respect to the n-dimensional Lebesgue measure.*

Theorem 3.6 in Cuesta-Albertos et al. [12] states the following:

Theorem 3.2.4 (Cuesta-Albertos, del Barrio, Fraiman y Matrán (2007))**.** *Let η be a dissipative distribution on $\mathbb{H}$. If $\boldsymbol{X}$ is an $\mathbb{H}$-valued random element and*

$$\eta(\{\boldsymbol{h} \in \mathbb{H} : \text{ the distribution of } \langle \boldsymbol{X}, \boldsymbol{h} \rangle \text{ is Gaussian}\}) > 0,$$

then $\boldsymbol{X}$ *is Gaussian.*

The importance of this result lies in the fact that if η is dissipative then the following $0-1$ law holds

$$\eta(\{\mathbf{h} \in \mathbb{H} : \text{ the distribution of } \langle \mathbf{X}, \mathbf{h} \rangle \text{ is Gaussian}\}) \in \{0, 1\}.$$

Moreover, $\mathbf{X}$ is not Gaussian if, and only if,

$$\eta(\{\mathbf{h} \in \mathbb{H} : \text{ the distribution of } \langle \mathbf{X}, \mathbf{h} \rangle \text{ is Gaussian}\}) = 0.$$

In other words, if we are interested in whether the distribution of $\mathbf{X}$ is Gaussian, then the only thing we have to do is to select at random a point $\mathbf{h} \in \mathbb{H}$ using a dissipative distribution and check if the real-valued random variable $\langle \mathbf{X}, \mathbf{h} \rangle$ is Gaussian. We will obtain the right answer with probability one.

3.3 Classical tests of Gaussianity for stationary processes

In this section, we present some tests for checking whether a stationary random process $(Y_t)_{t\in\mathbb{Z}}$, is Gaussian.

3.3.1 Notations and basic definitions

If Y is a random variable, we denote by Φ_Y its characteristic function; $\Phi_{\mu,\gamma}$ denotes the characteristic function of the Gaussian distribution with mean $\mu \in \mathbb{R}$ and variance $\gamma > 0$.

$\mathbf{Y}$ and $(Y_t)_{t\in\mathbb{Z}}$ denote indistinctly a process. Given a stationary process $\mathbf{Y}$, let us denote, if they exist, $\mu_Y := \mathbb{E}[Y_0]$ the mean and $\mu_{Y,k} := \mathbb{E}[(Y_0 - \mu_Y)^k]$, with $k \in \mathbb{N}$, the centered moment of order k. Further, let $\gamma_Y(t) := \mathbb{E}[(Y_0 - \mu_Y)(Y_t - \mu_Y)]$, with $t \in \mathbb{Z}$, be the autocovariance of order t.

Let $Y_1, Y_2, ..., Y_n$, $n \in \mathbb{N}$ be a sample of equally spaced observations of the random process $\mathbf{Y}$. Let $\hat{\mu}_Y := n^{-1}\sum_{i=1}^n Y_i$ be its sample mean, $\hat{\mu}_{Y,k} := n^{-1}\sum_{i=1}^n (Y_i - \hat{\mu}_Y)^k$, for

$k \in \mathbb{N}$, its sample centered moment of order k and

$$\hat{\gamma}_Y(t) := n^{-1} \sum_{i=1}^{n-|t|} (Y_i - \hat{\mu}_Y)(Y_{i+|t|} - \hat{\mu}_Y),$$

for $|t| \leq n-1$, the sample autocovariance of order t. When it is clear to which process they are referring we suppress the subindex Y. Note that then we write $\mu_{Y,k}$ as μ_k. For the sake of simplicity, let us denote $\gamma_Y := \gamma_Y(0)$ and analogously $\hat{\gamma}_Y := \hat{\gamma}_Y(0)$.

Finally, by i.i.d.r.vs. we mean independent and identically distributed random variables. Throughout this thesis, all the processes are assumed to be integrable and all the random elements are defined on the same, rich enough, probability space $(\Omega, \sigma, I\!P)$.

3.3.2 The Epps test

The test discussed in this section is a particular case of the one studied in Section 3 of Epps [26]. We begin with some notations and definitions. Given $N > 1$, let us define

$$\Lambda_N := \{\lambda := (\lambda_1, \ldots, \lambda_N)^T \in \mathbb{R}_N^+ : \lambda_i \neq \lambda_j \text{ for all } i \neq j,\ i, j = 1, ...N\},$$

where ^{T} denotes transposition.

Let $Y_1, Y_2, ..., Y_n$, $n \in \mathbb{N}$, be a sample of equally spaced observations of the random process $\mathbf{Y}$. Let $\lambda \in \Lambda_N$ and let $\hat{g}_n(\lambda)$ be the $2N$-dimensional column vector composed by the real and complex parts of the empirical characteristic function computed at λ. That is

$$\hat{g}_n(\lambda) := \frac{1}{n} \sum_{i=1}^{n} (\cos(\lambda_1 Y_i), \sin(\lambda_1 Y_i), \ldots, \cos(\lambda_N Y_i), \sin(\lambda_N Y_i))^T.$$

We often suppress the subindex n in $\hat{g}_n(\lambda)$ to simplify the notation. Further, for $\nu \in \mathbb{R}$ real and $\rho > 0$, let

$$g_{\nu,\rho}(\lambda) := (\mathrm{Re}(\Phi_{\nu,\rho}(\lambda_1)), \mathrm{Im}(\Phi_{\nu,\rho}(\lambda_1)), \ldots, \mathrm{Re}(\Phi_{\nu,\rho}(\lambda_N)), \mathrm{Im}(\Phi_{\nu,\rho}(\lambda_N)))^T,$$

be the $2N$-dimensional vector composed by the real and complex parts of $\Phi_{\nu,\rho}$ computed at λ.

We denote by $f_{\mathbf{Y}}(0,(\mu_Y,\gamma_Y),\lambda)$ the spectral density matrix (see, for example, Anderson [3]) of the process:

$$(g(Y_t,\lambda))_{t\in\mathbb{Z}} := ((\cos(\lambda_1 Y_t),\sin(\lambda_1 Y_t),\ldots,\cos(\lambda_N Y_t),\sin(\lambda_N Y_t)))_{t\in\mathbb{Z}}^T$$

at frequency 0. Note that if we assume that $(Y_t)_{t\in\mathbb{Z}}$ is a Gaussian stationary process with

$$\sum_{t\in\mathbb{Z}} |t|^{\varsigma}|\gamma_{\mathbf{Y}}(t)| < \infty, \text{ for some } \zeta > 0, \tag{3.5}$$

then the existence of $f_{\mathbf{Y}}(0,(\mu_Y,\gamma_Y),\lambda)$ is one of the conclusions of Lemma 2.1 in Epps [26].

For the construction of the test statistic, we will use the following estimator of $f_{\mathbf{Y}}(0,(\mu_Y,\gamma_Y),\lambda)$:

$$\hat{f}(0,\lambda) = (2\pi n)^{-1}\left(\sum_{t=1}^{n}\hat{G}(Y_{t,0},\lambda) + 2\sum_{i=1}^{\lfloor n^{2/5}\rfloor}(1-i/\lfloor n^{2/5}\rfloor)\sum_{t=1}^{n-i}\hat{G}(Y_{t,i},\lambda)\right), \tag{3.6}$$

where $\hat{G}(Y_{t,i},\lambda) = (g(Y_t,\lambda)-\hat{g}(\lambda))(g(Y_{t+i},\lambda)-\hat{g}(\lambda))^T$ and $\lfloor\cdot\rfloor$ denotes the integer part. The estimator (3.6) was used in Epps [26], but with 2/5 replaced by a general constant in the interval $(0,1/2)$. Notice also that this estimator is a particular case of the one proposed in Gaposhkin [33]. In Epps [26] it is proved that if $(Y_t)_{t\in\mathbb{Z}}$ is Gaussian, stationary and satisfies (3.5), then $\hat{f}(0,\lambda)$ converges almost surely to $f_{\mathbf{Y}}(0,(\mu_Y,\gamma_Y),\lambda)$.

Let $G_n^+(\lambda)$ be the generalized inverse of $2\pi\hat{f}(0,\lambda)$ and let $Q_n(\nu,\rho,\lambda)$ be the quadratic form

$$Q_n(\nu,\rho,\lambda) := (\hat{g}(\lambda)-g_{\nu,\rho}(\lambda))^T\, G_n^+(\lambda)\,(\hat{g}(\lambda)-g_{\nu,\rho}(\lambda))\,. \tag{3.7}$$

Let Θ be an open bounded subset of $\mathbb{R}\times\mathbb{R}^+$ and let $\lambda\in\Lambda_N$. We state two assumptions.

H1. The set $\Theta_0(\lambda) := \{(\nu,\rho)\in\Theta : \Phi_{\nu,\rho}(\lambda_i) = \Phi_{\mu_Y,\gamma_Y}(\lambda_i), i=1,...,N\}$ is nowhere dense in Θ.

H2. For each $(\nu,\rho)\in\Theta_0(\lambda)$ we have, $f_{\mathbf{Y}}(0,(\nu,\rho),\lambda)=f_{\mathbf{Y}}(0,(\mu_Y,\gamma_Y),\lambda)$ and

$$\left.\frac{\partial\Phi_{x,y}(\lambda_i)}{\partial(x,y)}\right|_{(x,y)=(\nu,\rho)}=\left.\frac{\partial\Phi_{x,y}(\lambda_i)}{\partial(x,y)}\right|_{(x,y)=(\mu_Y,\gamma_Y)},\ i=1,...,N.$$

Theorem 3.3.1 below describes the Gaussianity test proposed in Epps [26].

Theorem 3.3.1 (Epps (1987))**.** *Let $(Y_t)_{t\in\mathbb{Z}}$ be a stationary Gaussian process satisfying (3.5). Let Θ be an open and bounded subset of $\mathbb{R}\times\mathbb{R}^+$ and $\lambda\in\Lambda_N$ such that* **H1.** *and* **H2.** *hold. Further, let (μ_n,γ_n) be the minimizer on Θ nearest to $(\hat{\mu}_Y,\hat{\gamma}_Y)$ of the map*

$$(\nu,\rho)\to Q_n(\nu,\rho,\lambda).$$

Assume further that $f_{\mathbf{Y}}(0,(\mu_Y,\gamma_Y),\lambda)$ is positive definite. Then, for each fixed $\lambda\in\Lambda_N$, $nQ_n(\mu_n,\gamma_n,\lambda)$ converges in distribution to χ^2_{2N-2}.

Remark 3.3.1.1. Obviously a test based on Theorem 3.3.1 may not be consistent. First, the test might not detect some alternatives with Gaussian one-dimensional marginal. Second, it only focuses on the values of the characteristic function at some points, so that, the test might even fail against alternatives with non-Gaussian one-dimensional marginal but that satisfy that the characteristic functions of the one-dimensional marginal coincides with that of the corresponding Gaussian at the selected points.

3.3.3 Lobato and Velasco test

The test to assess the normality of time series that we discuss in this Subsection was introduced in Lobato and Velasco [58]. It uses the skewness-kurtosis test statistic, also called Jarque-Bera test (see Bowman and Shenton [8] and Jarque and Bera [44]), but improves previous tests of this kind because the statistic is studentized by standard error estimators.

Given a process $\mathbf{Y}$, let us denote $\tilde{F}_k:=2\sum_{t=1}^{n-1}\hat{\gamma}_Y(t)(\hat{\gamma}_Y(t)+\hat{\gamma}_Y(n-t))^{k-1}+\hat{\gamma}_Y^k$. This is an estimator of $F_k:=\sum_{t=-\infty}^{\infty}\gamma_Y(t)^k$. The test proposed in Lobato and Velasco [58] handles the statistic:

$$\tilde{G}_Y=\frac{n\hat{\mu}_{Y,3}^2}{6\tilde{F}_3}+\frac{n(\hat{\mu}_{Y,4}-3\hat{\mu}_{Y,2}^2)^2}{24\tilde{F}_4}.$$

Theorem 3.3.2 (Lobato and Velasco (2004)). *Let $(Y_t)_{t\in\mathbb{Z}}$ be an ergodic stationary process.*

- *If $(Y_t)_{t\in\mathbb{Z}}$ is Gaussian and satisfies $\sum_{t=0}^{\infty}|\gamma_{\mathbf{Y}}(t)|<\infty$, then $\tilde{G}_Y \longrightarrow \chi^2_2$ in distribution.*
- *If $(Y_t)_{t\in\mathbb{Z}}$ satisfies*
  - $\mathbb{E}[Y_t^{16}]<\infty$,
  - $\sum_{t_1=\infty}^{\infty}\cdots\sum_{t_{q-1}=\infty}^{\infty}|k_q(t_1,...,t_{q-1})|<\infty$, *for q=2,...,16, where $k_q(t_1,...,t_{q-1})$ denotes the qth-order cumulant of $Y_1, Y_{1+t_1},...,Y_{1+t_{q-1}}$,*
  - $\sum_{t=1}^{\infty}[\mathbb{E}|(\mathbb{E}(Y_0-\mu)^k|\mathcal{F}_{-t})-\mu_k|^2]^{1/2}<\infty$, *for $k=3,4$, where $\mathcal{F}_{-t}$ denotes the σ-field generated by Y_j, $j\leq -t$, and*
  - $\mathbb{E}[(Y_0-\mu)^k-\mu_k]^2+2\sum_{t_1=\infty}^{\infty}\mathbb{E}([(Y_0-\mu)^k-\mu_k][(Y_t-\mu)^k-\mu_k])>0$, *for $k=3,4$,*

  then the statistic $\tilde{G}_Y$ diverges to infinity whenever $\mu_{Y,3}\neq 0$ or $\mu_{Y,4}\neq 3\mu^2_{Y,2}$.

In Section 6.1, we will prove this theorem under lighter assumptions on the alternative. We will need the following recent result taken from Kavalieris [46]. This is an improvement of the well-known result in An et al. [2].

Theorem 3.3.3 (Kavalieris (2008)). *Let $(Y_t)_{t\in\mathbb{Z}}$ be a stationary process with the representation*

$$Y_t=\sum_{i=1}^{\infty}k(i)\epsilon_{t-i},\ \sum_{i=1}^{\infty}|k(i)|<\infty,\ \sum_{i=1}^{\infty}ik(i)<\infty,\ \mathbb{E}[\epsilon_n]=0,\ \textit{where } (\epsilon_t) \textit{ are i.i.d.r.vs.} \quad (3.8)$$

Assume that $\mathbb{E}[|\epsilon_n|^\alpha]<\infty$ for some $2<\alpha<4$. If $\tau_n<cn^\beta$ for $0<\beta<1$ and $c>0$, then

$$\max_{0\leq t\leq\tau_n}|\hat{\gamma}(t)-\gamma(t)|=o(n^{2/\alpha-1}) \textit{ almost surely.}$$

3.4 Multiple testing

In Section 6.2, we will propose applying several tests on the same sample to assess the Gaussianity of a process. Thus, we obtain several p-values $p_1,\ldots,p_k$, where k is the number of tests used and we need to employ a procedure to summarize them.

The most popular way to handle several p-values is to use the Bonferroni correction. However, it is very well-known that this procedure is too conservative. Several alternatives have been proposed in the literature in order to alleviate this problem including the bootstrap which is too computationally intensive and it is not easy to decide when it works or not in a given problem. Here, we will employ the *false discovery rate* (FDR). The FDR is the expected proportion of wrongly rejected hypotheses along the k tests. Taking into account that all the hypotheses we will make in Section 6.2 are equivalent, the FDR coincides with the level of the procedure.

The FDR was introduced in Benjamini and Hochberg [5] for independent tests. Here, we employ the improvement proposed in Benjamini and Yekutieli [6] that does not require dependence assumptions among the tests. This procedure, when applied to our case, works as follows:

Theorem 3.4.1 (Benjamini and Yekutieli (2001))**.** *Let us assume that we apply k statistical tests to check the same null hypothesis and that the ordered p-values that we obtain are $p_{(1)}, ..., p_{(k)}$, where $p_{(1)} \leq \ldots \leq p_{(k)}$.*

Let $\alpha \in (0, 1)$. The FDR of the test which rejects if the set

$$\left\{ i : p_{(i)} \leq \frac{i\alpha}{k \sum_{j=1}^{k} j^{-1}} \right\}$$

is not empty is, at most, α.

Therefore, according to the previous theorem, if we denote

$$p_0 := k \sum_{j=1}^{k} j^{-1} \min_{i=1,\ldots,k} p_{(i)}/i$$

we can reject at any level $\alpha \geq p_0$ and then, we can take p_0 as the resulting p-value of the procedure.

Chapter 4

The random Tukey depth

In this chapter we apply random projections to compute depths. The computation of the Tukey depth, also called the halfspace depth, is highly demanding, even in low dimensional spaces, because it requires all possible one-dimensional projections to be considered. In this chapter a random depth which approximates the Tukey depth is proposed. It only takes into account a finite number of one-dimensional projections which are chosen at random. Thus, this random Tukey depth requires a reasonable computational time even in high dimensional spaces. Moreover, it is easily extended to cover the functional framework. In addition, it is shown that almost surely if the random Tukey depths of two probabilities, P and Q, coincide and one of those distributions is discrete, then $P = Q$. The same is proved if the Tukey depths coincide, thus extending previously known results.

The first section introduces the notion of the random Tukey depth and establishes its properties: Section 4.1.1 is dedicated to the finite-dimensional case while Section 4.1.2 outlines the infinite-dimensional case. The detailed proofs of the results from both sections appear in Section 4.1.3. In Section 4.2 we show the characterization of discrete distributions by using the Tukey or the random Tukey depth.

4.1 Definition and main properties

In this section, we define the random Tukey depth and demonstrate that, with the exception of the invariance under non-singular linear transformations, it satisfies the definition of statistical depth given in Zuo and Serfling [83] and show its consistency, analyzing the possibility of extending it to cover infinite dimensional spaces.

4.1.1 Finite dimensional spaces

In this subsection, $\mathcal{P}$ denotes the class of probability distributions on the Borel sets of $\mathbb{R}^p$ and P_X the distribution of a general random vector X. In addition, the symbols $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ respectively denote the usual norm and scalar product in $\mathbb{R}^p$. Now, let us formally define the random Tukey depth.

Definition 4.1.1. *Let $P \in \mathcal{P}$. Let $\nu \in \mathcal{P}$ absolutely continuous, and let $v_1, ..., v_k$ be independent and identically distributed random vectors with distribution ν. The random Tukey depth of $x \in \mathbb{R}^p$ with respect to P based on k random vectors chosen with ν is*

$$D_{T,k,\nu}(x, P) = \min\{D_1(\Pi_{v_i}(x), P \circ \Pi_{v_i}^{-1}) : i = 1, ..., k\}, \ x \in \mathbb{R}^p.$$

In order to simplify the notation we will delete the subscript ν in the notation and simply write $D_{T,k}$.

Obviously, $D_{T,k}(x, P)$ is a random variable. It may seem somewhat paradoxical to take a random quantity to measure the depth of a point, which is inherently non-random. We have taken this approach for two reasons. Firstly, according to Corollary 3.2.1, if we have two different distributions and we randomly choose a marginal of them, those marginals are almost surely different. Thus, according to this result, one randomly chosen projection is enough to distinguish between two p-dimensional distributions. Since the depths determine one-dimensional distributions, a depth computed on just one random

projection allows us to distinguish between two distributions. Secondly, if the support of ν is $\mathbb{R}^p$, and, for every k, $\{v_1, ..., v_k\} \subset \{v_1, ..., v_{k+1}\}$, then

$$D_{T,k}(x, P) \geq D_{T,k+1}(x, P) \to D_T(x, P), \quad \text{a.s.} \tag{4.1}$$

Therefore, if we choose k large enough, the effect of the randomness in $D_{T,k}$ will be negligible. Of course, it is of interest to find how large k must be; values of k that are too large would render this definition useless. We analyze this point in Section 5.1.

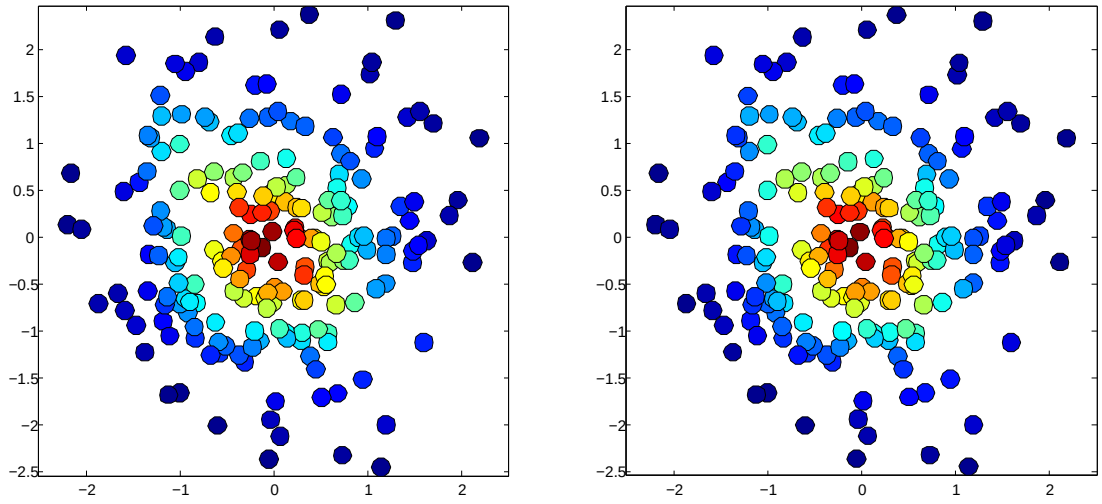

Figure 4.1: Random Tukey depth computed with $k = 5$ (left-hand side) and $k = 20$ (right-hand side) vectors, taken with a uniform distribution on the sphere, of a sample of size 200 drawn with a 2-dimensional standard Gaussian distribution.

Figure 4.1 illustrates the behavior of the random Tukey depth computed with $k = 5$ (left-hand side) and $k = 20$ (right-hand side) vectors, where ν is equal to the uniform distribution on the sphere. The plots represent the depths of a sample of size 200 drawn with a 2-dimensional standard Gaussian distribution. The scale of colors goes from dark red (high depth) to dark blue (low depth) through oranges, yellows and greens. Note that the sample to which we compute the random Tukey depth in these plots is the same as

the one used in Figure 3.1 of Chapter 3. The comparison of the plots in Figure 4.1 and the left-hand side plot in Figure 3.1 confirms the scarce differences between the random and non-random depths. In addition, at least in this case there is no point in taking $k = 20$ instead of $k = 5$ when computing the random depth, which suggests that a quite low value for k will render the randomness negligible.

Next, we show that, for every k, $D_{T,k}$ a.s. satisfies the last three properties of the definition of statistical depth, Definition 3.1.1. In Zuo and Serfling [83], it is shown that the Tukey depth satisfies this definition. Concerning the maximality at center (point *2.* in Definition 3.1.1), note that various notions of symmetry are possible, among them, central, angular and halfspace symmetry. As central symmetry implies angular, which implies halfspace, we will identify the center with the point of halfspace symmetry. For the sake of completeness, we include this definition next.

Definition 4.1.2. *θ is the center of halfspace symmetry of a distribution P if $P[H] \geq 1/2$ for every closed halfspace H containing θ.*

Theorem 4.1.3. *The random Tukey depth is a bounded and non-negative mapping which satisfies $D_{T,k}(x + b, P_{X+b}) = D_{T,k}(x, P_X)$, for any $\mathbb{R}^p$-valued random vector X and any $b \in \mathbb{R}^p$, as well as items 2. and 3. in Definition 3.1.1.*

Moreover, let $P \in \mathcal{P}$ and $k > 0$. If $\|x\| \to \infty$ with $x \in \mathbb{R}^p$, then $D_{T,k}(x, P)$ converges to zero in probability.

Remark 4.1.3.1. It is obvious that $D_{T,k,A\nu}(Ax, P_{AX}) = D_{T,k,\nu}(x, P_X)$ for any $\mathbb{R}^p$-valued random vector X and any $p \times p$ nonsingular matrix A, but it is not difficult to find examples, like Example 4.1.4, such that $D_{T,k,\nu}(Ax, P_{AX}) \neq D_{T,k,\nu}(x, P_X)$. However, in the particular case that A is a uniform scaling matrix, i.e., a constant times the identity matrix, we do have $D_{T,k,\nu}(Ax, P_{AX}) = D_{T,k,\nu}(x, P_X)$.

Example 4.1.4. We have generated a sample of size three using a 2-dimensional standard

Gaussian distribution and obtained

$$x_1 = (-0.0956, -1.3362),\ x_2 = (-0.8323, 0.7143) \text{ and } x_3 = (0.2944, 1.6236).$$

In order to obtain a 2×2 nonsingular matrix A, we have generated a sample of size two using a 2-dimensional standard Gaussian distribution and obtained

$$A = \begin{pmatrix} -0.1867 & -0.5883 \\ 0.7258 & 2.1832 \end{pmatrix},$$

which, obviously, is not singular. Taking $k = 1$ and ν being the uniform distribution in the sphere, we generate the vector in which to project obtaining $v_1 = (0.9985, 0.0555)$. Note that this setting is not particularly restrictive and the depth of x_1 and x_2 change when computing them before and after multiplying by A.

Remark 4.1.4.1. In Theorem 4.1.3, the randomness only affects item *4*. The problem is that it would be possible for all the k vectors to be included in the same hyperplane. In this case, it is obvious that item *4* could not be satisfied. For instance, if $D_{T,k}(0, P) > 0$, any sequence of points $(x_n)_{n\in\mathbb{N}}$ orthogonal to this hyperplane such that $\|x_n\| \to \infty$ does not fulfill it.

Another desirable property for depths is that its sample version converges to the population counterpart. More generally, it would be convenient if almost surely, $\sup_x |D(x, P_n) - D(x, P)| \to 0$ where P_n denotes the empirical distribution (i.e. if $x_1, ..., x_n$ is a random sample, $P_n[A] = \#(A \cap \{x_1, ..., x_n\})/n$). This property is satisfied by the Tukey depth (see Zuo and Serfling [83]) and Theorem 4.1.5 shows that the random Tukey depth also has this property.

Theorem 4.1.5. *Let $\nu \in \mathcal{P}$ and $v_1, ..., v_k$ be independent and identically distributed random vectors with distribution ν. Let $P \in \mathcal{P}$ and let $\{P_n\}$ be a sequence of empirical distributions computed on a random sample taken from P which is independent of the vectors $v_1, ..., v_k$.*

Then, conditionally on $v_1, ..., v_k$, we have

$$\sup_{x \in \mathbb{R}^p} |D_{T,k}(x, P_n) - D_{T,k}(x, P)| \to 0, \text{ almost surely [P].}$$

Remark 4.1.5.1. In Theorem 4.1.5, the almost sure convergence is with respect to the empirical samples taken from P. The random vectors employed in the computation of the depths are chosen independently of these samples. In fact, this result holds for every fixed vector in $\mathbb{R}^p$, whether randomly chosen with the distribution ν or not, the only condition being that the vector is independent of the random sample taken from P.

4.1.2 Infinite dimensional spaces

An interesting possibility of the random Tukey depth is that it can be straightforwardly extended to functional spaces. The only requirement of the main result in Cuesta-Albertos et al. [15] is that the sample space has to be a separable Hilbert space. In fact, in Cuevas and Fraiman [24] there is a generalization of this result to Banach spaces and so it is possible to extend the results on the random Tukey depth to even more general spaces. However, for this section, we focus on a separable Hilbert space. To fix ideas, we will handle the space, $\mathbb{H}$, of square-integrable functions in a given interval which, after re-scaling, we can assume to be $[0, 1]$. Thus, $\mathbb{H} = L^2[0, 1]$ and given $f, g \in \mathbb{H}$ we have $\langle f, g \rangle = \int_0^1 f(t)g(t)dt$ and $\|f\| = \langle f, f \rangle^{1/2}$.

Definition 3.1.1 can be extended to functional spaces. However, we have that the random Tukey depth is not a statistical depth in the functional case. The proofs for the invariance under translation and items *2.* and *3.* in Definition 3.1.1 (with obvious modifications such as replacing matrices with linear operators) are the same as in Theorem 4.1.3. However, the following example shows that item *4.* fails in this case even for statistical convergences.

Example 4.1.6. Let $\{\delta_n\}_{n \in \mathbb{N}} \subset \mathbb{R}^+$ with $\lim_n \delta_n = 0$, where $\mathbb{R}^+$ denotes the real positive numbers. Let $x_n \in \mathbb{H}$ such that $x_n(t) = 1/\delta_n$ if $t \in [0, \delta_n)$ and be zero otherwise.

Obviously, $\|x_n\| = (\int_0^{\delta_n} \delta_n^{-2} dt)^{1/2} = \delta_n^{-1/2}$ and then, $\lim_n \|x_n\| = \infty$. Let us take ν to be equal to the distribution of the standard Brownian motion and let $P = \nu$. Obviously Theorem 4.1 in Cuesta-Albertos et al. [15] works with this distribution. If X is a random element with distribution ν, then $\langle x_n, X\rangle$ converges to zero in probability because

$$\begin{aligned} E|\langle x_n, X\rangle| &= E\left|\int_0^{\delta_n} X(t)\delta_n^{-1} dt\right| \\ &\leq \int_0^{\delta_n} E|X(t)|\delta_n^{-1} dt = \int_0^{\delta_n} (2t/\pi)^{1/2}\delta_n^{-1} dt \leq (2\delta_n/\pi)^{1/2}, \end{aligned}$$

where the last equality holds because the distribution of $X(t)$ is $N(0,t)$. Thus, if $v_1, ..., v_k \in \mathbb{H}$ are randomly chosen with distribution ν, we have

$$\lim_n D_1(< v_i, x_n >, P \circ \Pi_{v_i}^{-1}) = D_1(0, P \circ \Pi_{v_i}^{-1}) = \max_x D_1(x, P \circ \Pi_{v_i}^{-1}) = 2^{-1},$$

because $P \circ \Pi_{v_i}^{-1}$ is a centered Gaussian distribution. Therefore, $\lim_n D_{T,k}(x_n, P) = 1/2$ while $\lim \|x_n\| = \infty$.

Thus, in this setting, the following results hold. Their proofs appear in Subsection 4.1.3.

Theorem 4.1.7. *The random Tukey depth is a bounded and non-negative mapping which satisfies $D_{T,k,\nu}(x + b, P_{X+b}) = D_{T,k,\nu}(x, P_X)$, for any $X, b \in \mathbb{H}$, as well as items 2. and 3. in Definition 3.1.1.*

Theorem 4.1.8. *Let $v_1, ..., v_k \in \mathbb{H}$. Let P be a probability distribution on $\mathbb{H}$, and let $\{P_n\}$ be a sequence of empirical distributions computed on a random sample taken from P which is independent of the vectors $v_1, ..., v_k$.*

Then, conditionally on $v_1, ..., v_k$, we have

$$\sup_{x \in \mathbb{R}^p} |D_{T,k}(x, P_n) - D_{T,k}(x, P)| \to 0, \text{ almost surely [P].}$$

4.1.3 Proofs of Subsections 4.1.1 and 4.1.2

The proofs are identical for finite or infinite dimensional spaces (except, of course, the proof of item *4.* in Theorem 4.1.3, which only works for the finite dimensional case).

Then, in this subsection the symbol $\mathcal{X}$ refers indistinctly to $\mathbb{R}^p$ or $\mathbb{H}$ and $\mathcal{P}$ denotes the class of distributions on the Borel sets of $\mathcal{X}$. Given a set $B \subset \mathcal{X}$, B^o and B^c are respectively its topological interior and complement. If $x, v \in \mathcal{X}$ and $P \in \mathcal{P}$, we denote $H_{x,v} := \{y : \langle y - x, v\rangle = 0\}$ and

$$S^P_{x,v} := \begin{cases} \{y : \langle y - x, v\rangle \geq 0\} & \text{if } P\{y : \langle y - x, v\rangle \geq 0\} \leq P\{y : \langle y - x, v\rangle \leq 0\} \\ \{y : \langle y - x, v\rangle \leq 0\} & \text{otherwise} \end{cases}. \quad (4.2)$$

The reason for using this notation is that it provides a geometrical view of the random Tukey depth which is useful for the proofs. To simplify, if there is no risk of confusion, the super-index P is omitted. With this notation, we have that $D_1(\Pi_v(x), P \circ \Pi_v^{-1}) = P(S_{x,v})$.

Proof of Theorems 4.1.3 and 4.1.7.

Clearly, the random Tukey depth is nonnegative and bounded because it is a minimum of probabilities.

To verify the remaining properties, let $P \in \mathcal{P}$, ν be an absolutely continuous distribution on $\mathcal{X}$, $k > 0$ and $v_1, ..., v_k$ be independent and identically distributed random vectors with distribution ν. Note that all the random Tukey depths used in this proof will be computed using this set of vectors.

1. Invariance under translation. This is straightforward due to the linearity of the projections.

2. Maximality at center. First, remember that if a distribution P is halfspace symmetric about θ then $P[H] \geq 1/2$ for every closed halfspace H containing θ. Assume that $\theta \in \mathcal{X}$ is the center of P, and that there exists $x \in \mathcal{X}$ satisfying

$$D_{T,k}(x, P) > D_{T,k}(\theta, P). \quad (4.3)$$

By definition of the random Tukey depth, there exists $v \in \{v_1, \ldots, v_k\}$ such that

$$D_{T,k}(\theta, P) = D_1(\Pi_v(\theta), P \circ \Pi_v^{-1}) = P(S_{\theta,v}) \geq 1/2,$$

where the inequality is due to the halfspace symmetry as $S_{\theta,v}$ is a halfspace with θ in its boundary. Thus, from (4.3) and that $P(S_{x,v}) \geq D_{T,k}(x, P)$, since this depth is computed using $v_1, ..., v_k$, we get

$$P(S_{x,v}) > 1/2. \tag{4.4}$$

We have three possibilities for the sets $S_{x,v}$ and $S_{\theta,v}$: if $x \in S_{\theta,v}$, then $S_{x,v} \subseteq S_{\theta,v}$ and if $x \notin S_{\theta,v}$, then $S_{\theta,v} \subset S_{x,v}$ or $S_{\theta,v} \subset S_{x,v}^c$. Note that there is not other possibility when $x \in S_{\theta,v}$ because of (4.2).

A $S_{x,v} \subseteq S_{\theta,v}$. Then $D_{T,k}(x, P) \leq P(S_{x,v}) \leq P(S_{\theta,v}) = D_{T,k}(\theta, P)$ which contradicts (4.3).

B $S_{\theta,v} \subset S_{x,v}$. From here and $P(S_{\theta,v}) \geq 1/2$, we obtain that $P(S_{x,v}^c \cup H_{x,v}) \leq 1/2$. Then, this and (4.4) implies $\min(P(S_{x,v}^c \cup H_{x,v}), P(S_{x,v})) = P(S_{x,v}^c \cup H_{x,v})$, which contradicts the definition of $S_{x,v}$, (4.2).

C $S_{\theta,v} \subset S_{x,v}^c$. Thus, $P(S_{x,v}^c) \geq 1/2$. Therefore, by (4.4) $1 = P(S_{x,v}) + P(S_{x,v}^c) > 1$.

3. Monotonicity relative to deepest point. Let us assume that P has a deepest point θ and there exist $x \in \mathcal{X}$ and $\alpha \in [0, 1]$ with

$$D_{T,k}(x, P) > D_{T,k}(\theta + \alpha(x - \theta), P). \tag{4.5}$$

Obviously, cases $\alpha = 0$ and $\alpha = 1$ are not possible. Then, $\alpha \in (0, 1)$. Since θ is the deepest point, we have

$$D_{T,k}(\theta, P) \geq D_{T,k}(y, P), \text{ for all } y \in \mathcal{X}. \tag{4.6}$$

Let $v \in \{v_1, ..., v_k\}$ such that $D_{T,k}(\theta + \alpha(x - \theta), P) = P(S_{\theta+\alpha(x-\theta),v})$. As $P(S_{\theta,v}) \geq D_{T,k}(\theta, P)$, from (4.5) and (4.6) it is inferred that

$$P(S_{\theta,v}) > P(S_{\theta+\alpha(x-\theta),v}). \tag{4.7}$$

Since $\alpha \in (0, 1)$, we have that $\theta + \alpha(x - \theta)$ belongs to the open segment joining the points x and θ. Thus, reasoning similarly to the final part of the proof of Statement *2*, we have one of the following three possibilities:

A $x, \theta \in H_{\theta+\alpha(x-\theta),v}$; which is impossible because this contradicts (4.7).

B $\theta \in S^o_{\theta+\alpha(x-\theta),v}$ and $x \in S^c_{\theta+\alpha(x-\theta),v}$. Here, we also have a contradiction with (4.7) because $\theta \in S^o_{\theta+\alpha(x-\theta),v}$ implies $S_{\theta,v} \subset S_{\theta+\alpha(x-\theta),v}$.

C $x \in S^o_{\theta+\alpha(x-\theta),v}$ and $\theta \in S^c_{\theta+\alpha(x-\theta),v}$. Similarly to (**B**), $x \in S^o_{\theta+\alpha(x-\theta),v}$ implies $S_{x,v} \subset S_{\theta+\alpha(x-\theta),v}$. Thus,

$$D_{T,k}(\theta+\alpha(x-\theta),P) = P(S_{\theta+\alpha(x-\theta),v}) \geq P(S_{x,v}) \geq D_{T,k}(x,P),$$

which contradicts (4.5).

4. Vanishing at infinity (only for Theorem 4.1.3). We prove this property for the case in which ν is not a probability, but the Lebesgue measure. The proof for probabilities follows the same steps until statement (4.10) below. From this point on, only some additional technicalities are required.

Let $\epsilon > 0$ such that $\epsilon < 1/2$. Since $\lim_{H\to\infty} P\{y \in \mathbb{R}^p : \|y\| \leq H\} = 1$, there exists $H_\epsilon > 0$ such that $P\{y \in \mathbb{R}^p : \|y\| \leq H_\epsilon\} > 1-\epsilon$. Furthermore, if $v \in \mathbb{R}^p$ then

$$\Pi_v^{-1}[-H_\epsilon\|v\|, H_\epsilon\|v\|] \supset \{y \in \mathbb{R}^p : \|y\| \leq H_\epsilon\}.$$

Thus, $P \circ \Pi_v^{-1}[-H_\epsilon\|v\|, H_\epsilon\|v\|] > 1-\epsilon$, for all $v \in \mathbb{R}^p$. In consequence,

$$\sup\left(D_1(-H_\epsilon\|v\|, P\circ\Pi_v^{-1}), D_1(H_\epsilon\|v\|, P\circ\Pi_v^{-1})\right) < \epsilon, \text{ for all } v \in \mathbb{R}^p. \tag{4.8}$$

Let $M > 0$ and let $x \in \mathbb{R}^p$ with $\|x\| \geq M$. Thus

$$\begin{aligned}\nu^k\left\{(v_1,...,v_k) \in (\mathbb{R}^p)^k : D_{T,k}(x,P) < \epsilon\right\} &\geq \nu^k\left\{(v_1,...,v_k) \in (\mathbb{R}^p)^k : D_{T,1}(x,P) < \epsilon\right\} \\ &= \nu\left\{v \in \mathbb{R}^p : D_{T,1}(x,P) < \epsilon\right\}, \end{aligned} \tag{4.9}$$

where we assume that $D_{T,1}$ is computed using v_1.

If $v \in \mathbb{R}^p$ satisfies that $|\langle x, v\rangle| \geq H_\epsilon\|v\|$, then by (4.8), we get:
$D_1(\Pi_v(x), P\circ\Pi_v^{-1})) < \epsilon$ and so $\{v \in \mathbb{R}^p : |\langle x,v\rangle| \geq H_\epsilon\|v\|\} \subseteq \{v \in \mathbb{R}^p : D_{T,1}(x,P) < \epsilon\}$.

Therefore, from (4.9),

$$\begin{aligned}\nu^k\left\{(v_1,...,v_k)\in(\mathbb{R}^p)^k : D_{T,k}(x,P)<\epsilon\right\} &\geq \nu\left\{v\in\mathbb{R}^p : |\langle x,v\rangle|\geq H_\epsilon\|v\|\right\}\\ &\geq \nu\left\{v\in\mathbb{R}^p : \frac{|<x,v>|}{\|x\|\|v\|}\geq\frac{H_\epsilon}{M}\right\} \quad (4.10)\\ &= \nu\left\{v\in\mathbb{R}^p : \frac{|<e_1,v>|}{\|v\|}\geq\frac{H_\epsilon}{M}\right\},\end{aligned}$$

where (4.10) comes from $\|x\| \geq M$ as it implies $\{v\in\mathbb{R}^p : |\langle x,v\rangle|M\geq H_\epsilon\|v\|\|x\|\} \subseteq \{v\in\mathbb{R}^p : |\langle x,v\rangle|\|x\|\geq H_\epsilon\|v\|\|x\|\}$. In the last equality, e_1 denotes the first element in a fixed orthonormal base of $\mathbb{R}^d$ and the equality holds because ν is rotationally invariant. Therefore, from this chain, we have

$$\inf_{x:\|x\|\geq M}\nu^k\left\{(v_1,...,v_k)\in(\mathbb{R}^p)^k : D_{T,k}(x,P)<\epsilon\right\}\geq\nu\left\{v\in\mathbb{R}^p : \frac{|<e_1,v>|}{\|v\|}\geq\frac{H_\epsilon}{M}\right\},$$

and the proof ends because, trivially,

$$\lim_{M\to\infty}\nu\left\{v\in\mathbb{R}^p : \frac{|<e_1,v>|}{\|v\|}\geq\frac{H_\epsilon}{M}\right\}=1.$$

□

Proof of Theorems 4.1.5 and 4.1.8.

Let $k>0$ and $v_1,...,v_k\in\mathcal{X}$, which remain fixed during the proof. Let P be a probability distribution on $\mathcal{X}$, let $x_1,...,x_n\in\mathcal{X}$ be a random sample taken from P and let P_n be the associated empirical distribution. Let $v\in\{v_1,...,v_k\}$. It is obvious that $\Pi_v(x_1),...,\Pi_v(x_n)$ is a random sample taken from the distribution $P\circ\Pi_v^{-1}$ and that the empirical distribution associated to those projections coincides with $P_n\circ\Pi_v^{-1}$. Moreover, $P\circ\Pi_v^{-1}$ is a distribution on the real line and, then, the Glivenko-Cantelli theorem gives:

$$\sup_{y\in\mathbb{R}}\sup\left(\left|P_n\circ\Pi_v^{-1}(-\infty,y]-P\circ\Pi_v^{-1}(-\infty,y]\right|,\left|P_n\circ\Pi_v^{-1}[y,\infty)-P\circ\Pi_v^{-1}[y,\infty)\right|\right)\to 0,\text{a.s.}$$

From here, we obtain

$$\sup_{y\in\mathbb{R}}\left(|D_1(y,P_n\circ\Pi_v^{-1})-D_1(y,P\circ\Pi_v^{-1})|\right)\to 0,\text{a.s.} \quad (4.11)$$

Therefore,

$$\begin{aligned}
&\sup_{x\in\mathcal{X}} |D_{T,k}(x,P_n) - D_{T,k}(x,P)| \\
&\qquad = \sup_{x\in\mathcal{X}} |\min_{i=1,\dots,k} D_1(\Pi_{v_i}(x), P_n\circ\Pi_{v_i}^{-1}) - \min_{i=1,\dots,k} D_{T,k}(\Pi_{v_i}(x), P\circ\Pi_{v_i}^{-1})| \\
&\qquad \leq \sup_{x\in\mathcal{X},i=1,\dots k} |D_1(\Pi_{v_i}(x), P_n\circ\Pi_{v_i}^{-1}) - D_1(\Pi_{v_i}(x), P\circ\Pi_{v_i}^{-1})| \\
&\qquad = \sup_{y\in\mathbb{R},i=1,\dots k} |D_1(y, P_n\circ\Pi_{v_i}^{-1}) - D_1(y, P\circ\Pi_{v_i}^{-1})|,
\end{aligned}$$

which converges a.s. to zero because of (4.11). □

4.2 Characterization of discrete distributions

In this section, we show that the Tukey depth (random or not) determines its corresponding distribution under the appropriate hypotheses.

Here, we are in $\mathbb{R}^p$ and additionally to the notation we introduced in Section 4.1.3, we employ the following: the unit sphere in $\mathbb{R}^p$ is denoted by $\mathbb{S}^{p-1}$, σ_{p-1} is the geometrical measure on $\mathbb{S}^{p-1}$ and, of course, $\langle\cdot,\cdot\rangle$ is the usual scalar product in $\mathbb{R}^p$. If $x\in\mathbb{R}^p$, $v\in\mathbb{S}^{p-1}$ and P is a Borel probability distribution on $\mathbb{R}^p$, let

$$\begin{aligned}
A_x^P &:= \{v\in\mathbb{S}^{p-1} : P(H_{x,v}) > P(x)\} \\
c_{x,v}^P &:= \begin{cases} 1 & \text{if } P\{y\in\mathbb{R}^p : \langle y-x,v\rangle \geq 0\} \leq P\{y\in\mathbb{R}^p : \langle y-x,v\rangle \leq 0\} \\ -1 & \text{otherwise} \end{cases}
\end{aligned}$$

Note that using $c_{x,v}^P$ we can rewrite $S_{x,v}^P$ as $\{y\in\mathbb{R}^p : c_{x,v}^P\langle y-x,v\rangle \geq 0\}$. As we did in previous section, to simplify, if there is no risk of confusion, the super-index P will be omitted. Remember that with this notation, we have $D_1(\Pi_v(x), P\circ\Pi_v^{-1}) = P(S_{x,v})$.

Given $V\subset\mathbb{S}^{p-1}$ and $x\in\mathbb{R}^p$, let us define $D_V(x,P) := \inf_{v\in V} D_1(\Pi_v(x), P\circ\Pi_v^{-1})$. Thus, if P is a discrete distribution and Z is its support, we have that

$$D_V(x,P) := \inf_{v\in V} \min\left(\sum_{z\in Z:\Pi_v(z)\leq\Pi_v(x)} P(z), \sum_{z\in Z:\Pi_v(z)\geq\Pi_v(x)} P(z) \right). \tag{4.12}$$

In the case that V contains a single element v, we will write $D_v(x, P)$. Notice that, if V is a set composed of k independent and identically distributed randomly chosen vectors using the distribution ν, then $D_V(x, P) = D_{T,k,\nu}(x, P)$. In addition, the Tukey depth of a point x with respect to P coincides with $D_{\mathbb{S}^{p-1}}(x, P)$.

Two auxiliary results follow. They require no assumption on P.

Proposition 4.2.1. *If P is a Borel distribution on $\mathbb{R}^p$ and $x \in \mathbb{R}^p$, then $\sigma_{p-1}(A_x^P) = 0$.*

Proof.
If $p = 1$, the result is trivial because $H_{x,v} = \{x\}$ for every $v \in \mathbb{S}^{p-1}$. Thus, let us assume that $p > 1$ and, also, that, on the contrary, $\sigma_{p-1}(A_x) > 0$. Thus, there exists $\alpha > 0$ such that if we denote

$$A_x^* := \{v \in \mathbb{S}^{p-1} : P(H_{x,v}) > P(x) + \alpha\},$$

then $\sigma_{p-1}(A_x^*) > 0$. Every sequence $\{v_n\} \subset A_x^*$ contains at least a couple of elements v_{i_1}, v_{i_2} such that $P(H_{x,v_{i_1}} \cap H_{x,v_{i_2}}) > P(x)$ because, if not, we would have

$$P(\cup_n H_{x,v_n}) = P(x) + P(\cup_n (H_{x,v_n} - \{x\})) = P(x) + \sum_n P(H_{x,v_n} - \{x\}) = \infty.$$

From here, the proof is ready if $p = 2$ because we can choose a sequence in A_x^* such that all their components are pairwise linearly independent and, then $H_{x,v_{i_1}} \cap H_{x,v_{i_2}} = \{x\}$.

Thus, let us assume that $p > 2$. Let us fix a hyperplane $H \subset \mathbb{R}^p$ such that $0 \in H$. Let Π_H be the projection map from $\mathbb{R}^p$ on H and S^H be the unit sphere in H. Given $h \in S^H$, let $A_{x,h}^* = \{v \in A_x^* : \Pi_H(v) = \lambda h, \text{ for some } \lambda \in \mathbb{R}^+\}$. By Fubini's theorem we have that

$$0 < \sigma_{p-1}(A_x^*) = \int_{S^H} \sigma_1(A_{x,h}^*) \sigma_{p-2}(dh).$$

Therefore, we have that $\sigma_{p-2}\{h \in S^H : \sigma_1(A_{x,h}^*) > 0\} > 0$, and, there exists $H^* \subset S^H$ with $\sigma_{p-2}(H^*) > 0$ such that for every $h \in H^*$ there exists a sequence $\{v_n^h\}_{n \in \mathbb{N}} \subset A_{x,h}^*$ composed of pairwise linearly independent vectors. Since $A_{x,h}^* \subset A_x^*$, each of those sequences contains a pair of vectors $v_{n_1}^h$ and $v_{n_2}^h$ such that

$$P\left(H_{x,v_{n_1}^h} \cap H_{x,v_{n_2}^h}\right) > P(x).$$

Thus, there exists $\beta > 0$ such that if we denote

$$H^\beta := \left\{ h \in H^* : P\left(H_{x,v^h_{n_1}} \cap H_{x,v^h_{n_2}} \right) > P(x) + \beta \right\},$$

then, $\sigma_{p-2}\left(H^\beta\right) > 0$.

Now, repeating the same reasoning as above, we have that for every sequence of $\{h_k\}_{k\in\mathbb{N}} \subset H^\beta$ there exists, at least, a couple h, h^* such that

$$P\left[\left(H_{x,v^h_{n_1}} \cap H_{x,v^h_{n_2}} \right) \cap \left(H_{x,v^{h^*}_{n_1}} \cap H_{x,v^{h^*}_{n_2}} \right) \right] > P(x).$$

Moreover, by the construction of the sequences, it turns out that the dimension of $H_{x,v^h_{n_1}} \cap H_{x,v^h_{n_2}}$ is $p-2$ and if we choose h and h^* linearly independent, then the dimension of $\left(H_{x,v^h_{n_1}} \cap H_{x,v^h_{n_2}} \right) \cap \left(H_{x,v^{h^*}_{n_1}} \cap H_{x,v^{h^*}_{n_2}} \right)$ is $p-3$. Thus the problem is solved if $p = 3$.

If $p > 3$, we have only to apply the previous reasoning to H^β and the problem will be solved if $p = 4$. If not, we will obtain a new set, whose dimension is a unit less. Thus, since the dimension is finite, we only need to repeat the process a finite number of times to get a contradiction. □

Lemma 4.2.2. *Let $x \in \mathbb{R}^p$ and $\{x_n\} \subset \mathbb{R}^p$ a sequence such that $x_n \neq x$ for all $n \in \mathbb{N}$ and $\lim_n x_n = x$. Then, if $V \subset A^c_x$, we have*

$$\liminf_n D_V(x_n, P) \geq D_V(x, P) - P(x). \tag{4.13}$$

Proof.

Let $\{u_n\} \subset V$ be such that $\lim_n (D_{u_n}(x_n, P) - D_V(x_n, P)) = 0$. By the definition of A^c_x we have that

$$P(H_{x,u_n}) = P(x), \text{ for all } n \in \mathbb{N}. \tag{4.14}$$

To obtain the result, it is sufficient to show that every subsequence $\{x_{n_k}\}$ contains a further subsequence which satisfies (4.13). To do this, let $\{x_{n_k}\}$ be a subsequence of $\{x_n\}$.

Let $z, z' \in \mathbb{R}^p$, $z \neq z'$ and $u \in \mathbb{S}^{p-1}$. It is impossible that $S_{z,u} \cap S_{z',u} = H_{z,u}$ by the definition of those sets, (4.2), because this equality implies $P(S^o_{z,u}) = P(S^o_{z',u})$ and then

by (4.2) we have $S_{z,u} = S_{z',u}$ and not $S_{z,u} \cap S_{z',u} = H_{z,u}$. Thus we get that $S_{z,u} \subset S_{z',u}$; $S_{z',u} \subsetneq S_{z,u}$ or $S_{z,u} \cap S_{z',u} = \emptyset$. Therefore the subsequence $\{x_{n_k}\}$ contains a subsequence $\{x_{n_k^*}\}$ which satisfies one of the following statements

A $S_{x,u_{n_k^*}} \subseteq S_{x_{n_k^*},u_{n_k^*}}$, for every $k \in \mathbb{N}$.

B $S_{x_{n_k^*},u_{n_k^*}} \subset S_{x,u_{n_k^*}}$, for every $k \in \mathbb{N}$.

C $S_{x_{n_k^*},u_{n_k^*}} \cap S_{x,u_{n_k^*}} = \emptyset$, for every $k \in \mathbb{N}$.

If (**A**) is satisfied, then $S_{x_{n_k^*},u_{n_k^*}} = S_{x,u_{n_k^*}} \cup (S^c_{x,u_{n_k^*}} \cap S_{x_{n_k^*},u_{n_k^*}})$. The fact that $\lim_k x_{n_k^*} = x$ implies that $\lim_k P\left(S^c_{x,u_{n_k^*}} \cap S_{x_{n_k^*},u_{n_k^*}}\right) = 0$. Thus,

$$\liminf_k D_{u_{n_k^*}}(x_{n_k^*}, P) = \liminf_k D_{u_{n_k^*}}(x, P).$$

From this, the definition of $\{u_n\}$, and the definition of depth we deduce that

$$\liminf_k D_V(x_{n_{k^*}}, P) \geq D_V(x, P) \geq D_V(x, P) - P(x).$$

In the case that (**B**) holds, we proceed similarly as in (**A**) since we have $S^o_{x,u_{n_k^*}} = S_{x_{n_k^*},u_{n_k^*}} \cup (S^o_{x,u_{n_k^*}} \cap S^c_{x_{n_k^*},u_{n_k^*}})$ and $\lim_k P\left(S^o_{x,u_{n_k^*}} \cap S^c_{x_{n_k^*},u_{n_k^*}}\right) = 0$. Then, this together with the definition of $\{u_n\}$ implies

$$\begin{aligned}\liminf_k D_V(x_{n_k^*}, P) &= \liminf_k \left(D_{u_{n_k^*}}(x, P) - P(H_{x,u_{n_k^*}})\right) \\ &= \liminf_k D_{u_{n_k^*}}(x, P) - P(x) \geq D_V(x, P) - P(x),\end{aligned}$$

where the second equality is due to (4.14) and the inequality to the definition of depth.

If the subsequence verifies (**C**) we have that $S^c_{x,u_{n_k^*}} = S_{x_{n_k^*},u_{n_k^*}} \cup (S^c_{x,u_{n_k^*}} \cap S^c_{x_{n_k^*},u_{n_k^*}})$ and $\lim_k P\left(S^c_{x,u_{n_k^*}} \cap S^c_{x_{n_k^*},u_{n_k^*}}\right) = 0$, and then

$$\begin{aligned}\liminf_k D_V(x_{n_k^*}, P) &= \liminf_k D_{u_{n_k^*}}(x_{n_k^*}, P) = \liminf_k P(S^c_{x,u_{n_k^*}}) \\ &\geq \liminf_k \left(P(S_{x,u_{n_k^*}}) - P(H_{x,u_{n_k^*}})\right) \geq D_V(x, P) - P(x),\end{aligned}$$

where the first inequality is due to the definition of $S_{x,u_{n_k^*}}$, while the second one comes from the definition of depth and (4.14). □

Now, we are in a position to prove the characterization result for random depths. Note that when V is finite, Theorem 4.2.3 provides a characterization of the random Tukey depth.

Theorem 4.2.3. *Let P and Q be two probability measures. Assume that the support of P is at most denumerable. Let V be a set at most denumerable of identically distributed random vectors $v : \Omega \to \mathbb{S}^{p-1}$ with distribution ν, absolutely continuous with respect to σ_{p-1}, defined on the probability space (Ω, σ, κ). Let*

$$\Omega_0 := \{\omega \in \Omega : D_{V(\omega)}(x, P) = D_{V(\omega)}(x, Q), \quad \text{for every } x \in \mathbb{R}^p\}.$$

Then $\kappa(\Omega_0) \in \{0, 1\}$, and $\kappa(\Omega_0) = 1$ if and only if $P = Q$.

Proof.
Obviously, if $P = Q$, then $\Omega_0 = \Omega$. Thus, the result will be proved if we show that $\kappa(\Omega_0) > 0$ implies that $P = Q$. Therefore, let us assume that $\kappa(\Omega_0) > 0$.

Let Z be the support of P. To prove the theorem, it is enough to check that $P(z) = Q(z)$ for every $z \in Z$. The proof will be based on the following lemma.

Lemma 4.2.4. *Let us assume the hypothesis of Theorem 4.2.3 and that Z is the support of P. Let $z \in Z$. Let us define $\Omega_z^P = \{\omega \in \Omega : V(\omega) \cap A_z^P = \emptyset\}$ and similarly for Q. If $\Omega_0 \cap \Omega_z^P \cap \Omega_z^Q \neq \emptyset$, then*

$$P(z) \leq Q(z).$$

By Lemma 4.2.4, if we take $z \in Z$, since ν is absolutely continuous with respect to σ_{p-1} and V is at most denumerable, from Proposition 4.2.1, we have that $\kappa(\Omega_z^P) = \kappa(\Omega_z^Q) = 1$.

Thus, $\Omega_0 \cap \Omega_z^P \cap \Omega_z^Q \neq \emptyset$, and, from Lemma 4.2.4, we obtain $P(z) \leq Q(z)$, which implies $P = Q$ because if there were a $z \in Z$ such that the inequality were strict, we would have the contradiction

$$1 = \sum_{z \in Z} P(z) < \sum_{z \in Z} Q(z) \leq 1.$$

□

Note that the independence assumption in Definition 4.1.1 is not required in Theorem 4.2.3. We are not aware of any result generalizing this characterization to the continuous case although we conjecture that it should remain valid.

Proof of the Lemma 4.2.4.

Let $z \in Z$ and $\omega \in \Omega$. Let $\{v_n(\omega)\} \subseteq V(\omega)$ be such that

$$\lim_n D_{v_n(\omega)}(z, P) = D_{V(\omega)}(z, P). \tag{4.15}$$

In what follows the symbol ω will be omitted in the notation.

As $\mathbb{S}^{p-1}$ is a compact set and $V \subset \mathbb{S}^{p-1}$, there exists a subsequence $\{v_{n_k}\}$ of $\{v_n\}$, $v_z \in \mathbb{S}^{p-1}$ and $c \in \{-1, 1\}$ such that $\lim_n v_{n_k} = v_z$ and that $c^P_{z,v_{n_k}} = c$, for every $k \in \mathbb{N}$. Without loss of generality, we can identify $\{v_{n_k}\}$ with $\{v_n\}$ and so write

$$\lim_n v_n = v_z \text{ and } c^P_{z,v_n} = c \text{ for all } n \in \mathbb{N}. \tag{4.16}$$

Let $\{z_n\} \subset (S^P_{z,v_z})^o$ such that $\lim_n z_n = z$. As $z_n \notin H_{z,v_z}$, it happens that $S^P_{z_n,v_z} \subset S^P_{z,v_z}$, for every $n \in \mathbb{N}$ and so, that $P(S^P_{z_n,v_z}) \leq P(S^P_{z,v_z}) - P(H_{z,v_z}),$ for all $n \in \mathbb{N}$. Then,

$$\limsup_n D_{v_z}(z_n, P) \leq D_{v_z}(z, P) - P(H_{z,v_z}). \tag{4.17}$$

Denoting $S := \{y \in \mathbb{R}^p : c\langle y - z, v_z\rangle \geq 0\}$ and taking into account that $P(S) \geq P(S^P_{z,v_z})$ and (4.15), we get

$$\begin{aligned} D_{v_z}(z, P) - D_V(z, P) &\leq \lim_n (P(S) - P(S^P_{z,v_n})) \\ &= \lim_n \left(P(S^o \cap (S^P_{z,v_n})^c) + P(H_{z,v_z} \cap (S^P_{z,v_n})^c) - P(S^P_{z,v_n} \cap S^c)\right). \end{aligned} \tag{4.18}$$

Let us see what happens with these three summands. Regarding the first one, due to (4.16) we have

$$\begin{aligned} \lim_n P(S^P_{z,v_n} \cap S^c) &= \lim_n P(\{y \in \mathbb{R}^p\colon\ c\langle y - z, v_n\rangle \geq 0,\ c\langle y - z, v_z\rangle < 0\}) \\ &= P(\{y \in \mathbb{R}^p : 0 > c\langle y - z, v_z\rangle \geq 0\}) = 0 \end{aligned}$$

and proceeding analogously with the first one, it is shown that $\lim_n P(S^o \cap (S^P_{z,v_n})^c) = 0$. Finally, focusing on the second one, as $z \notin (S^P_{z,v_n})^c$, we have that $P((S^P_{z,v_n})^c \cap H_{z,v_z}) \leq$

$P(H_{z,v_z}) - P(z)$. From here and (4.18),

$$D_{v_z}(z,P) - P(H_{z,v_z}) \leq D_V(z,P) - P(z). \tag{4.19}$$

As v_z is in the closure of V, because of the definition of depth (4.12), we have $D_V(z_n,P) \leq D_{v_z}(z_n,P)$. This, (1) and (2) imply that

$$\limsup_n D_V(z_n,P) \leq D_V(z,P) - P(z). \tag{4.20}$$

Remember that $V = V(\omega)$ is a random set and that (4.20) holds for every $\omega \in \Omega$. Now let us take $\omega \in \Omega_0 \cap \Omega_z^P \cap \Omega_z^Q$. Thus by Lemma 4.2.2, it happens that

$$\lim_n D_V(z_n,P) \quad = \quad D_V(z,P) - P(z) \tag{4.21}$$

and that

$$\liminf_n D_V(z_n,Q) \quad \geq \quad D_V(z,Q) - Q(z). \tag{4.22}$$

By the definition of Ω_0, $D_V(z_n,P) = D_V(z_n,Q)$ for all $n \in \mathbb{N}$ and $D_V(z,P) = D_V(z,Q)$. From here, (4.21) and (4.22), we obtain that $P(z) \leq Q(z)$. □

We end the chapter with a result which generalizes the main result in Koshevoy [47]. Its proof follows closely the one given for Theorem 4.2.3 after the following technical lemma.

Lemma 4.2.5. *Let P be a probability distribution and let $x \in \mathbb{R}^p$. If $V \subseteq (A_x^P)^c$ is a dense set in $\mathbb{S}^{p-1}$, then*

$$D_V(x,P) = D_{\mathbb{S}^{p-1}}(x,P).$$

Proof.

Let $v_0 \in A_x$. By definition of A_x, we have that $P(H_{x,v_0}) > P(x)$. Let $w_1, ..., w_{p-1} \in \mathbb{R}^p$ such that $v_0, w_1, ..., w_{p-1}$ is an orthogonal basis. Since

$$H_{x,v_0} = \{y \in H_{x,v_0} : c_{x,v_0}\langle y - x, w_1\rangle \geq 0\} \cup \{y \in H_{x,v_0} : c_{x,v_0}\langle y - x, w_1\rangle \leq 0\},$$

we have that

$$P\{y \in H_{x,v_0} : c_{x,v_0}\langle y-x, w_1\rangle \geq 0\} > P(x)$$

or, else,

$$P\{y \in H_{x,v_0} : c_{x,v_0}\langle y-x, w_1\rangle \leq 0\} > P(x).$$

Without loss of generality, we can assume that the second inequality holds. Repeating the same reasoning for w_i, $i = 2, ..., p-1$, we can also assume that

$$P\left[\cap_{i=1}^{p-1}\{y \in H_{x,v_0} : c_{x,v_0}\langle y-x, w_i\rangle \leq 0\} - \{x\}\right] > 0. \tag{4.23}$$

Furthermore, the set $W^- := \{v \in \mathbb{R}^p : \langle v, w_i\rangle < 0, i = 1, ..., p-1\}$ is open in $\mathbb{R}^p$. Since V is a dense set and v_0 belongs to the topological boundary of W^-, there exists $\{v_n\} \subset W^-$ which converges to v_0. This sequence satisfies that

$$\begin{aligned} D_{\mathbb{S}^{p-1}}(x,P) &\leq \liminf_n D_{v_n}(x,P) \leq \liminf_n P(\{y \in \mathbb{R}^p : c_{x,v_0}\langle y-x, v_n\rangle \geq 0\}) && (4.24)\\ &\leq P\left[S_{x,v_0} - \left(\cap_{i=1}^{p-1}\{y \in H_{x,v_0} : c_{x,v_0}\langle y-x, w_i\rangle \leq 0\} - \{x\}\right)\right] && (4.25)\\ &< P(S_{x,v_0}) = D_{v_0}(x,P), && (4.26) \end{aligned}$$

where the first inequality is due to $v_n \in \mathbb{S}^{p-1}$ for all n, the second one to (4.2) and the last one due to $v_n \in (A_x)^c$ for all $n \in \mathbb{N}$ but $v_0 \in A_x$. The strict inequality is due to (4.23).

The result follows on from (4.24) because it is equal to the disjoint union of A_x and $(A_x)^c$, that is, $D_{\mathbb{S}^{p-1}}(x,P) = D_{A_x \cup (A_x)^c}(x,P)$. Therefore, using (4.24), we have $D_{\mathbb{S}^{p-1}}(x,P) = D_{(A_x)^c}(x,P)$ and as $V \subseteq (A_x)^c$ is dense in $\mathbb{S}^{p-1}$, we obtain $D_{\mathbb{S}^{p-1}}(x,P) = D_V(x,P)$. □

Theorem 4.2.6. *Let P and Q be two probability measures such that P is discrete and that for any $x \in \mathbb{R}^p$, $D_{\mathbb{S}^{p-1}}(x,P) = D_{\mathbb{S}^{p-1}}(x,Q)$. Then $P = Q$.*

Proof.

Let $z \in Z$, with Z being the support of P. From Proposition 4.2.1, it is obvious that $V_{P,Q} := (A_z^P)^c \cap (A_z^Q)^c$ is a dense subset of $\mathbb{S}^{p-1}$, which satisfies Lemma 4.2.5 for P and Q.

Moreover, we can consider that the set $V_{P,Q}$ is composed of a family of (non identically distributed) random vectors with constant values equal to each element in this set. Let us denote by (Ω, σ, κ) the probability space in which those random vectors are defined.

Obviously, $\kappa\{\omega \in \Omega : V_{P,Q}(\omega) \cap A_z^P = \emptyset\} = \kappa\{\omega \in \Omega : V_{P,Q}(\omega) \cap A_z^Q = \emptyset\} = 1$, and, from this point on, we can repeat the proof of Theorem 4.2.3 to obtain the result since, in the proof of Theorem 4.2.3, we only required the hypothesis of the set V to be denumerable and of identically distributed random vectors in order to guarantee that $\kappa(\Omega_z^P) = \kappa(\Omega_z^Q) = 1$. □

Chapter 5

Applications of the random Tukey depth

As stated before, it seems rather strange to employ a random quantity to measure something not random. The only way to decide whether this is reasonable or not is to look at the results in practice. For this, we have selected some applications of depths which are proposed in the literature. We have replaced in those applications the depth used by the random Tukey depth and have compared the results. For more applications of depths, see for example Li and Liu [50], Liu [51] and Liu and Singh [55, 56]. Before starting with this program, we need to know how many projections should be taken in the computation of the random Tukey depth. For this, we have carried out some simulations indicating how many projections should be considered, depending on the kind of problem, sample size and dimension of the sample space, among others. As a conclusion of this chapter, it should be noted that this depth, based on a very low number of projections, obtains results very similar to those obtained with other depths.

Section 5.1 is dedicated to the study of the number of vectors required in the random Tukey depth, while the remaining two Sections compare the random Tukey depth with other depths through simulations; Section 5.2 analyzes the multidimensional case and

Section 5.3 the functional one.

5.1 How many random projections?

Obviously, Theorem 4.1 in Cuesta-Albertos et al. [15] also holds if ν is a probability distribution absolutely continuous with respect to the surface measure on the unit sphere in $\mathbb{R}^p$. In this section, we fix ν as the uniform distribution on the unit sphere to analyze the question of the selection of k. Our proposal is to make this selection according to the problem we have at hand; for instance with bootstrap (as in Section 5.2) or with cross-validation (as in Section 5.3). However, it is good to first have an idea about the range in which to look for this value. The obvious way to do this is to make some comparisons between D_T and $D_{T,k}$ for several dimensions, sample sizes and distributions; however, the long computational times required to obtain D_T make those comparisons impractical. Rather than performing these comparisons, we have chosen situations in which the depth of the points are clearly defined and can easily be computed with a different depth.

Let P be a probability distribution such that there exist $\mu \in \mathbb{R}^p$ and $\Sigma \in \mathbb{R}^{p\times p}$ positive definite and the density function of P can be written as $f_P(x) = h[g(\Sigma^{-1/2}(x-\mu))]$, where $h : \mathbb{R}^+ \to \mathbb{R}$ is a decreasing function and $g : \mathbb{R} \to \mathbb{R}^+$ is a convex function satisfying $g(x) = g(|x|)$. Note that this definition includes the elliptical distributions (where $g(x) = x'x$) as well as those distributions with an independent double exponential (where $g(x) = \sum_{i=1}^p |x_i|$) or Cauchy marginals (where $g(x) = \prod_{i=1}^p (1+x_i^2)$). As this kind of distributions are halfspace symmetric, by item *2.* in Definition 3.1.1, μ is the deepest point. Moreover, it seems reasonable to consider x to be deeper than y if $g[\Sigma^{-1/2}(x-\mu)] \geq g[\Sigma^{-1/2}(y-\mu)]$. Thus, in this situation, every depth of a given point x should be a monotone function of a decreasing function of $g(\Sigma^{-1/2}(x-\mu))$. In particular, in the case of elliptical distributions, every depth should be a monotone function of the Mahalanobis depth (see (3.2) for the definition). As $D_M(x,P) = 1/(1+g(\Sigma^{-1/2}(x-\mu)))$

when P is elliptical, let us define $D_{M^*}(x, P) := 1/(1 + g(\Sigma^{-1/2}(x - \mu)))$ for the kind of distributions we have here. Therefore, we can have an idea about the right k in $D_{T,k}$ as follows: if P is of the kind we have here defined, $D_T(\cdot, P)$ should be a monotone function of $D_{M^*}(\cdot, P)$. Thus, from (4.1), the larger the k, the larger the resemblance between $D_{T,k}(\cdot, P)$ and a monotone function of $D_{M^*}(\cdot, P)$. However, there should exist a value k_0 from which this resemblance starts to stabilize, and, then, there is no point in taking a $k > k_0$. The analysis of these k_0 will give us the information we are looking for.

Given that depths only attempt to rank points according to their closeness to the center of P, it is logical to measure the resemblance between $D_{T,k}(\cdot, P)$ and $D_{M^*}(\cdot, P)$ looking only at the ranks of the points. This is equivalent to using the Spearman correlation coefficient, ρ. Then, the resemblance that we handle here is

$$r_{k,P} := \rho\left(D_{T,k}(X, P), D_{M^*}(X, P)\right), \tag{5.1}$$

where X is a random variable with distribution P. As stated, if P is a distribution of the kind we have defined here, then the function $k \to r_{k,P}$ is strictly increasing and we try to identify the point k_0 from which the increments become negligible. However, in practice, we will not have a distribution P, but a random sample $x_1, ..., x_n$ taken from P. This leads us to replace P in (5.1) by the empirical distribution P_n. To illustrate the behavior of the function $k \to r_{k,P_n}$, we have represented it for different distributions, sample sizes and dimensions in Figure 5.1. In this figure, the first column corresponds to centered Gaussian distributions having covariance matrices with ones on the diagonal and 0.9 in all positions off-diagonal. The remaining columns in Figure 5.1 represent, from left to right, standard Gaussian distributions, distributions with independent double exponential marginals and distributions with independent Cauchy marginals.

Dimensions and sample sizes vary in rows. We consider, from top to bottom, sample sizes $n = 25, 100$ for $\mathbb{R}^2$, $n = 50, 100$ for $\mathbb{R}^8$ and $n = 100, 500$ for $\mathbb{R}^{50}$. The case $n = 100$ (second, fourth and fifth rows) can be used to see how the dimension affects the function

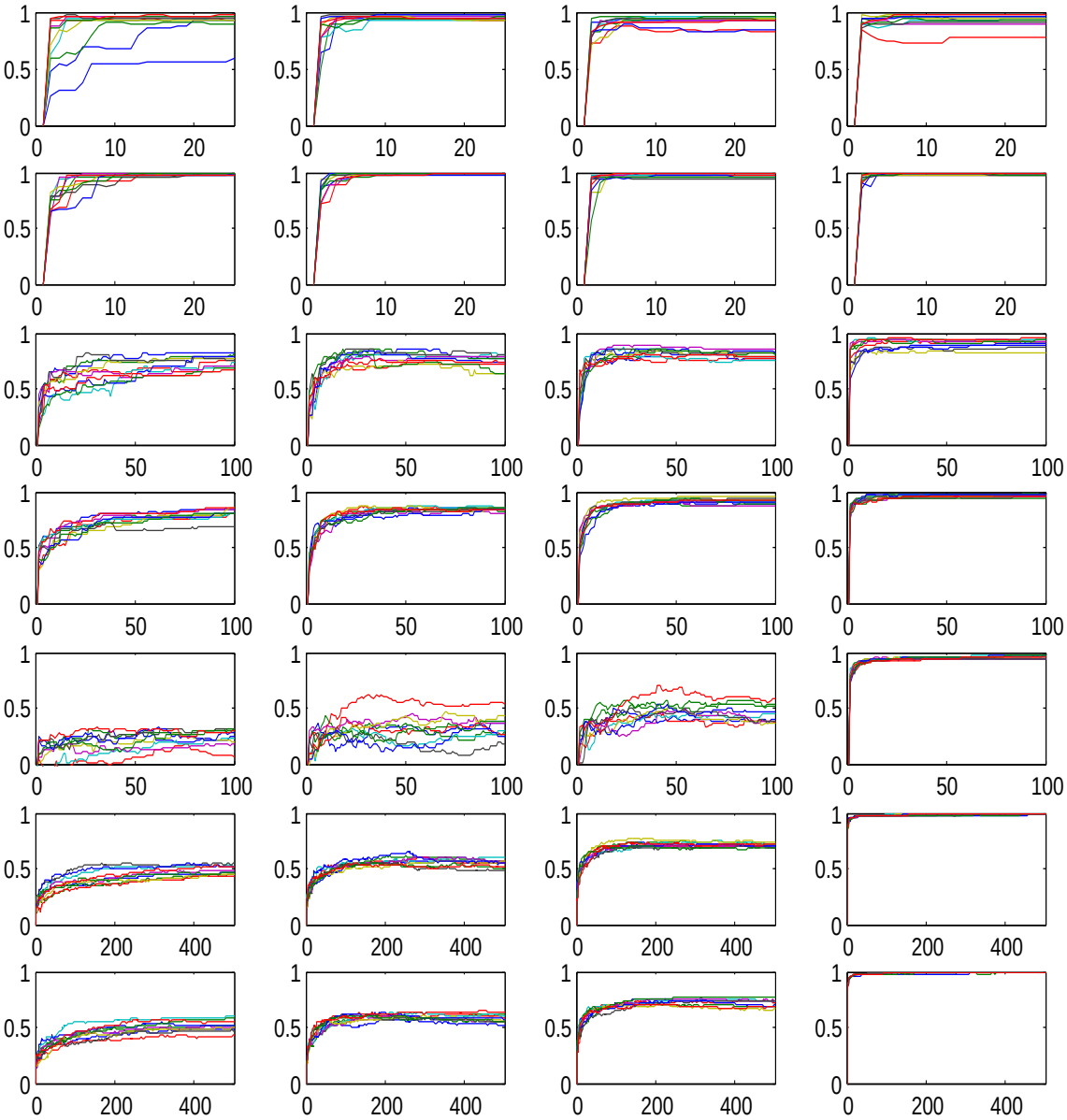

Figure 5.1: Representation of the function $k \to r_{k,P_n}$ defined in (5.1) for several dimensions, sample sizes and distributions. The underlying models are described in the text.

for a fixed sample size. The last row is different. In this row we take advantage of the fact that we know the exact covariance matrix of the theoretical distribution. Thus, in row number seven $D_{M^*}(\cdot, P)$ is computed with the exact value of Σ. In this case, we have taken $n = 500$ in $\mathbb{R}^{50}$.

A final comment is related to the computation of the location center and the dispersion matrix (except in the last row) of P_n, to be used in D_{M^*}. These parameters should depend on the distribution which generated the sample. Hence, the covariance matrix is an appropriate parameter in the Gaussian and double exponential case. However, it is not adequate for the Cauchy distribution, where we have identified Σ with the robust covariance matrix proposed in Maronna et al. [63, p. 206]. Furthermore, we have replaced μ by the sample mean in the Gaussian case and by the coordinate-wise median in the exponential and Cauchy settings.

In the graphs, k varies in set $\{1, ..., 25\}$ in the first and second rows, in $\{1, ..., 100\}$ in the third, fourth and fifth rows, and in $\{1, ..., 500\}$ in the last two rows. Moreover, there are no obvious differences between using the theoretical covariance matrices or their estimation nor between using the case of independent marginals or dependent ones. We have verified more cases (not shown here) with similar results, for which we have analyzed some intermediate dimensions, other sample sizes, and dispersion matrices with 0.5 in all off-diagonal elements for the Gaussian, exponential and Cauchy distributions.

It seems that the graphs stabilize for $k \leq 10$ if $p = 2$, $k \leq 60$ if $p = 8$ and $k \leq 250$ if $p = 50$. These values are suitable for computations and, of course, are well below those normally used to compute the Tukey depth.

Since P_n does not follow the model exactly, the function r_{k,P_n} is not necessarily increasing and in fact it may sometimes, after an initial increase, start to decrease. We

believe that this occurs because, although $D_T(x, P_n)$ is not exactly an increasing function of $D_{M^*}(x, P_n)$, there exists a increasing function, δ, such that $D_T(\cdot, P_n)$ and $\delta[D_{M^*}(\cdot, P_n)]$ are very similar. Furthermore, as k increases, $D_{T,k}(x, P_n)$ approaches $D_T(x, P_n)$. Thus, while $D_{T,k}(x, P_n)$ is not too close to $D_T(x, P_n)$, increments in k mean more similarity between $D_{T,k}(x, P_n)$ and $D_M(x, P_n)$. However, from a certain point on, getting closer to $D_T(x, P_n)$ implies moving away from $D_M(x, P_n)$.

5.1.1 Computational time

We end this section by paying some attention to the computational time required to compute the random Tukey depth. As a comparison, we have selected the time necessary to compute the Mahalanobis depth, which is one of the quickest depths according to Table 1 in Mosler and Hoberg [65]. In Table 5.1 we present the mean time required, from 200 simulations, to compute the random Tukey and Mahalanobis depths for all points in a sample with the shown sizes and dimensions. The samples were drawn using a standard normal distribution and the numbers of random directions used correspond with the upper bounds obtained in the previous part of this section. Therefore, in this case $D_{M^*} = D_M$.

To make a reliable comparison between the computational times, we need to compare the time required to compute the random Tukey depth of a sample to the time required to compute the Mahalanobis depth of the same sample. However, we must keep in mind that the first depth to be computed may have an advantage as the RAM memory may be cleaner than when the second depth is computed. In order to avoid this, we have computed the random Tukey depth first 100 times and the Mahalanobis depth first 100 times. The computations have been carried out on a Xserve G5 PowerPC G5 Dual 2.3 GHz computer with 2Gb of RAM memory.

It can be observed that the time required to compute the random Tukey depth is

Dimension	Random vectors	Sample size	Random Tukey	Mahalanobis
$p = 2$	$k = 10$	$n = 25$	$4.349 \cdot 10^{-4}$	.0014
		$n = 100$	$6.322 \cdot 10^{-4}$	.0024
$p = 8$	$k = 60$	$n = 50$	.0047	.0017
		$n = 100$	.0105	.0028
$p = 50$	$k = 250$	$n = 100$	.1153	.0047
		$n = 500$	.5596	.0158

Table 5.1: Time, in seconds, to compute the random Tukey and the Mahalanobis depths of all points in a sample

acceptable in every case. Moreover, it is better than that required to compute the Mahalanobis depth for low dimensions like $p = 2$, of the same order of magnitude for $p = 8$ and worse for dimensions around 50.

5.2 Multidimensional random Tukey depth. Testing homogeneity

Our goal in this section is to show how the random Tukey depth, with values for k of the order suggested by Figure 5.1, provides results which are similar to those obtained in practice with the Tukey depth. To this end, we are going to reproduce the simulation study carried out in Liu and Singh [57], where the authors apply depth measures to test differences in homogeneity between two 2-dimensional distributions. Since the Tukey depth is computable when the dimension is two, this is a good framework for the comparison.

Let us begin with a brief description of the procedure proposed in Liu and Singh [57] to test differences in homogeneity by using depth measures. Additional details can be

found in Liu and Singh [57].

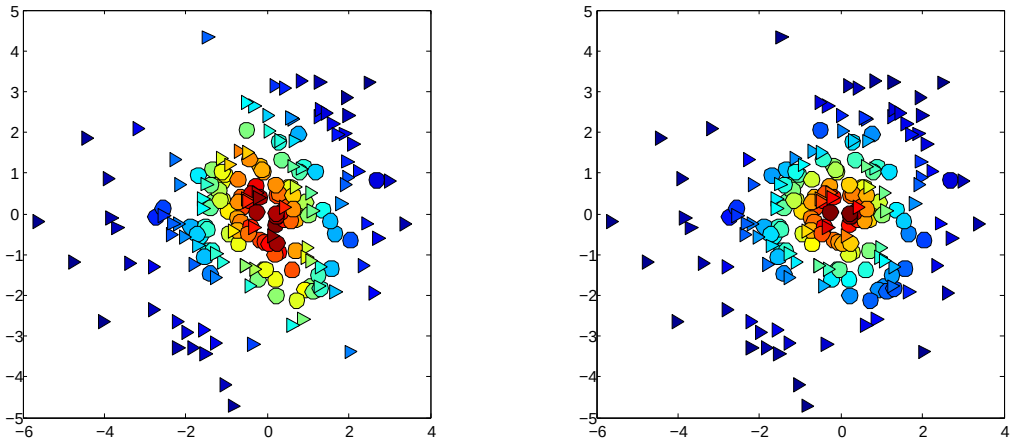

Figure 5.2: Random Tukey depth using five random projections (left-hand side) and Tukey depth (right-hand side) of two samples of size 100 drawn with 2-dimensional Gaussian distribution with a different scale.

Assume that we have two random samples $\{X_1, ..., X_{n_1}\}$ and $\{Y_1, ..., Y_{n_2}\}$ taken from the centered distributions P and Q, respectively. Let us assume that those distributions coincide except for a scale factor, i.e., we assume that there exists $r > 0$ such that the random variables $\{rX_1, ..., rX_{n_1}\}$ and $\{Y_1, ..., Y_{n_2}\}$ are identically distributed. The problem consists in testing, at level α, the hypotheses:

$$H_0 : \quad r = 1 \text{ (both scales are the same)}$$
$$H_a : \quad r > 1 \text{ (Q has a larger scale).}$$

Under the alternative, the observations in the second sample should appear in the outside part of the joint sample $\{X_1, ..., X_{n_1}, Y_1, ..., Y_{n_2}\}$, and, consequently, should have lower depths than the points in the first sample. We can see this in each plot of Figure 5.2 where two samples are represented, the first one in circles and the second one in triangles. In the left-hand side plot, the colors are chosen using the random Tukey depth with five random projections and in the right-hand side with the Tukey depth. As before, dark red

means high depth and dark blue low depth. Both samples have size 100 and were drawn using a 2-dimensional standard Gaussian distribution and multiplying the second sample by two. The colors of each of the elements depend on their depth in the joint sample. We have taken in Figure 5.2 only five vectors to compute the random Tukey depth due to we will see later that a low number is enough to test homogeneity.

Therefore, it is possible to test H_0 against H_a by computing the depths of the points $\{Y_1, ..., Y_{n_2}\}$ in the joint sample, replacing them by their ranks and rejecting H_0 if those ranks are small. The Wilcoxon rank-sum test can be used to test when the ranks of the points $\{Y_1, ..., Y_{n_2}\}$ are small. In Liu and Singh [57] several possibilities are proposed to break the ties. We have tried all of them, with no relevant differences. Thus, we have chosen random tie-breaking as the only method to be presented here.

To select the number of random projections, we have come up with the following bootstrap-based process. According to the graphs in Figure 5.1, the number of required directions should be less than 10, since we are in dimension two. Just to be on the safe side, we begin by selecting 25 vectors at random, $v_1, ..., v_{25}$. Our aim is to choose a subset $v_1, ..., v_k$, with $k \leq 25$. For that, let us denote by Z a sample formed by joining the samples that we have. That is, $Z := \{X_1, ..., X_{n_1}, Y_1, ..., Y_{n_2}\}$. To choose k we have applied the test 100 times to two bootstrap samples taken from Z where the second bootstrap sample is modified so that it satisfies the alternative hypothesis for some r in the grid $R := \{1.1, 1.2, 1.3, 1.5, 1.7, 2, 2.5, 3, 4, 5, 7, 10, 20, 30, 40, ...\}$. The process is as follows:

1. Draw two bootstrap samples, I and J, from Z, respectively with sizes n_1 and n_2.
2. Center I and J in median, separately.
3. Initialize $k = 1$ and $i = 1$.
4. Multiply the vectors in J by the i-th element in R.
5. Compute the random Tukey depth of the points in $I \cup J$ using the vectors $v_1, ..., v_k$.

6. If H_0 is rejected at level α, then keep a record of k and finish.

   Else: if $k = 25$, then go back to point (4) with $k = 1$ and $i = i + 1$.

   Else: go back to point (5) with $k = k + 1$.

With respect to the centering in median of the bootstrap samples (item 2.), it should be noted that when there is at most one observation with the same value as the median, the procedure we have presented above remains valid. However, let us assume that the sample J contains several values (let us say $h > 1$) which coincide with its median, m_J. After centering, those values go to zero, and they are not modified when multiplying by a value r. Thus, since those values remain in the inner part of the joint sample, the null hypothesis is never rejected in the bootstrap world. In order to avoid this undesirable behavior, we have defined

$$J_l := \sup\{z \in J : z < m_J\} \text{ and } J_u := \inf\{z \in J : z > m_J\},$$

and replaced those repeated values which coincide with the median by a random sample with size h taken with the uniform distribution in the interval $((J_l + m_J)/2, (J_u + m_J)/2)$. Then, we have centered the modified sample of J using the median computed in this modified sample which contains no repeated data.

At the end of the bootstrap step, we have 100 values of k, where each corresponds to the number of vectors used the first time the homogeneity hypothesis was rejected in the bootstrap world. Equivalently, every $k = 1, ..., 25$ has associated the number of times, $n_k \geq 0$, in which the null hypothesis was rejected using the vectors in the set $\{v_1, ..., v_k\}$. Since we have made 100 bootstrap replications, it is obvious that $n_1 + ... + n_{25} = 100$. The precise k to be used in the real-world test should be chosen based on the information provided by the probability distribution which gives mass $n_k/100$ to the point $k = 1, 2, \ldots, 25$. We have considered four possibilities: the mean, the median, the 80% percentile and even the maximum of this distribution. We have repeated the procedure

5,000 times using the four posibilities.

In Table 5.2 we show the rates of rejections when we carry out the test at the significance level $\alpha = .05$. The distributions used in the simulations are the 2-dimensional standard Gaussian, and the double exponential and Cauchy with independent marginals. We have considered the values $r = 1, 1.2, 2$, and $n_1 = n_2 = n$ with $n \in \{20, 30, 100\}$, and have made 5,000 simulations for each combination of distribution, sample size and r.

Sample size		$n = 20$			$n = 30$			$n = 100$		
Scale factor		$r = 1$	$r = 1.2$	$r = 2$	$r = 1$	$r = 1.2$	$r = 2$	$r = 1$	$r = 1.2$	$r = 2$
Cauchy	mean	.053	.119	.541	.053	.157	.714	.054	.270	.991
	medi	.053	.121	.525	.054	.153	.699	.053	.261	.986
	perc	.054	.120	.540	.053	.159	.716	.053	.279	.993
	max	.053	.122	.540	.054	.155	.715	.053	.279	.991
Gaussian	mean	.047	.208	.937	.048	.294	.993	.049	.656	1
	medi	.050	.200	.928	.048	.279	.989	.048	.636	1
	perc	.048	.212	.938	.046	.297	.995	.048	.687	1
	max	.047	.214	.940	.049	.303	.995	.044	.702	1
D. Exp.	mean	.053	.170	.827	.048	.207	.948	.049	.457	1
	medi	.053	.164	.813	.050	.201	.936	.048	.444	1
	perc	.051	.170	.828	.050	.216	.950	.048	.476	1
	max	.046	.164	.824	.052	.214	.951	.048	.487	1

Table 5.2: Rates of rejections in 5,000 simulations using the random Tukey depth for the considered methods to choose k, distributions, sample sizes and values of r. The dimension is $p = 2$. The significance level is .05.

From the table, it can be observed that the rejection rates depend on the distribution and, of course, on the value of r we have each time and not so much on the method used to select the number of random projections. Despite the low differences, the worst rates

were obtained with the median and the next worst with the mean. The rates obtained were very close when employing the 80% percentile and the maximum.

The number of projections used varies with the method employed to select them, the distribution and the sample size. In particular, they decrease with the sample size. This can be observed in Table 5.3, where the medians of the number of vectors used in each case are displayed.

Sample size		$n = 20$			$n = 30$			$n = 100$		
Scale factor		$r = 1$	$r = 1.2$	$r = 2$	$r = 1$	$r = 1.2$	$r = 2$	$r = 1$	$r = 1.2$	$r = 2$
Cauchy	mean	5	5	5	4	4	4	3	3	3
	med	3	3	3	2	2	2	2	2	2
	perc	8	8	8	6	6	6	3	3	3
	max	24	24	24	23	23	24	18	18	18
Gaussian	mean	5	5	5	4	4	4	3	3	3
	medi	3	2	2	2	2	2	2	2	2
	perc	6	6	7	5	5	5	3	3	3
	max	24	24	24	23	23	23	17	17	18
D. Exp.	mean	5	5	5	4	4	4	3	3	2
	medi	2	2	2	2	2	2	2	2	2
	perc	7	7	7	5	5	5	3	3	3
	max	24	24	24	23	23	23	18	18	18

Table 5.3: Medians of the number of employed random vectors in each of the cases of Table 5.2.

The small differences in the rejection rates of Table 5.2 suggest that the precise value of k is not significant in terms of application. The small values of k obtained for all the procedures (except the maximum) reinforce the impression provided by Figure 5.1 that values for k well below 10 are enough for dimension $p = 2$.

In order to compare with the Tukey depth, Table 5.4 contains the rejections obtained with the 80% percentile and, between parenthesis, the rejection rates when the random Tukey depth is replaced by the Tukey depth computed using 1,000 directions uniformly scattered on the upper halfspace.

Sample size	$n = 20$			$n = 30$			$n = 100$		
Scale factor	$r = 1$	$r = 1.2$	$r = 2$	$r = 1$	$r = 1.2$	$r = 2$	$r = 1$	$r = 1.2$	$r = 2$
Cauchy	.054	.120	.540	.053	.159	.716	.053	.279	.993
	(.055)	(.125)	(.539)	(.049)	(.146)	(.704)	(.049)	(.291)	(.991)
Gaussian	.048	.212	.938	.046	.297	.995	.048	.687	1
	(.049)	(.216)	(.940)	(.052)	(.292)	(.995)	(.049)	(.699)	(1)
D. Exp.	.051	.170	.828	.050	.216	.950	.048	.476	1
	(.057)	(.174)	(.824)	(.050)	(.223)	(.943)	(.048)	(.495)	(1)

Table 5.4: Rates of rejections in 5,000 simulations using the random Tukey depth (between parentheses, the rate with D_T) for the considered distributions, sample sizes and values of r. The dimension is $p = 2$. The significance level is .05.

In Liu and Singh [57], previous ideas are also applied to verify the homogeneity among K samples, $K > 2$. Let $\{X_{1,1}, ..., X_{1,n_1}\}, ..., \{X_{K,1}, ..., X_{K,n_K}\}$ be random samples obtained, respectively, from the distributions $P_1, ..., P_K$ and let us assume that there exists $r_1, ..., r_{K-1} > 0$ such that the random vectors

$$\{r_1 X_{1,1}, ..., r_1 X_{1,n_1}\},\ ...,\ \{r_{K-1} X_{K-1,1}, ..., r_{K-1} X_{K-1,n_{K-1}}\} \text{ and } \{X_{K,1}, ..., X_{K,n_K}\}$$

are identically distributed. We are interested in testing the following hypotheses:

$$H_0 :\quad r_i = 1, i = 1, ..., K-1 \text{ (all scales are the same)}$$
$$H_a :\quad \text{there exists } r_i \neq 1 \text{ (scales are different).}$$

For this, we center each sample separately, join all the observations in a single sample, compute the depths of all the points with respect to the empirical distribution of the single sample and transform those depths into ranks. Note that under the alternative, we should expect that some of the samples have higher ranks in mean than the rest. Then, we can apply the Kruskal-Wallis test (see Hettmansperger [43]) to verify whether there is a lack of homogeneity among the samples.

We have carried out a simulation study applying this procedure in the 2-dimensional case using the Tukey depth and the random Tukey depth with Gaussian distributions, $K = 3$ and sample sizes $n_1 = n_2 = n_3 = n$, where $n \in \{20, 30\}$. We have carried out 5,000 replications in each case at the significance level $\alpha = .05$. The selection of k to compute the random Tukey depth is made analogously to the previous case, excepting that in the bootstrap procedure we take now three bootstrap samples and, after centering, multiply just one of them by the values of r in the grid.

	Sample size							
Covariance matrix	$n = 20$				$n = 30$			
	mean	medi	perc	max	mean	medi	perc	max
$r_1 = r_2 = 1$	.040	.044	.040	.041	.043	.044	.041	.043
$r_1 = r_2 = 1.2$	.119	.114	.116	.119	.177	.170	.179	.177
$r_1 = 2$, $r_2 = 1.2$	.845	.815	.845	.849	.964	.954	.970	.974
$r_1 = r_2 = 2$	.930	.914	.938	.939	.994	.990	.997	.996

Table 5.5: Rates of rejections for 3 samples in 5,000 simulations, using the random Tukey depth, for the considered methods to choose k, distributions, sample sizes and values of r. The dimension is $p = 2$. The significance level is .05.

At the end of the bootstrap step, we have tried the same four procedures to select k as in the previous case, also obtaining here similar rejection rates for the four cases. These

rejection rates are displayed in Table 5.5.

In order to compare the random Tukey depth with the Tukey depth, in Table 5.6 we present the rejection rates obtained when the precise value of k is selected with the 80% percentile procedure and, in brackets, the corresponding value of the Tukey depth. Note that the rejection rates using D_T of Table 5.6 have been taken directly from Liu and Singh [57].

Covariance matrix	Sample size			
	$n = 20$		$n = 30$	
$r_1 = r_2 = 1$	.04	(.04)	.04	(.04)
$r_1 = r_2 = 1.2$	.12	(.13)	.18	(.18)
$r_1 = 2,\ r_2 = 1.2$	.85	(.85)	.97	(.98)
$r_1 = r_2 = 2$	.94	(.94)	1	(.99)

Table 5.6: Rates of rejections in 5,000 simulations using $D_{T,k}$ (between parentheses the rate with D_T) to test the homogeneity in three samples of Gaussian distributions with independent, identically distributed marginals and the exposed values of r. The dimension is $p = 2$. The significance level is .05.

In addition, we have computed the medians of the selected numbers of vectors used in each of the four procedures for each of the covariances and sample sizes. These are displayed in Table 5.7.

The results of both studies in this subsection are quite encouraging because there are no important differences between the rejection rates with the two depths despite the comparatively low number of directions used to compute the random Tukey depth.

Covariance matrix	Sample size							
	$n = 20$				$n = 30$			
	mean	medi	perc	max	mean	medi	perc	max
$r_1 = r_2 = 1$	4	2	5	23	4	2	4	22
$r_1 = r_2 = 1.2$	4	2	5	23	4	2	4	22
$r_1 = 2$, $r_2 = 1.2$	4	2	5	23	3	2	4	22
$r_1 = r_2 = 2$	4	2	5	23	3	2	4	22

Table 5.7: Medians of the number of employed random vectors in each of the cases of Table 5.5.

5.3 Functional random Tukey depth. Functional classification

In this section, we deal with an application of the functional random Tukey depth. Here, we will select the number of random directions to employ using cross-validation.

In this setting, we have an additional problem. In the finite dimensional case, it seems reasonable to choose the random directions using the uniform distribution on the sphere because of its invariance properties. Regrettably, in infinite dimensional spaces, there is no distribution with such good properties, making the selection of the random directions more arduous. One interesting possibility is to choose the distribution depending on the problem. This way a problem-specific procedure would be designed to select a distribution with some optimality properties. In the subsection which follows, some predictions of the results that could be obtained with a complete development of the theory are proposed, since the work required to do that exceeds the limits of this thesis. Here, we have taken first ν to be equal to the distribution of the standard Brownian motion. Then, we have tried some modifications of this distribution, which have improved the behavior of the procedure. Moreover, since those modifications are, in fact, parameter-dependent, we have chosen the values of the parameters with cross-validation. Some related results on the selection of referential measures in functional spaces appear in Ferraty and Vieu [28].

5.3.1 Application to classification

Here we study a real example. Our aim is to compare the random Tukey depth with some other functional depths in a practical situation. The situation we have chosen is a supervised classification problem which was carried out in López-Pintado and Romo [60]. In this paper, the authors analyze a data set consisting of the growth curves of a sample of 39 boys and 54 girls, the aim being to classify them, by sex, using just this information.

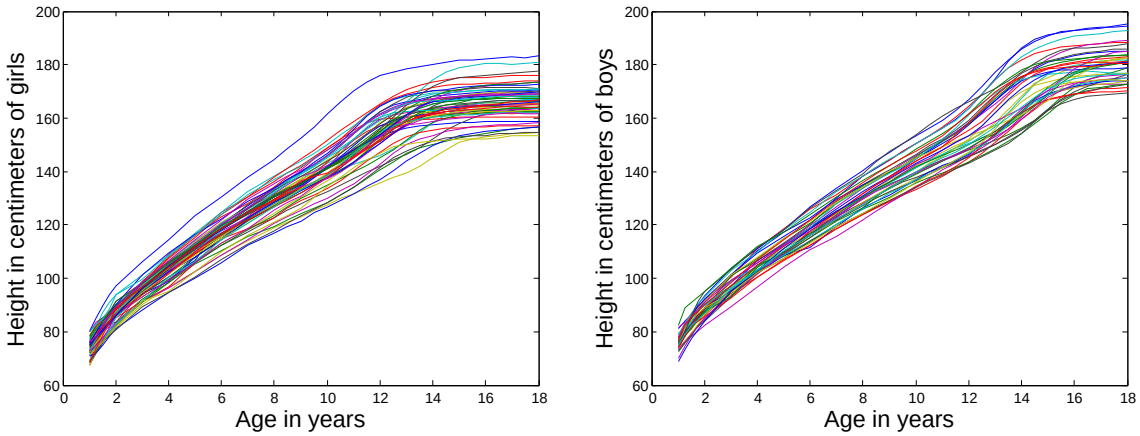

Figure 5.3: Growth curves of 54 girls (left-hand side) and 39 boys (right-hand side) measured 31 times each between 1 and 18 years of age.

Heights were measured in centimeters 31 times in the period from one to eighteen years. In the period from one to two years, the measures were taken each three months, in the period from three to seven years one time a year and, finally, in the period from eight to eighteen years two times a year. The data are in the file growth.zip, downloaded from ftp://ego.psych.mcgill.ca/pub/ramsay/FDAfuns/Matlab. In this web-page can also be found some notes that make use of the data. Those notes were designed to accompany the books by Ramsay and Silverman [72, 73]. In addition, these data are used in the recent book by Ramsay et al. [74]. We represent the data in Figure 5.3.

We are mainly interested in comparing the random Tukey depth in the functional setting with other functional depths. Thus, in the first part we will follow as closely as possible the steps in López-Pintado and Romo [60], except that we will replace the depth that those authors employ with the random Tukey depth.

After this, we have compared our results with those obtained with two other procedures not based on depths. The k nearest neighbors, k-NN, (see, for instance, Biau et al. [7]) and a kernel procedure (see Abraham et al. [1] and Ferraty and Vieu [29]). The idea for the kernel procedure is to consider a new random variable $Z \in \{0, 1\}$ which contains the group to which the observation belongs. Thus, if x_0 is the observed curve, it is possible to apply a kernel method to estimate the conditional probability $\mathbb{P}[Z = i/X = x_0]$ for $i = 0, 1$ and, then, classify the observation in the group in which this probability is highest. We have employed two kernels: the first kernel tried is $K(u) = 1_{[0,1]}(u)$ and the second is the quadratic one, $K(u) = (1 - u^2)1_{[0,1]}(u)$.

Furthermore, since we only have 31 observations for each individual, we can also consider the data as multidimensional ones and so, it is of some interest to make a comparison with a multidimensional classification procedure. To this end, we have used the random forests procedure. This is a procedure which is a combination of tree predictors in which each tree is determined by the values of a random vector which has the same distribution for all the trees in the forest (see Breiman [9] for more details).

It is well known that when handling this kind of data, it is useful to consider not only the growth curve but also accelerations of height (see, for instance, Ramsay and Silverman [72]). However, we only consider here the growth curves, as did López-Pintado and Romo [60], because our main interest is to compare our results with those obtained by them.

Through the following, we first summarize the classification procedures using depths

that we will use; secondly, we see how to choose the distribution and number of vectors when working with the random Tukey depth; then we see how to handle the problem in practice when reproducing López-Pintado and Romo [60]; and finally we compare the results obtained with the random Tukey depth with the other classification procedures.

Classification procedures using depths

Thus, let us assume that we have two samples $X_1, ..., X_n$ and $Y_1, ..., Y_m$ in a separable Hilbert space, $\mathbb{H}$ selected from two populations and that we are interested in classifying another curve $Z \in \mathbb{H}$ in one of those groups using a depth D to be chosen later. Three classification methods are proposed in López-Pintado and Romo [60]. They are

1. Distance to the trimmed mean ($\mathrm{M}_{\alpha,\beta}$)

   For this method, we first compute the depths of the points in the sample $X_1, ..., X_n$ with respect to their empirical distribution and choose $\alpha \in [0, 1)$. Then, the α-trimmed mean of this sample, $\mu_\alpha(X)$, is the mean of the $n \times (1 - \alpha)$ deepest points. Given $\beta \in [0, 1)$, compute similarly $\mu_\beta(Y)$, the β-trimmed mean of the sample $Y_1, ..., Y_m$.

   Now, we classify Z in the first group if

   $$\|Z - \mu_\alpha(X)\| < \|Z - \mu_\beta(Y)\|.$$

   Otherwise, we classify Z in the second group.

2. Weighted average distance (AM)

   In some sense, in method M, each group is represented by its trimmed mean. Here, we compute the distance between Z and the group as a weighted mean of the distances between Z and the members of the group where the weights are the depths of the points.

   Thus, we classify the function Z in the first group if

   $$\frac{\sum_{i=1}^{n} \|Z - X_i\| D_X(X_i)}{\sum_{i=1}^{n} D_X(X_i)} < \frac{\sum_{j=1}^{m} \|Z - Y_j\| D_Y(Y_j)}{\sum_{j=1}^{m} D_Y(Y_j)}, \tag{5.2}$$

where the subscripts in D_X and D_Y mean that the depths are computed with respect to the empirical distribution associated to the corresponding sample.

3. Trimmed weighted average distance (TAM)

   In the AM method, the result of the classification could be affected by the number of elements in each sample if $n \neq m$. The solution for this consists in taking a third value $l \leq \min(n, m)$ and replacing (5.2) by

$$\frac{\sum_{i=1}^{l} \|Z - X_{(i)}\| D_X(X_{(i)})}{\sum_{i=1}^{l} D_X(X_{(i)})} < \frac{\sum_{i=1}^{l} \|Z - Y_{(i)}\| D_Y(Y_{(i)})}{\sum_{i=1}^{l} D_Y(Y_{(i)})},$$

   where $X_{(1)}$ is the deepest point in the X-sample, $X_{(2)}$ is the second deepest point in the X-sample, ... and similarly for the Y-sample. When handling this procedure, $l = \min(n, m)$ in López-Pintado and Romo [60] is applied.

We have also included two additional procedures not considered in López-Pintado and Romo [60]. The first one was proposed in Ghosh and Chaudhuri [37] and first used in the functional case in Cuevas et al. [23].

4. Maximum Depth (MD)

   It consists, simply, of adding the observation Z to the two training samples, computing its depth in each of the two new samples and classifying Z in the group in which its depth is greater.

The MD procedure performs well only if the two populations differ in location and the prior probabilities are equal, as previously recognized by Ghosh and Chaudhuri [37]. In order to alleviate this problem, the following procedure is proposed in Li et al. [49].

5. DD-plot ratio (DD)

   Let us denote $\hat{X} := \{X_1, ..., X_n, Z\}$ and $\hat{Y} := \{Y_1, ..., Y_n, Z\}$. We first compute the r which minimizes

$$\sum_{i=1}^{n} I_{\{D_{\hat{Y}}(X_i) > r D_{\hat{X}}(X_i)\}} + \sum_{i=1}^{m} I_{\{D_{\hat{Y}}(Y_i) < r D_{\hat{X}}(Y_i)\}}$$

   and classify the function Z in the first group if $D_{\hat{Y}}(Z) < r D_{\hat{X}}(Z)$.

We thank R. Liu for the suggestion of using $D_{\hat{X}}$ instead of D_X which, in practice, gives a smaller classifying error.

Distribution and number of vectors for the random Tukey depth

Regarding the selection of the distribution ν used to select the directions to project, we have tried several procedures. The first is to choose ν as the distribution of the standard Brownian motion. The remaining possibilities are intended to take advantage of the differences which appear among the training samples. To do this, we first compute the functions containing the point-wise medians of the trajectories in both training samples. That is, for every $t \in [1, 18]$ we compute

$$m_X(t) := \text{median}\{X_1(t), ..., X_n(t)\}, \text{ and } m_Y(t) := \text{median}\{Y_1(t), ..., Y_m(t)\}.$$

Now, we take for ν the distribution of the solution of the following stochastic differential equation

$$S_{a,c}(0) = c \text{ and } dS_{a,c}(t) = |m_X(t) - m_Y(t)|^a dB(t),$$

where B is a standard Brownian motion. Here, we choose $a \in \{0, 1\}$. In the first case, the difference between the functions m_X and m_Y has no influence on ν. The constant c specifies the initial value for the solution. We have tried the values $c = 0, 1, 5$. The reason for introducing c is that the Brownian motion always starts at 0 and is continuous, thus erasing the differences in the early states of the process. Particularly, the distribution of $S_{0,0}$ is the standard Brownian motion.

Given a, c, to simulate the random trajectories and bearing in mind the times in which the heights were measured, we have taken $t_i \in [1, 18]$, $i = 1, ..., 31$ such that

- $t_i = 3/4 + i/4$ for $i = 1, ..., 5$,
- $t_i = i - 3$ for $i = 6, ..., 10$,
- $t_i = 2.5 + i/2$ for $i = 11, ..., 31$.

Then we have defined

$$\begin{aligned} S_{a,c}(t_1) &= c \\ S_{a,c}(t_i) &= S_{a,c}(t_{i-1}) + |m_X(t_i) - m_Y(t_i)|^a Z_i,\ i = 2, ..., 31, \end{aligned}$$

where Z_i, $i = 2, ..., 31$, are independent random variables with distribution $N(0, t_i - t_{i-1})$.

Concerning k, the simulations in Section 5.1 suggest that high values for k are not required. The following results have been obtained by selecting $k \in \{1, ..., 100\}$. Although the length of the interval might be considered too low, we have repeated the process replacing 100 by 1,000 and the results obtained have been similar.

The right values of k, a and c have been obtained by leave-one-out cross validation.

The problem in practice

In this section, we compare our depth with those proposed in López-Pintado and Romo [60]. To do this, we have repeated the study made there with three differences:

1. Most importantly, we have replaced the functional depths handled there with the random Tukey depth.
2. In López-Pintado and Romo [60], the authors consider the curves as elements in $L^1[0, 1]$, which is not possible here, because we need a separable Hilbert space. We take $\mathbb{H} = L^2[0, 1]$.
3. In López-Pintado and Romo [60], the authors smoothed the original data using a spline basis. We have skipped this step because it is not necessary for our method.

Regarding item 2., remember that the heights were measured 31 times in the interval $[1, 18]$. Therefore, first, we need to modify the time in order to transform this interval to

$[0,1]$ and, then, we can employ properties of the Rieman integral to make the approximation

$$< X, s_{a,c} > = \int_0^1 X\,(17u+1)\, s_{a,c}(17u+1)du \approx \sum_{i=1}^{31} X(t_i) s_{a,c}(t_i)\Delta_i,$$

where $s_{a,c}$ is drawn with distribution $S_{a,c}$ and Δ_i denotes the length of the interval associated to the point t_i. Concerning those intervals, if $i = 2, \ldots, 30$, then the observation $X(t_i)$ works for the interval $((t_i + t_{i-1})/2, (t_{i+1} + t_i)/2)$. Taking into account that, in the last part, the measurements were taken every half a year, we can assume that $X(t_{31})$ is valid for the period $(17.5, 18.5)$. Finally, it seems safest to assume that the $X(t_1)$ is not good to represent previous heights. Then, if we define $t_0 = 1$ and $t_{32} = 18.5$ we take

$$\Delta_i \;\; = \;\; (t_{i+1} - t_{i-1})/35, \;\; i = 1, ..., 31.$$

In López-Pintado and Romo [60], the authors consider three possibilities to split the sample into training and validation sets. For the sake of brevity, we split the sample using only leave-one-out cross-validation.

Let us briefly explain how the whole process works. Note that we have a sample of size 93. Therefore, we have repeated 100 times the following: for each observation in the sample, we consider the training sample composed of the remaining 92 observations. Then, we have generated at random 100 vectors with each of the distributions of the random variables $S_{a,c}$ for $a = 0, 1$ and $c = 0, 1, 5$, which gives 6 different samples of random directions with size 100 each.

Firstly, we have focused our attention on the $S_{0,0}$ distribution. Here we only have to select the value of k. As stated previously, this value is chosen by leave-one-out cross-validation applied to the remaining sample with 92 observations. From now on, this procedure is called $S_{0,0}$.

Moreover, we have applied the procedure, allowing variations in a and c. Here, we have chosen, also using leave-one-out cross-validation, the best combination of k, a and c.

From now on, this procedure is denoted by $S_{a,c}$. Note that in this case, it may occur that the chosen a and c satisfy $a = c = 0$.

The results of the comparison appear in Table 5.8, which include the obtained failure rates using the methods proposed in López-Pintado and Romo [60], $M_{\alpha,\beta}$, AM and TAM, when applied to the random Tukey depth and to their depths. We have chosen $\alpha = \beta = 0.2$ as done in López-Pintado and Romo [60]. The depths handled in López-Pintado and Romo [60] are the band depth determined by three different curves (DS3), (3.3), by four different curves (DS4), (3.3), and the generalized band depth (DGS), (3.4). Their error rates are contained in the last three columns of Table 5.8 and have been taken from Tables 1-3 in López-Pintado and Romo [60]. The previous two columns of Table 5.8 concern the random Tukey depth. The first includes the failure rates when using the procedure $S_{0,0}$ and the second one when using $S_{a,c}$. In this last case, a varies in $\{0, 1\}$ and c in $\{0, 1, 5\}$.

Classification	Random Tukey		Depths proposed in [60]		
method	$S_{0,0}$	$S_{a,c}$	DS3	DS4	DGS
$M_{\alpha,\beta}$	.1858	.1825	.1828	.1828	.1613
AM	.1403	.1368	.2473	.2473	.1935
TAM	.1542	.1430	.2436	.2436	.1690

Table 5.8: Rates of mistakes when classifying the growth curves by sex for the shown methods and depths.

According to Table 5.8, for the AM and TAM methods, the random Tukey depth provides better results than the depths used in López-Pintado and Romo [60] when we take the standard Brownian motion and even better when parameters a, c in $S_{a,c}$ are chosen with cross-validation. The medians of the number of random vectors used have been 1 for each of the three methods with $S_{0,0}$. In the case of $S_{a,c}$, the median of the number of random vectors has been 2 for the $M_{\alpha,\beta}$ method and 1 for any of the other two methods.

The conclusion is that the AM method works better than $M_{\alpha,\beta}$, and TAM. It is worth pointing out that TAM is worse than AM because AM uses some information that TAM does not.

One possibility which we have not pursued is to modify α and β in $M_{\alpha,\beta}$. The reason for this is that, in this section, we are mostly interested in comparing our depth with those proposed in López-Pintado and Romo [60].

Comparison with other classification procedures

In this section, we compare the DD method using the random Tukey depth with other non depth-based procedures. However, since, as far as we know, this is the first time that this method has been applied in the functional setting, we also provide the results from applying the MD method.

We combine these with the two methods we have presented to select the random directions: $S_{0,0}$ and $S_{a,c}$, where the values of a and c are chosen by cross-validation. The results appear in Table 5.8 and for MD and DD in Table 5.9. The median of the number of employed random directions is 20 for the MD method and 25 for the DD method when combined with $S_{0,0}$. It is 6 for the MD method and 25 for the DD method when combined with $S_{a,c}$.

Classification method	$S_{0,0}$	$S_{a,c}$
MD	.1317	.1141
DD	.1194	.0945

Table 5.9: Rates of mistakes when classifying the growth curves by sex for the shown methods and the random Tukey depth.

As expected, the DD method outperforms the MD, but the improvement is not dramatic. This fact suggests the interesting question, not pursued here, of when the differences in the curves are due only to a translation. Figure 5.4 suggests that the answer could be affirmative. This figure contains the curves of both sexes, girls in red and boys in blue, after centering each of the groups by its coordinate-wise median.

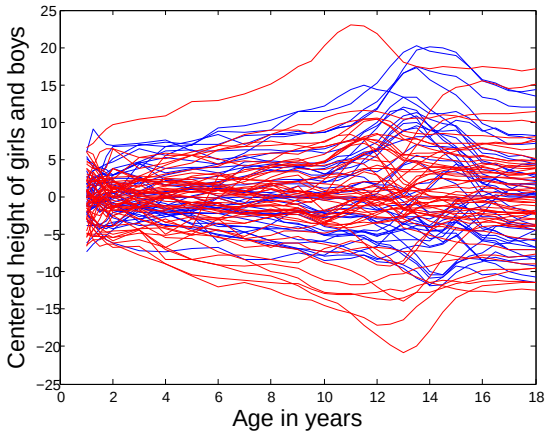

Figure 5.4: Growth curves of 54 girls (red) centered by its coordinate-wise median and of 39 boys (blue) also centered by its coordinate-wise median.

The next step is to classify the same data using the random forests, the k-NN and the kernel methods. Regarding the random forests, we have employed it with 100 trees. Concerning the k-NN method, we have used three possibilities for selecting the number of nearest neighbors. The first two have consisted in fixing $k = 1$ and 3, respectively. In the third, we have chosen k between $1, 3, ..., 91$ with leave-one-out cross-validation. The value chosen in the last possibility was always 3. This explains that the rates of failures were the same in the second and third case (see Table 5.10). Finally, we have applied a kernel method with the indicator of the interval $[0, 1]$ and the quadratic kernel. The selection of the window was accomplished as follows: given the training sample $X_1, ..., X_{92}$, we have

considered the values

$$\begin{aligned} h_m &= \min\{\|X_i - X_j\|,\ i \neq j, i, j = 1, ..., 92\}, \\ h_M &= \max\{\|X_i - X_j\|,\ i, j = 1, ..., 92\}. \end{aligned}$$

We have chosen the window applying leave-one-out cross-validation to the grid of values $h_m + i(h_M - h_m)/50, i = 0, 1, ..., 50$.

Random Forests	k-NN			Kernel	
	1-NN	3-NN	cross-val.	Indicator	Quadratic
.0968	.0753	.0323	.0323	.0645	.0430

Table 5.10: Rates of mistakes when classifying the growth curves by sex using cross validation for the shown methods.

The rates of mistakes appear in Table 5.10. Note that the rate of mistakes of both random Tukey procedures are well above those obtained with the k-NN and the kernel method, but they are similar to that obtained with random forests. However, the improvement which appears between the first and the second column in Table 5.9 makes us relatively optimistic about the results which could be obtained if an optimal procedure to select the distribution ν were applied.

Chapter 6

Test of Gaussianity for stationary processes

In this chapter, we address the statistical problem of testing whether a stationary process is Gaussian. The observation consists in a sample of a path of the process. Using the random projection technique introduced and studied in Cuesta-Albertos et al. [12] in the frame of goodness of fit test for functional data, we develop some statistical tests. The main idea is to test the Gaussianity of the marginal distribution of some random linear combinations of the process. This leads to consistent decision rules which analyze the whole distribution of the process and not only its marginal distribution at a fixed order as other proposed procedures do. Some numerical simulations show the pertinence of our approach.

The first section introduces the procedure proposed. It has several subsections, devoted to explaining the procedure and its application with the Epps and the Lobato and Velasco tests. Section 6.2 explains how to handle the test in practice. Finally, in Section 6.3 we study some simulations, where the alternatives include both processes with Gaussian and non-Gaussian one-dimensional marginal. This section ends with a short comment on the effect of increasing the number of projections.

The notation and definitions used here were stated in Chapter 3.

6.1 The procedure

In this section we present a universal test to verify whether the distribution of a stationary process is Gaussian. Thus, given $\mathbf{X} := (X_t)_{t\in\mathbb{Z}}$, a stationary process of real-valued random variables we are interested in constructing a test for the null hypothesis

H_0 : $\mathbf{X}$ is Gaussian

against the alternative

H_a : $\mathbf{X}$ is not Gaussian.

Note that H_0 holds if, and only if $(X_1, \ldots, X_t)^T$ is a Gaussian vector, for all $t \in \mathbb{N}$. As $\mathbf{X}$ is stationary, this is equivalent to the distribution of $(X_t)_{t\leq 0}$ is Gaussian. In addition, it is the same as the Gaussianity of the process $\mathbf{X}^{(t)} := (X_j)_{j\leq t}$, for any $t \in \mathbb{Z}$. To check whether $\mathbf{X}^{(t)}$ is Gaussian, we only need to

- include $\mathbf{X}^{(t)}$ in an appropriate Hilbert space,
- select a vector $\mathbf{h}$ using a dissipative distribution (see Definition 3.2.3),
- compute the scalar product $\langle \mathbf{X}^{(t)}, \mathbf{h}\rangle$,
- check if $\langle \mathbf{X}^{(t)}, \mathbf{h}\rangle$ is Gaussian,

since, according to Theorem 3.2.4, almost surely, $\mathbf{X}^{(t)}$ is Gaussian if, and only if, $\langle \mathbf{X}^{(t)}, \mathbf{h}\rangle$ is Gaussian.

6.1.1 The Hilbert space

Concerning the Hilbert space in which the process is included, let us consider the space of sequences

$$l^2 = \left\{ (x_n)_{n\in\mathbb{N}} : \sum_{n\in\mathbb{N}} x_n^2 a_n < \infty \right\},$$

with $a_0 := 1$ and $a_n = n^{-2}, n \geq 1$, endowed with the scalar product

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{n \in \mathbb{N}} x_n y_n a_n, \text{ where } \mathbf{x} = (x_n)_{n \in \mathbb{N}}, \mathbf{y} = (y_n)_{n \in \mathbb{N}}. \tag{6.1}$$

If $\mathbf{X}$ is a stationary process and the variance of X_t is finite, then $E[\sum_{n \in \mathbb{N}} X_{t-n}^2 a_n] < \infty$ and, in consequence $\sum_{n \in \mathbb{N}} X_{t-n}^2 a_n$ is almost surely finite. Thus, almost surely, $\mathbf{X}^{(t)} \in l^2$. Furthermore, obviously the Gaussianity in this space is equivalent to the (usual sense) Gaussianity of $\mathbf{X}^{(t)}$.

6.1.2 The distribution

Now, we need a dissipative distribution on l^2. We will use the so-called Dirichlet distribution (see Pitman [70]) and build it using the so-called stick-breaking method. That is, let $\alpha_1, \alpha_2 > 0$. Then, we choose $(\beta_n)_{n \in \mathbb{N}}$ independent and identically distributed with beta distribution of parameters α_1 and α_2. Further, we consider the probability distribution which selects a random point in l^2 according to the following iterative procedure:

- $l_0 = \beta_0 \in [0, 1]$.
- Given $n \geq 1$, $l_n \in [0, 1 - \sum_{i=0}^{n-1} l_i]$ equal to $\beta_n(1 - \sum_{i=0}^{n-1} l_i)$.

Let us define $H_n = (l_n/a_n)^{1/2}$ for $n \in \mathbb{N}$ and take $\mathbf{H} = (H_n)_{n \in \mathbb{N}}$. It can easily be checked that the distribution of $\mathbf{H}$ is dissipative (see Definition 3.2.3). Moreover, $\mathbf{H} \in l^2$ almost surely because, as shown in Proposition 6.1.1, $\|\mathbf{H}\| = 1$, almost surely.

Proposition 6.1.1. *Let $\mathbf{H} = (H_n)_{n \geq 0}$ be a stochastic process constructed as described above. Let $\alpha := \alpha_1/(\alpha_1 + \alpha_2)$ be the mean of the beta distribution of parameters α_1 and α_2. Then, we have that*

1. $\mathbb{E}[l_n] = \alpha(1 - \alpha)^n$, *for every* $n \in \mathbb{N}^*$.
2. $\|\mathbf{H}\| = 1$, *almost surely.*

Proof.

Obviously *1.* holds for $n = 0$. Thus, let us assume that *1.* is satisfied for $n \in \mathbb{N}$ and let us show that it also holds for $n+1$. By the construction of $(l_n)_{n\in\mathbb{N}}$, we have that if β_{n+1} is a random variable with beta distribution of parameters α_1 and α_2 independent of $(l_i)_{i\in[0,n]}$, then

$$\mathbb{E}[l_{n+1}] = \mathbb{E}[\beta_{n+1}]\left(1 - \sum_{i=0}^{n}\mathbb{E}[l_i]\right) = \alpha\left(1 - \sum_{i=0}^{n}\alpha(1-\alpha)^i\right) = \alpha(1-\alpha)^{n+1},$$

where the last equality comes from the application of the formula giving the sum of n numbers in a geometric progression.

Concerning *2.*, by using the scalar product given in (6.1), we have that

$$\|\mathbf{H}\| = \sum_{i=0}^{\infty} H_i^2 a_i = \sum_{i=0}^{\infty} l_i \leq 1, \tag{6.2}$$

because, by the construction of $(l_n)_{n\in\mathbb{N}}$, $\sum_{i=0}^{n} l_i \leq 1$, for every $n \in \mathbb{N}$. However, applying *1.*, we have that

$$\mathbb{E}\left[\|\mathbf{H}\|\right] = \sum_{i=0}^{\infty}\alpha(1-\alpha)^i = 1.$$

So that, by (6.2) we obtain *2.* □

6.1.3 The projection and its properties

Now, let $\mathbf{h} = (h_i)_{i\in\mathbb{N}}$ be a fixed realization of the random element $\mathbf{H}$, drawn independently from the process $\mathbf{X}$. Let us consider the process $\mathbf{Y} = (Y_t)_{t\in\mathbb{Z}}$ given by the projections of $(\mathbf{X}^{(t)})_{t\in\mathbb{Z}}$ on the one-dimensional subspace generated by $\mathbf{h}$, i.e.

$$Y_t = \sum_{i=0}^{\infty} h_i X_{t-i} a_i, t \in \mathbb{Z}. \tag{6.3}$$

As will be seen in Proposition 6.1.3, the properties of the process $\mathbf{X}$ are inherited by the process $\mathbf{Y}$. Moreover, according to Theorem 3.2.4, the Gaussianity of $\mathbf{X}$ can be assessed simply by verifying the one-dimensional marginal distributions of $\mathbf{Y}$. This can be done for instance with the Epps or Lobato and Velasco tests presented in Section 3.3 whenever $\mathbf{Y}$

satisfies the appropriate assumptions. Subsections 6.1.5 and 6.1.6 are devoted to this task.

We begin by proving Lemma 6.1.2 which is necessary for Proposition 6.1.3. Remember that $\gamma_X(t)$ denotes the autocovariance of order t and that for simplicity we write γ_X instead of $\gamma_X(0)$.

Lemma 6.1.2. *Let $\boldsymbol{X}$ be an ergodic and stationary process such that $\sum_{t=0}^{\infty}|\gamma_X(t)|<\infty$. If we select $\mathbf{H}$ as described above, then,*

1. *$\sum_{i=0}^{\infty} H_i a_i < \infty$ almost surely.*
2. *Almost surely, the random variable $L := \sum_{i,j=0}^{\infty} H_i H_j a_i a_j |X_{-i}-\mu_X||X_{t-j}-\mu_X|$ is conditionally integrable given $\mathbf{H}$.*

Proof.

1. This is straightforward since the Cauchy-Schwartz inequality gives that

$$\sum_{i=0}^{\infty} H_i a_i \leq \left(\sum_{i=0}^{\infty} l_i\right)^{1/2}\left(1+\sum_{i=1}^{\infty} 1/i^2\right)^{1/2} = \left(1+\sum_{i=1}^{\infty} 1/i^2\right)^{1/2} < \infty, \text{ almost surely,}$$

where the last equality comes from Proposition 6.1.1.

To prove *2.*, let $\mathbf{h}=(h_0,h_1,\ldots)$ be a fixed realization of $\mathbf{H}$. We have that

$$\begin{aligned}\mathbb{E}[L|\mathbf{h}] &= \sum_{i,j=0}^{\infty} h_i h_j a_i a_j \mathbb{E}[|X_{-i}-\mu_X||X_{t-j}-\mu_X|] \\ &\leq \sum_{i,j=0}^{\infty} h_i h_j a_i a_j (\mathbb{E}[(X_{-i}-\mu_X)^2])^{1/2}(\mathbb{E}[(X_{t-j}-\mu_X)^2])^{1/2} = \left(\sum_{i}^{\infty} h_i a_i\right)^2 \gamma_X,\end{aligned}$$

where we have used Hölder inequality. Thus, L is conditionally integrable thanks to *1.* and that $\gamma_X \leq \sum_{t=0}^{\infty}|\gamma_X(t)|<\infty$. □

In the sequel, $\gamma_{Y|\mathbf{h}}(t)$ denotes the conditional autocovariance of order t of Y given $\mathbf{h}$. That is, denoting by $\mu_{Y|\mathbf{h}}$, the conditional expectation of Y_0 given $\mathbf{h}$,

$$\gamma_{Y|\mathbf{h}}(t) := \mathbb{E}[(Y_0-\mu_{Y|\mathbf{h}})(Y_t-\mu_{Y|\mathbf{h}})|\mathbf{h}].$$

Proposition 6.1.3. *Let $(X_t)_{t\in\mathbb{Z}}$ be an ergodic and stationary process such that $\sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t)| < \infty$, with $\zeta \geq 0$. Then, conditionally on $\mathbf{h}$, the process $(Y_t)_{t\in\mathbb{Z}}$ defined in (6.3) is ergodic and stationary. In addition, $\mathbb{E}[|Y_0||\mathbf{h}]$ and $\sum_{t=0}^{\infty} t^{\zeta}|\gamma_{Y|\mathbf{h}}(t)|$ are finite.*

Proof.

Since $(X_t)_{t\in\mathbb{Z}}$ is a stationary ergodic process, conditionally on $\mathbf{h}$, $(Y_t)_{t\in\mathbb{Z}}$ is also a stationary ergodic process (see Doob [25, p. 458]).

Using the definition of the process $\mathbf{Y}$, we have

$$\mathbb{E}[|Y_0||\mathbf{h}] \leq \mathbb{E}\left[\sum_{i=0}^{\infty} h_i a_i |X_{-i}| \middle| \mathbf{h}\right] = \mathbb{E}[|X_0|]\sum_{i=0}^{\infty} h_i a_i < \infty, \quad \text{a.s.}$$

because of *1.* in Lemma 6.1.2.

By *2.* in Lemma 6.1.2, we have that

$$\gamma_{Y|\mathbf{h}}(t) = \mathbb{E}[\sum_{i,j=0}^{\infty} h_i h_j a_i a_j (X_{-i} - \mu_X)(X_{t-j} - \mu_X)]$$

exists. Thus, using the dominated convergence theorem, we obtain that

$$\gamma_{Y|\mathbf{h}}(t) = \sum_{i,j=0}^{\infty} h_i h_j a_i a_j \gamma_X(t - j + i)$$

and

$$\sum_{t=0}^{\infty} t^{\zeta}|\gamma_{Y|\mathbf{h}}(t)| \leq \sum_{i,j=0}^{\infty} h_i h_j a_i a_j \sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t - j + i)|.$$

Obviously, $\sum_{i,j=0}^{\infty} h_i h_j a_i a_j \sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t-j+i)| =: T_1 + T_2 + T_3$, where

- $T_1 = \sum_{j=0}^{\infty} h_j a_j \sum_{i=j}^{\infty} h_i a_i \sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t-j+i)|$,
- $T_2 = \sum_{j=0}^{\infty} h_j a_j \sum_{i=0}^{j-1} h_i a_i \sum_{t=2j+1}^{\infty} t^{\zeta}|\gamma_X(t-j+i)|$,
- $T_3 = \sum_{j=0}^{\infty} h_j a_j \sum_{i=0}^{j-1} h_i a_i \sum_{t=0}^{2j} t^{\zeta}|\gamma_X(t-j+i)|$.

If $i \geq j$, as $t \in \mathbb{N}$ and $\zeta \geq 0$, we have $t^{\zeta} \leq (t-j+i)^{\zeta}$. Thus,

$$T_1 \leq \sum_{j=0}^{\infty} h_j a_j \sum_{i=j}^{\infty} h_i a_i \sum_{t=0}^{\infty} (t-j+i)^{\zeta}|\gamma_X(t-j+i)| \leq \sum_{j=0}^{\infty} h_j a_j \sum_{i=j}^{\infty} h_i a_i \sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t)|,$$

because $t-j+i \geq t$. Then, due to $\sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t)| < \infty$ and *1.* in Lemma 6.1.2 we obtain $T_1 < \infty$.

Concerning T_2, as $j > i$ and $t-j+i > 0$, we can apply the $c_\zeta-$inequality (see Loève [59] p.157) to $t = (t-j+i)+(j-i)$ to obtain that there exists $c_\zeta > 0$ such that $t^{\zeta} \leq c_\zeta(t-j+i)^{\zeta} + c_\zeta(j-i)^{\zeta} \leq 2c_\zeta(t-j+i)^{\zeta}$. Thus,

$$T_2 \leq 2c_\zeta \sum_{j=0}^{\infty} h_j a_j \sum_{i=0}^{j-1} h_i a_i \sum_{t=2j+1}^{\infty} (t-j+i)^{\zeta}|\gamma_X(t-j+i)| \leq 2c_\zeta \sum_{j=0}^{\infty} h_j a_j \sum_{i=0}^{j-1} h_i a_i \sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t)|.$$

Then, using the same tricks as for T_1 we obtain that $T_2 < \infty$.
For T_3, the fact that $\sum_{t=0}^{\infty} t^{\zeta}|\gamma_X(t)| < \infty$, implies that there exists an $R > 0$ such that $|\gamma_X(t)| \leq R$ for all $t \in \mathbb{Z}$. Therefore,

$$T_3 \leq R\left(\sum_{i=0}^{\infty} h_i a_i\right) \sum_{j=0}^{\infty} h_j a_j (2j)^{\zeta}(2j+1) =: R\left(\sum_{i=0}^{\infty} h_i a_i\right) T_3^*.$$

By *1.* in Lemma 6.1.2, to show that $T_3 < \infty$, we only need to prove that $T_3^* < \infty$. Furthermore, applying the Jensen inequality and *1.* in Proposition 6.1.1, we have that

$$\mathbb{E}[T_3^*] \leq \sum_{j=0}^{\infty} a_j^{1/2}(2j)^{\zeta}(2j+1)\alpha^{1/2}(1-\alpha)^{j/2}. \tag{6.4}$$

This series is convergent ($\alpha \in (0,1)$). Hence, T_3^* is finite almost surely and the proof ends. □

6.1.4 Characterization of one-dimensional Gaussian distributions

The result we prove here will be used later in the section. Let us start by stating the definition of analytic characteristic function which has been taken from Laha and Rohatgi [48].

Definition 6.1.4. *A characteristic function Φ is said to be analytic if there exist*

- *a complex valued function, ϕ, of the complex variable z which is holomorphic in a circle $\{z : |z| < \rho\}$, where $\rho > 0$,*

- *a positive real number* δ *such that* $\Phi(t) = \phi(t)$, *for* $|t| < \delta$.

That is, an analytic characteristic function is a characteristic function which coincides with a holomorphic function in some neighborhood of zero.

Some properties of analytic characteristic functions may be found in Laha and Rohatgi [48]. In particular, it is proved therein that the characteristic function of a Gaussian distribution is analytic (this is a well-known fact). Some other well-known distributions having analytic characteristic function are the binomial, Poisson and gamma distributions but not the Cauchy one.

The following result will be useful to assess that our goodness of fit test will work with all non-Gaussian alternatives.

Proposition 6.1.5. *Let* P *be a Borel probability measure defined on* $\mathbb{R}$. *Assume that* P *is absolutely continuous with respect to the Lebesgue measure. Let* Y *be a random variable having an analytic characteristic function* Φ_Y.

Then, Y *is Gaussian if, and only if,*

$$\exists m \in \mathbb{R},\ \exists s \in \mathbb{R}^+ \text{ such that } P(\{y \in \mathbb{R} : \Phi_Y(y) = \Phi_{m,s}(y)\}) > 0. \tag{6.5}$$

Proof.

The necessary part is obvious. Let us prove the sufficiency. As Y satisfies (6.5), and P is absolutely continuous, we have that the set $R := \{y \in \mathbb{R} : \Phi_Y(y) = \Phi_{m,s}(y)\}$ is infinite and not denumerable. Thus, it contains at least one accumulation point.

Furthermore, the function $y \to \Phi_Y(y) - \Phi_{m,s}(y)$ is analytic, and it vanishes on R. Therefore, this function has a non-isolated zero but the only analytical function with at least one non-isolated zero is the null function (see for example Rudin [75]) which proves the result. □

Proposition 6.1.5 may be seen as a spectral counterpart of Theorem 3.2.4.

6.1.5 Conditions for applying the Epps test

In this subsection, we analyze the theoretical behavior of the random projection procedure when using the Epps test. That is, we analyze the behavior of Epps test when applied to the randomly projected process (see Theorem 6.1.9). Moreover, in a corollary (Corollary 6.1.10) we will show that if the values in λ are drawn randomly, then the Epps test is consistent against many more alternatives.

Let us first state Lemma 6.1.6 that gives the consistency for the estimator of the spectral density function at zero, defined in (3.6). Let us denote by $k_{lmno}(q, r, q+r; \lambda)$ the fourth-order cumulant of $Z_{0,l}$, $Z_{q,m}$, $Z_{r,n}$, and $Z_{q+r,o}$, where, for instance, $Z_{q,m}$ is the m-th component of the vector $g(Y_q, \lambda) - g_{\mu_Y,\gamma_Y}(\lambda)$ (see Subsection 3.3.2).

Lemma 6.1.6. *Let $\lambda \in \Lambda_N$. If $\boldsymbol{Y}$ is a stationary process such that*

$$\sup_{-\infty<q<\infty} \sum_{r=-\infty}^{\infty} |k_{lmno}(q, r, q+r; \lambda)| < \infty, \text{ for each } l, m, n, o \in \{1, ..., N\}, \tag{6.6}$$

then, $\hat{f}(0, \lambda) \to f_{\mathbf{Y}}(0, (\mu_Y, \gamma_Y), \lambda)$ almost surely.

Proof.
This is straightforward from the proof of Lemma 2.2 in Epps [26] but substituting by (6.6) the use of (3.5) and the Gebelein inequality (Gebelein [35]) for Gaussian processes. The Gebelein inequality states that the autocovariance of a multidimensional process is smaller than or equal to the product of variances of the marginals. □

Lemma 3.1 in Epps [26] proves that if $\mathbf{Y}$ is a stationary Gaussian process that satisfies (3.5), then (6.6) holds. In Mielniczuk [64], the Gebelein inequality is extended to two-dimensional vectorial processes with diagonal densities. Thus, any stationary process that satisfies (3.5) and whose two-dimensional marginal has diagonal density, also satisfies (6.6).

Let Θ be an open and bounded subset of $\mathbb{R} \times \mathbb{R}^+$. In Epps [26], it is proved that **H1** and **H2** (see Subsection 3.3.2) are satisfied if λ_i is equal to a rational number times λ_1, $i = 2, ..., N$. Now, thanks to Lemma 6.1.7 below, we have that λ can be taken at random and still fulfill **H1** and **H2**.

Lemma 6.1.7. *Assume that $\lambda = (\lambda_1, \ldots, \lambda_N)^T \in \Lambda_N$ $(N > 1)$ is drawn randomly with distribution P_λ having the following properties. First, P_λ is such that λ_1 and λ_2 are independent and identically distributed and have a density. Further, for $N > 2$, λ_i is a rational number times λ_1. Then, **H1** and **H2** are fulfilled almost surely.*

Proof.
Proceeding as in Epps [26] we have that

$$\Theta_0(\lambda) \subseteq \{(\nu, \gamma_Y) : \nu\lambda_1 = \mu_Y\lambda_1 + 2\pi k \text{ and } \nu\lambda_2 = \mu_Y\lambda_2 + 2\pi k^*, \text{ with } k, k^* \in \mathbb{Z}\}.$$

Now, in order to get that the cardinal of $\Theta_0(\lambda)$ is larger than one, we need λ_2 to be equal to a rational number times λ_1. However, this happens with probability zero and so, with probability one $\Theta_0(\lambda) \subseteq \{(\mu_Y, \gamma_Y)\}$. Thus, **H1** and **H2** follow directly. □

Note that in case $N > 1$, Lemma 6.1.7 remains valid if we draw independently at random λ_i, $i = 3, ..., N$. In addition, thanks to this lemma, we have the following corollary of Theorem 3.3.1.

Corollary 6.1.8. *Let $(Y_t)_{t\in\mathbb{Z}}$ be a stationary Gaussian process which satisfies (3.5) and λ be as in Lemma 6.1.7. Let (μ_n, γ_n) be the minimizer on Θ of the map*

$$(\nu, \rho) \to Q_n(\nu, \rho, \lambda)$$

nearest to $(\hat{\mu}, \hat{\gamma})$. If we assume that $f_{\mathbf{Y}}(0, (\mu_Y, \gamma_Y), \lambda)$ is positive definite, then $nQ_n(\mu_n, \gamma_n, \lambda)$ converges in distribution to χ^2_{2N-2}.

In the next theorem, the function Q_n also depends on the random $\mathbf{h}$. However, for the sake of simplicity, we have not expressed this dependence in the notation.

Theorem 6.1.9. *Let $\boldsymbol{X}$ be an ergodic stationary process satisfying (3.5). Draw respectively λ as in Lemma 6.1.7 and $\boldsymbol{h}$ independently of λ using $P_{\mathbf{H}}$ (as described in Section 6.1.2).*

Assume that, conditionally on $\mathbf{h}$, $\mathbf{Y}$ defined in (6.3) satisfies (6.6), that the characteristic function of its one-dimensional marginal is analytic and that $f_{\mathbf{Y}|\mathbf{h}}(0,(\mu_{Y|\mathbf{h}},\gamma_{Y|\mathbf{h}}),\lambda)$ exists and is positive definite for almost every $\mathbf{h}$. Let $Q_n(\cdot,\cdot,\lambda)$ be the quadratic form defined in (3.7) applied to $\boldsymbol{Y}$ and (μ_n,γ_n) its minimizer on Θ nearest to $(\hat{\mu}_{Y|\mathbf{h}},\hat{\gamma}_{Y|\mathbf{h}})$. Let further $A := \{(\lambda,h) : nQ_n(\mu_n,\gamma_n,\lambda) \to_d$ a non-degenerated distribution$\}$.

Then, $\boldsymbol{X}$ is Gaussian if, and only if, $(P_\lambda \otimes P_{\mathbf{H}})[A] > 0$.

Proof.

The necessary part is obvious, because if $\mathbf{X}$ is Gaussian, then $\mathbf{Y}$ also is Gaussian and Proposition 6.1.3 implies that $\mathbf{Y}$ satisfies the assumptions of Corollary 6.1.8.

Let us prove the sufficient part. As $(P_\lambda \otimes P_{\mathbf{H}})[A] > 0$ we have that there exist $\mathbf{h}$ and λ with $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$ such that $nQ_n(\mu_n,\gamma_n,\lambda)$ converges in law to a non-degenerated distribution. Therefore,

$$Q_n(\mu_n,\gamma_n,\lambda) \to \text{c.p. } 0. \tag{6.7}$$

In addition, we may assume without loss of generality that

$$\Phi_{Y_0}(\lambda_1) \neq 0 \text{ and } \Phi_{Y_0}(\lambda_2) \neq 0,$$

because, as Φ_{Y_0} is an analytic characteristic function, it has only isolated zeros.

By Lemma 6.1.6, $\hat{f}(0,\lambda)$ converges to $f_{\mathbf{Y}|\mathbf{h}}(0,(\mu_{Y|\mathbf{h}},\gamma_{Y|\mathbf{h}}),\lambda)$. Thus, $\lim_n G_n^+$ is positive definite because it is the inverse of $2\pi f_{\mathbf{Y}|\mathbf{h}}(0,(\mu_{Y|\mathbf{h}},\gamma_{Y|\mathbf{h}}),\lambda)$. This, together with (6.7), and (3.7) gives that

$$\hat{g}(\lambda) - g_{\mu_n,\gamma_n}(\lambda) \to \text{c.p. } 0. \tag{6.8}$$

Since $\mathbf{X}$ is an ergodic stationary process, by Doob [25, p. 458] we have that $(g(Y_t,\lambda))_{t\in\mathbb{Z}}$ is also an ergodic stationary process. Thus, as $\mathbb{E}|\cos(\lambda_i Y_0)| < \infty$ and $\mathbb{E}|\sin(\lambda_i Y_0)| < \infty$

for all $i = 1, ..., N$, we have by Theorem 2 in Hannan [40, Chap. IV] that

$$\hat{g}(\lambda) \to_{\text{c.p.}} \mathbb{E}[g(Y_0, \lambda)].$$

From this and (6.8), we can conclude that $\Phi_{\mu_n,\gamma_n}(\lambda_i)$ converges in probability to $\Phi_{Y_0}(\lambda_i)$ $(i = 1, ..., N)$.

Let us see how this implies that the sequence $\{(\mu_n, \gamma_n)\}_{n\in\mathbb{N}}$ converges. We have that

$$\lim_{n\to\infty} |\Phi_{\mu_n,\gamma_n}(\lambda_1)| = \lim_{n\to\infty} e^{-\lambda_1^2\gamma_n/2} = |\Phi_{Y_0}(\lambda_1)|, \text{ in probability,}$$

and, since $\lambda_1 \neq 0$ and $\Phi_{Y_0}(\lambda_1) \neq 0$, this implies that there exists $s \in \mathbb{R}$ such that $s = \lim_{n\to\infty} \gamma_n$ in probability. Note that there exists $\theta \in [0, 2\pi)$ such that

$$\Phi_{Y_0}(\lambda_1) = |\Phi_{Y_0}(\lambda_1)| \exp(i\theta).$$

As $\lambda_1 \neq 0$, if we take $m := \theta/\lambda_1$, then, we have that $\Phi_{Y_0}(\lambda_1) = \Phi_{m,s}(\lambda_1)$.

Analogously, we have that $|\Phi_{Y_0}(\lambda_2)| = \lim_{n\to\infty} e^{-\lambda_2^2\gamma_n/2}$, in probability, and as $s = \lim_{n\to\infty} \gamma_n$ we obtain

$$|\Phi_{Y_0}(\lambda_2)| = e^{-\lambda_2^2 s/2}. \tag{6.9}$$

Denoting $r = \lambda_2/\lambda_1$, we obtain that

$$\frac{\Phi_{Y_0}(\lambda_2)}{|\Phi_{Y_0}(\lambda_2)|} = \lim_n e^{ir\lambda_1\mu_n} = \left(\frac{\Phi_{Y_0}(\lambda_1)}{|\Phi_{Y_0}(\lambda_1)|}\right)^r = e^{ir\lambda_1 m}.$$

Together with (6.9), this gives $\Phi_{Y_0}(\lambda_2) = \Phi_{m,s}(\lambda_2)$.

As λ_2 was drawn independently from λ_1 with a distribution absolutely continuous with respect to the Lebesgue measure and as Φ_{Y_0} is analytic, by Proposition 6.1.5 we get that Y_0 is Gaussian. Then, by Theorem 3.2.4, we obtain that the process $\mathbf{X}$ is Gaussian. □

Remark 6.1.9.1. It is only necessary to assume that $\mathbf{X}$ is ergodic to prove the inverse part in Theorem 6.1.9 since every stationary Gaussian process which satisfies (3.5) is ergodic (see, Doob [25, p. 494] and Grenander [38, p. 44]).

Applying the arguments of Theorem 6.1.9 directly to the process $\mathbf{X}$, we obtain the following corollary. This provides a modification of the Epps test with better consistency properties.

Corollary 6.1.10. *Let $\boldsymbol{X}$ be an ergodic stationary process. Assume that the characteristic function of its one-dimensional marginal is analytic. Assume further that (3.5) holds. Let us take λ as in Lemma 6.1.7, $Q_n(\cdot,\cdot,\lambda)$ as in (3.7), let (μ_n,γ_n) be its minimizer on Θ nearest to $(\hat{\mu}_X,\hat{\gamma}_X)$ and*

$$B := \{\lambda : nQ_n(\mu_n,\gamma_n,\lambda) \to_d \text{ a non-degenerated distribution}\}.$$

If we assume that $f_{\mathbf{X}}(0,(\mu_X,\gamma_X),\lambda)$ exists and is positive definite, then, $\boldsymbol{X}$ is Gaussian if, and only if, $P_\lambda(B) > 0$.

The remark below can obviously be deduced from Theorems 3.3.1 and 6.1.9. This remark allows a test to be performed based on the asymptotic distribution of $nQ_n(\mu_n,\gamma_n,\lambda)$.

Remark 6.1.10.1. Theorem 6.1.9 and Corollary 6.1.10 remain valid if we change in the definition of sets A and B "non-degenerated distribution" for "chi-squared distribution with $2(N-1)$ degrees of freedom".

In addition, we have the following corollary.

Corollary 6.1.11. *Under the assumptions of Theorem 6.1.9, $(P_\lambda \otimes P_{\mathbf{H}})[A] \in \{0,1\}$ and $\boldsymbol{X}$ is Gaussian if, and only if, $(P_\lambda \otimes P_{\mathbf{H}})[A] = 1$.*

Analogously, under the assumptions of Corollary 6.1.10, $P_\lambda(B) \in \{0,1\}$ and $\boldsymbol{X}$ is Gaussian if, and only if, $P_\lambda(B) = 1$.

6.1.6 Conditions to apply Lobato and Velasco test

In this subsection, we show that a slight modification of the statistic $\tilde{G}_Y$ satisfies Theorem 3.3.2 under different assumptions from the ones used in Lobato and Velasco [58].

The test statistic is

$$G_Y = \frac{n\hat{\mu}_3^2}{6|\hat{F}_3|} + \frac{n(\hat{\mu}_4 - 3\hat{\mu}_2^2)^2}{24|\hat{F}_4|}$$

with

$$\hat{F}_k = 2\sum_{t=1}^{\tau_n} \hat{\gamma}(t)(\hat{\gamma}(t) + \hat{\gamma}(\tau_n + 1 - t))^{k-1} + \hat{\gamma}^k,$$

where, according to Theorem 3.3.3, we take $\tau_n < cn^{\beta_0}$ for $\beta_0 = 1 - 2/\alpha$, $c > 0$ and $2 < \alpha < 4$. Thus, the differences between G_Y and $\tilde{G}_Y$ are the absolute values in the denominator and the number of terms involved in the estimator of F_k.

Theorem 6.1.12. *Let $(X_t)_{t\in\mathbb{Z}}$ be an ergodic and stationary process such that $\sum_{t=0}^{\infty} |\gamma_X(t)| < \infty$. We have that*

1. *If $(X_t)_{t\in\mathbb{Z}}$ is a Gaussian process, then $G_Y \longrightarrow_d \chi_2^2$.*
2. *If $(X_t - \mu_X)_{t\in\mathbb{Z}}$ can be written as (3.8) and $\mathbb{E}[X_0^4] < \infty$, then, conditionally on $\mathbf{h}$, G_Y diverges almost surely to infinity whenever $\mu_3 \neq 0$ or $\mu_4 \neq 3\mu_2^2$.*

Proof.
Using Proposition 6.1.3 for $\zeta = 0$ we get that $(Y_t)_{t\in\mathbb{Z}}$ is an ergodic and stationary process with $\sum_{t=0}^{\infty} |\gamma_Y(t)| < \infty$.

If $(X_t)_{t\in\mathbb{Z}}$ is Gaussian, the process $(Y_t)_{t\in\mathbb{Z}}$ is also Gaussian. Thus, the assumptions of the first part of Theorem 3.3.2 hold for the process $(Y_t)_{t\in\mathbb{Z}}$ and so $\tilde{G}_Y \longrightarrow_d \chi_2^2$. Now, as $\mathbf{Y}$ is Gaussian, by Gasser [34, p. 568], we have that $F_k > 0$ for $k = 3, 4$. Repeating the proof of Lemma 1 in Lobato and Velasco [58], we have that $\lim_{n\to\infty} \hat{F}_k = F_k$. Therefore, we may conclude that $\lim_{n\to\infty} G_Y = \lim_{n\to\infty} \tilde{G}_Y$, which shows *1.*

Let us now prove statement *2.* First, let us show that $\mathbb{E}[|Y|^k|\mathbf{h}] < \infty$, almost surely, for $k = 1, ..., 4$. By Hölder inequality, we have that $|Y_0| \leq (\sum_{i=0}^{\infty} a_i)^{1/2}(\sum_{i=0}^{\infty} h_i^2 a_i X_{-i}^2)^{1/2}$ and, as by Proposition 6.1.1 $\sum_{i=0}^{\infty} h_i^2 a_i = 1$, almost surely, we can apply Jensen inequality. We obtain that

$$Y_0^4 \leq \left(\sum_{i=0}^{\infty} a_i\right)^2 \left(\sum_{i=0}^{\infty} h_i^2 a_i X_{-i}^4\right), \text{ almost surely.}$$

Thus, $\mathbb{E}[|Y_0|^k|\mathbf{h}] < \infty$, almost surely, for $k = 1, ..., 4$. By Doob [25, p. 458], we have that $\left(Y_t^k\right)_{t\in\mathbb{Z}}$ is stationary and ergodic, for all $k = 1, ..., 4$. Therefore, Theorem 2 in Hannan [40, Chap. IV] implies that

$$\lim_{n\to\infty} \hat{\mu}_k = \mu_k, \text{ for almost every } \mathbf{h} \text{ and } k = 2, 3, 4. \tag{6.10}$$

Further, let us prove that $\lim_{n\to\infty} |\hat{F}_k| < \infty$ for almost every $\mathbf{h}$ and $k = 3, 4$. We have

$$\hat{F}_k = \hat{\gamma}_Y^k + 2\sum_{t=1}^{\tau_n}\sum_{j=0}^{k-1} \begin{pmatrix} k-1 \\ j \end{pmatrix} \hat{\gamma}_Y(t)^{k-j}\hat{\gamma}_Y(\tau_n + 1 - t)^j.$$

Taking into account that $|a^{k-j}b^j| \leq |a|^k + |b|^k$, with $k, j \in \mathbb{N}$ such that $j < k$, we have

$$|\hat{F}_k| \leq |\hat{\gamma}_Y|^k + 2^k \sum_{t=1}^{\tau_n} (|\hat{\gamma}_Y(t)|^k + |\hat{\gamma}_Y(\tau_n + 1 - t)|^k),$$

and then we obtain $|\hat{F}_k| \leq 2^{k+1}(\sum_{t=0}^{\tau_n} |\hat{\gamma}_Y(t)|)^k$. Let us prove now that

$$\lim_{n\to\infty} \sum_{t=0}^{\tau_n} |\hat{\gamma}_Y(t)| < \infty.$$

To prove this, we must start by proving that $\lim_{n\to\infty} \sum_{t=0}^{\tau_n} |\hat{\gamma}_X(t)| < \infty$. Note that as $\mathbb{E}[X_0^4] < \infty$, using (3.8) we also have

$$\begin{aligned} \infty > \mathbb{E}[(X_0 - \mu_X)^4] &= \sum_{j_1,...,j_4=1}^{\infty} \prod_{r=1}^{4} k(j_r) E\left[\prod_{r=1}^{4} \epsilon_{n-j_r}\right] \\ &= \mathbb{E}[\epsilon_1^4] \sum_{j=1}^{\infty} k(j)^4 + \mathbb{E}[\epsilon_1^2]^2 \sum_{i,j=1, i\neq j}^{\infty} k(i)^2 k(j)^2, \end{aligned}$$

because (ϵ_n) are i.i.d.r.vs. with $\mathbb{E}[\epsilon_1] = 0$. This implies $\mathbb{E}[\epsilon_1^4] < \infty$. Further, using Theorem 3.3.3 we obtain that

$$\left|\sum_{t=0}^{\tau_n} (|\hat{\gamma}_X(t)| - |\gamma_X(t)|)\right| \leq (\tau_n + 1) o\left(n^{2/\alpha - 1}\right) = o(1).$$

Thus, $\lim_{n\to\infty} \sum_{t=0}^{\tau_n} |\hat{\gamma}_X(t)| < \infty$. Then, by proceeding similarly as in the proof of Proposition 6.1.3, we get $\lim_{n\to\infty} \sum_{t=0}^{\tau_n} |\hat{\gamma}_Y(t)| < \infty$ and so, $\lim_{n\to\infty} |\hat{F}_k| < \infty$ for $k = 3, 4$. Using (6.10), it can be concluded that *2.* holds. □

Finally, applying Theorem 6.1.12 directly to the process $\mathbf{X}$, we obtain the following corollary.

Corollary 6.1.13. *Let $(X_t)_{t\in\mathbb{Z}}$ be an ergodic and stationary process such that $\sum_{t=0}^{\infty}|\gamma_X(t)|<\infty$. We have that*

1. *If $(X_t)_{t\in\mathbb{Z}}$ is a Gaussian process, then $G_X \longrightarrow_d \chi^2_2$.*
2. *If $(X_t-\mu_X)_{t\in\mathbb{Z}}$ can be written as (3.8) and $\mathbb{E}[X_0^4]<\infty$, then G_X diverges almost surely to infinity whenever $\mu_{X,3}\neq 0$ or $\mu_{X,4}\neq 3\mu_{X,2}^2$.*

6.2 The tests in practice

In this section, we discuss the practical implementation of our procedure. We start by making some remarks on the Epps test.

6.2.1 Remark on the Epps test

Although Theorem 3.3.1 works for any $\lambda\in\Lambda_N$, with $N>1$, that satisfies **H1** and **H2**, in Epps [26] it is stated that:

- *When either N is large or the spacing between the λ_j is small, relative to the scale of the data, the matrix $2\pi\hat{f}(0,\lambda)$ often appears computationally singular.*
- *Also, values of λ_j which are large, relative to the scale of the data, makes difficult to find a minimum of $Q_n(\cdot,\cdot,\lambda)$ with much precision.*

Epps suggests taking

$$\lambda_j=\xi_j/\sqrt{\hat{\gamma}},\ \text{with}\ \xi_j>0,\ j=1,...,N. \tag{6.11}$$

Recall that $\hat{\gamma}$ denotes the sample variance of the process. He proved that Theorem 3.3.1 works taking such λ. In the simulations of Epps, and also in those of Lobato and Velasco [58], $N=2$ and $(\xi_1,\xi_2)=(1,2)$.

However, we need to draw λ randomly in order to have a consistent test (Theorem 6.1.9). Thus, we take $N = 2$, ξ_1 distributed as the absolute value of a standard normal distribution and ξ_2 distributed as the absolute value of a normal distribution with mean zero and variance 4. With this selection, although seldom, we have found that $\hat{f}(0, \lambda)$ might be singular. This is the main reason for choosing $G_n^+(\lambda)$ as the generalized inverse of $2\pi\hat{f}(0, \lambda)$.

Another important practical issue is the procedure used to find the minimizer nearest to $(\hat{\mu}, \hat{\gamma})$ of the map $(\nu, \rho) \to Q_n(\nu, \rho, \lambda)$. In the simulations of Epps [26] and Lobato and Velasco [58], they use the simplex method developed in Nelder and Mead [67]. We did the same. The code of this method can be found in Press et al. [71] under the name `amoeba`.

6.2.2 The random projection procedure to test Gaussianity

The theoretical development of Section 6.1 was carried out assuming that the observed sample is infinite. However, in practice, only a finite number of measurements $X_0, \ldots, X_n$ are available. Thus, only a finite number of components of $\mathbf{h}$ are computed. This last difficulty is handled by fixing a small $\delta > 0$ (equal to 10^{-15} in the simulations that we present in Section 6.3) and by taking $\mathbf{h} = (h_0, \ldots, h_m)^T$ with

$$m - 1 = \min\{\min\left\{t : \|(h_0, \ldots, h_t)^T\| \geq 1 - \delta\right\}, n - 1\},$$

where $h_0, \ldots, h_{m-1}$ are drawn by the stick-breaking procedure described in Section 6.1. Further, h_m is fixed such that $\|\mathbf{h}\| = 1$. Concerning the projected process, several possibilities are available, but here we use

$$Y_t = \sum_{i=0}^{\min(m,t)} h_i X_{t-i} a_i, \ t = 0, \ldots, n.$$

Let us now make a short comment on the choice of the parameters $\alpha_1, \ \alpha_2 > 0$ of the beta distribution used to generate $\mathbf{h}$. Here, we have to deal with the following situation: If m is large, then the random variables Y_t are linear combinations of many random variables

from the first sample and then, because of the Central Limit Theorem, the distribution of the random variables Y_t will become close to a normal law. This will cause some loss of power when the marginal of $\mathbf{X}$ is not Gaussian. Thus, in order to detect a non-Gaussian marginal, it is wise to select α_1 and α_2 in such a way that m is small or even 0 or 1. This goal is achieved if we take $\alpha_2 = 1$ and $\alpha_1 \gg 1$. Our selection in Section 6.3 is $\alpha_1 = 100$. However, in this case the samples $Y_0, \ldots, Y_n$ and $X_0, \ldots, X_n$ are quite similar. Thus, the test will not be good at detecting non-Gaussian alternatives with Gaussian marginal. In order to overcome this problem, we should take $\mathbf{h}$ in such a way that the projections mix several variables from the initial sample. To achieve this goal we need to take $\alpha_2 > \alpha_1$ but with α_2 being not too big to avoid the effect of the Central Limit Theorem. In this case, a selection like $\alpha_1 = 2$ and $\alpha_2 = 7$ seems appropriate. Therefore, it seems that in a practical situation, we should decide which alternative is more plausible and then, select the appropriate parameters. However, there is another possibility: select two projections (one with each pair of parameters) and apply Theorem 3.4.1 to mix the p-values. This is our proposal.

Finally, we need a Gaussianity test for the one dimensional marginal of $(Y_0, \ldots, Y_n)$. We have seen two such tests (which have some advantages and disadvantages discussed in Section 6.3) and we can also mix them. Bearing all these requirements in mind, we propose the following procedure:

1. Draw $\mathbf{h}^{(1)}$ with the $\beta(100, 1)$ distribution and apply the Epps test to the projections to obtain the p-value $p^{(1)}$.

2. Draw $\mathbf{h}^{(2)}$ (independently of $\mathbf{h}^{(1)}$) with the $\beta(100, 1)$ distribution and apply the Lobato and Velasco test to the projections to obtain the p-value $p^{(2)}$.

3. Draw $\mathbf{h}^{(3)}$ (independently of $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$) with the $\beta(2, 7)$ distribution and apply the Epps test to the projections to obtain the p-value $p^{(3)}$.

4. Draw $\mathbf{h}^{(4)}$ (independently of $\mathbf{h}^{(1)}$, $\mathbf{h}^{(2)}$ and $\mathbf{h}^{(3)}$) with the $\beta(2, 7)$ distribution and

apply the Lobato and Velasco test to the projections to obtain the p-value $p^{(4)}$.

5. Combine the p-values $p^{(1)}, \ldots, p^{(4)}$ using the procedure described in Section 5.4 to decide the Gaussianity hypothesis at the level α. Thus, ordering these four p-values such that $p_{(1)} \leq ... \leq p_{(4)}$ we obtain that the p-value of the random projection test is equal to $(25/3) \cdot \min_{i=1,\ldots,4} p_{(i)}/i$.

6.3 Simulations

In this section, we study the behavior of the proposed procedure in different situations. We have used the same distributions as in Lobato and Velasco [58], in order to perform comparisons. Further, we will study a situation where the process has Gaussian marginal but is not Gaussian (see Section 6.3.1). In addition, in Subsection 6.3.3 we apply the random projection test to real data.

The authors of Lobato and Velasco [58] study the case of an AR(1) process depending on a parameter q defined by

$$X_t = qX_{t-1} + \varepsilon_t, \tag{6.12}$$

where $q \in \{-.9, -.5, 0, .5, .6, .7, .8, .9\}$, $t \in \mathbb{Z}$ and ε_t are i.i.d. random variables with distribution D_ε which may be any of the following ones:

- standard normal (N(0,1)),
- standard log-normal (log N),
- Student t with 10 degrees of freedom, (t_{10}),
- chi-squared with 1 (χ^2_1) and 10 degrees of freedom (χ^2_{10}),
- uniform on $[0, 1]$ ($U(0, 1)$),
- beta with parameters $(2, 1)$ ($\beta(2, 1)$).

To simulate the process, we generate a large number of independent realizations $\varepsilon_t,\ t = 1, \ldots, M$ with distribution D_ε and we take

- $X_1 = \varepsilon_1$
- $X_t = qX_{t-1} + \varepsilon_t,\ t = 2, ..., M.$

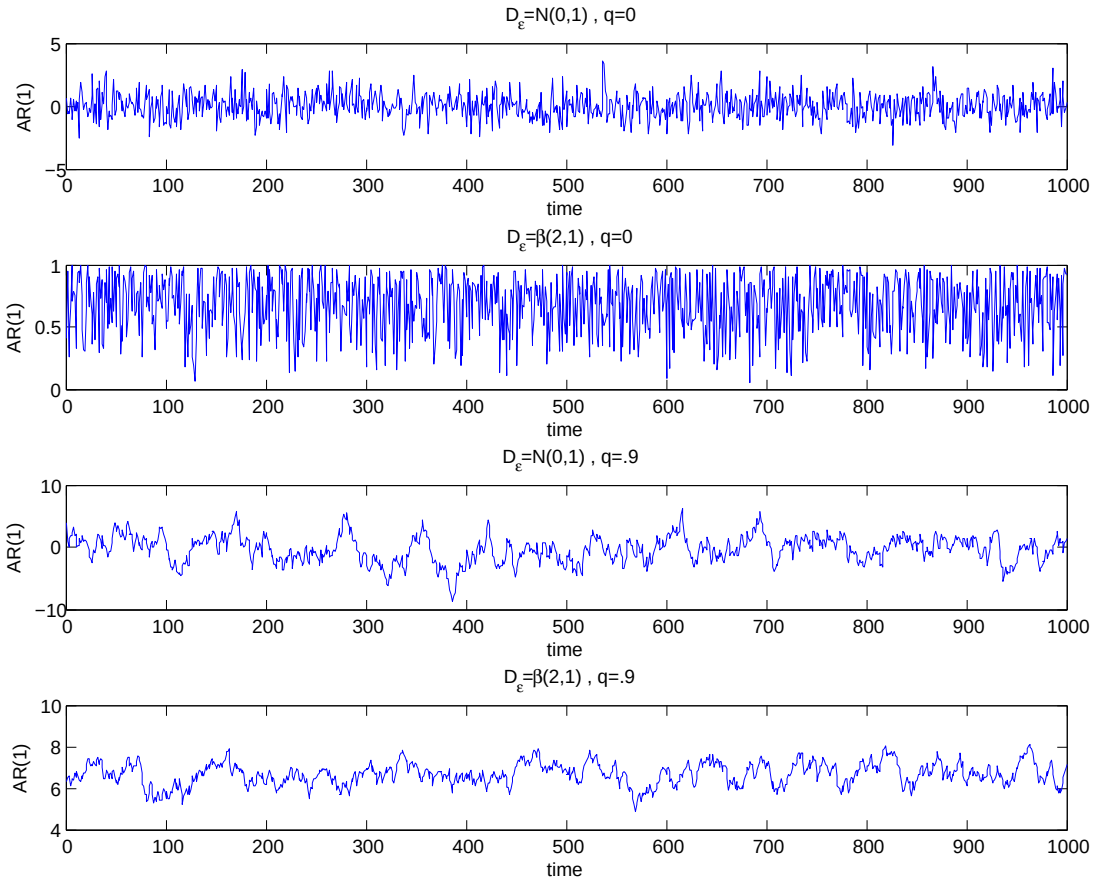

Figure 6.1: From top to bottom. AR(1) processes with $D_\varepsilon = N(0,1)$ and $q = 0$, $D_\varepsilon = \beta(2,1)$ and $q = 0$, $D_\varepsilon = N(0,1)$ and $q = .9$, and $D_\varepsilon = \beta(2,1)$ and $q = .9$,.

It is obvious that if $q \neq 0$, this process is not stationary. For instance, $\mathrm{Var}[X_t] = \mathrm{Var}[\varepsilon_1](1-q^{2t})/(1-q^2)$ which is not constant and, obviously, the differences increase with $|q|$. In order to alleviate this problem, we discarded a certain number, *past*, of observations. We have taken *past* to be equal to 1000 and $n = M - past$ equal to $100, 500, 1000,$

which are the sample sizes handled in Lobato and Velasco [58]. For some examples of these processes, see Figure 6.1.

We have performed 5000 simulations in each situation. In every run, we have computed the p-values using the asymptotic distributions. This may have caused the rejection rates under the null hypothesis to move somewhat far from the nominal level (mostly for the lowest sample size $n = 100$) and to decrease under some alternatives with the sample size (mostly for high values of $|q|$).

There are some differences between our rates and those published in Lobato and Velasco [58]. We think that these could be due to the fact that the *past* taken in Lobato and Velasco [58] is not large enough. For example, in the case $n = 100$, $q = .7$ and D_ε being $\beta(2, 1)$ we obtain a rejection rate of .2214 when using the Epps test while in Lobato and Velasco [58] they obtain one of .080, which is noticeably worse. As explained before, our simulations were made with *past*= 1000, but from Table 6.1 we see that .080 is a reasonable rejection rate for *past* = 0 and that the rejection rates increase with *past*, approaching the value we have obtained.

past	0	1	2	10
rejections	.0750	.1378	.1998	.2210

Table 6.1: Rejection rates along 5,000 simulations for different *past*, with the Epps test, $n = 100$, D_ε a $\beta(2, 1)$ and $q = .7$.

We have observed the same problem with the Lobato and Velasco test, except that with this other test, our rejection rates are lower than those reported in Lobato and Velasco [58]. We think that those differences are also due to the same problem.

Furthermore, another difference to highlight between what we do here and Lobato and

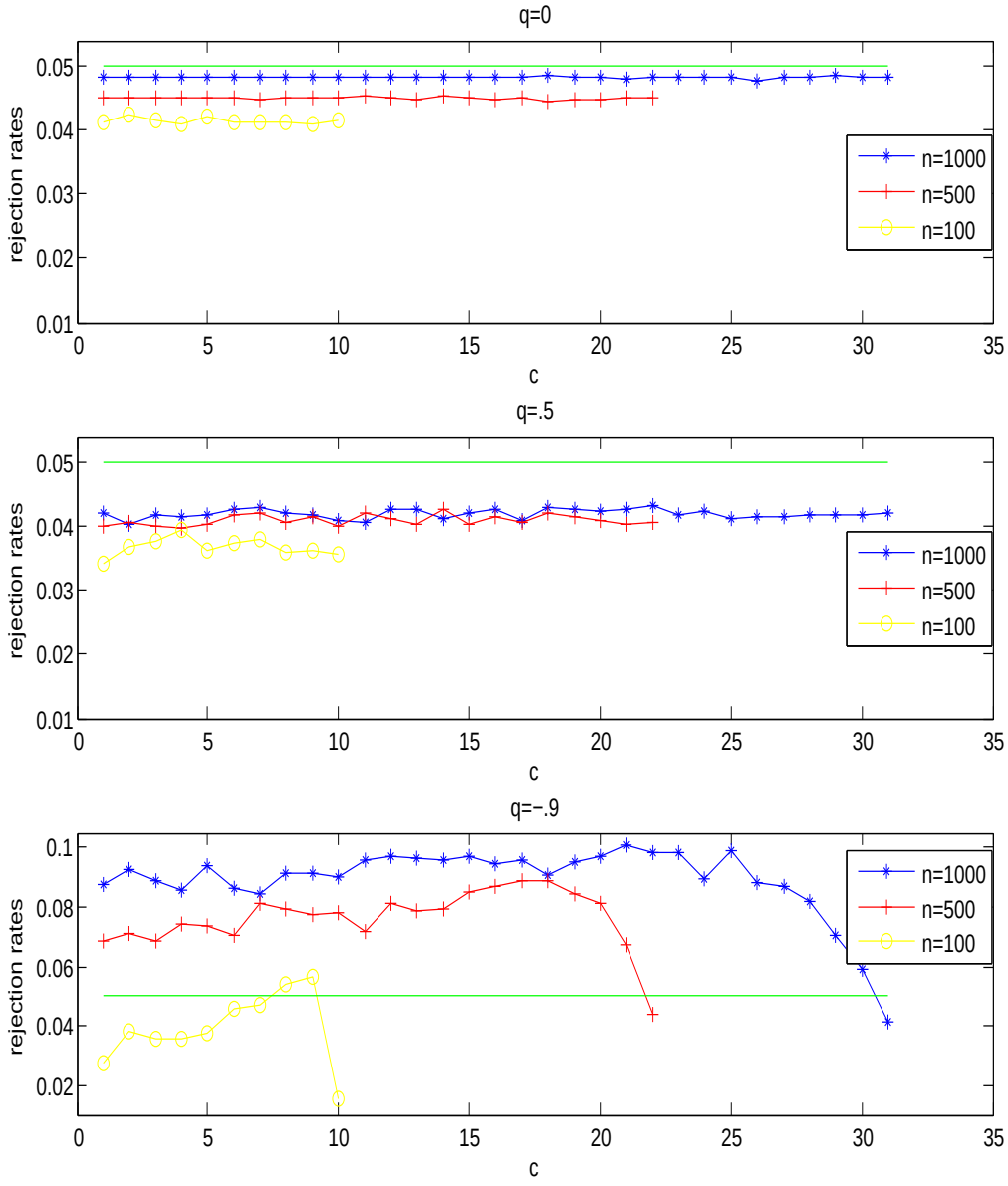

Figure 6.2: Rejection rates under the null hypothesis for an AR(1) process with $q = 0$ (upside graph), $q = .5$ (middle graph) and $q = -.9$ (downside graph), using the Lobato and Velasco test for different values of c and sample sizes.

Velasco [58] is that in Subsection 6.1.6, a sum until τ_n is involved in the estimation of F_k while in Lobato and Velasco [58] the sum goes until $n-1$, where n is the sample size. Here, we have to take $\tau_n < cn^{\beta_0}$, where $\beta_0 = 1 - 2/\alpha$ with $2 < \alpha < 4$ and $c > 0$. Thus, β_0 may be as close as desired to .5 and so, we have decided to fix its value at $\beta_0 = .5$ for the simulations. In order to select the right value of c, we have made a small analysis to see how sensitive the method is to this parameter. We run the Lobato and Velasco test under the null hypothesis for all values of q and $c = 1, 2, \ldots, c_n$, where $c_n = \lfloor \sqrt{n} \rfloor$ and $n = 100, 500, 1000$. Therefore, $c_{100} = 10$, $c_{500} = 22$ and $c_{1000} = 31$. The results suggest that the value of c has little influence on the rejection rates and so, we choose $c = 1$. The results for the cases $q = 0$, $q = .5$ and $q = -.9$ appear in Figure 6.2. It is worth pointing out that the situation for $q = -.9$ is slightly different than for all the other values of q, as with $q = -.9$ the rejection rates look more or less constant up to a point in which those rates strongly decrease.

Tables 6.2, 6.3 and 6.4 contain the rejection rates for several procedures when applied at the level .05. Next, we mention the procedures we have selected and make some comments on the results of our simulations.

1. **Epps test, E-test**. We take $(\xi_1, \xi_2) = (1, 2)$ in (6.11).

   It seems that this test behaves poorly when D_ε is t_{10}. Moreover, broadly speaking, its power decreases for the considered alternative distributions when $|q|$ increases, having low powers when $|q| = .9$. Note also that under the null hypothesis (except the case $q = 0$ with $n = 1000$), the rejection rates are above the level of the test and that they increase with $|q|$.

   The power decreases when the sample size increases in the cases in which $|q| = .9$ and the alternative is t_{10}, χ^2_{10}, $U(0, 1)$ or $\beta(2, 1)$ (and even with $q = .8$ when $D_\varepsilon = t_{10}$).

2. **Lobato and Velasco test, G-test.** The rejection rates displayed have been simulated using the statistic G_X. However, they are similar to those obtained using

$\tilde{G}_X$.

The **G**-test has very low powers when $|q|$ is large, sometimes even lower than those of the **E**-test. In addition it suffers from a lack of power when D_ϵ is $U(0,1)$ or $\beta(2,1)$. The rejections under the null hypothesis are above the level of the test only in 4 cases out of 24. In contrast with the **E**-test, here the rejection rates under the null hypothesis decrease when q increases.

3. **Combined Epps and Lobato and Velasco test, GE-test.** In previous paragraphs we have commented some problems of the **E** and **G** tests which go, let us say, in opposite directions. In order to solve these problems we combine both tests, using the multiple testing procedure presented in Section 5.4.

   As stated in Subsection 6.2.1, the **GE**-test has been obtained by drawing independently ξ_1 with the absolute value of a standard normal distribution and ξ_2 with the absolute value of a normal distribution with mean zero and standard deviation 2. It is worth noting that the rejection rates we have obtained have been slightly larger than in the case we take $(\xi_1, \xi_2) = (1, 2)$.

   We can observe from Tables 6.2, 6.3 and 6.4 that this combination gives rejection rates between those of the **E** and **G**-tests although closest to the highest one, and, sometimes, even above. This is because, as previously stated, the rejection rates of **E** are slightly larger here than when $(\xi_1, \xi_2) = (1, 2)$.

4. **Random projection test, RP-test.** We apply this test following the guidelines provided in Subsection 6.2.2.

   When q is negative and we are under the alternative, we always get the highest rejection rates with the **RP**-test. The most striking behavior of this test occurs for $q = .9$ and $D_\varepsilon = \chi^2_{10}$ and $\beta(2,1)$, where the **RP**-test obtains rejection rates larger than 0.8 while the second more successful test remains below 0.25. For the remaining values, the rejection rates using the **RP**-test are between the rates obtained with the **E, G** and **GE** tests but closer to the highest than to the lowest.

q	Test	N(0,1)	log N	t_{10}	χ^2_1	χ^2_{10}	$U(0,1)$	$\beta(2,1)$
	E	.1264	.0508	.1104	.0656	.1124	.1390	.1354
-.9	G	.0292	.1414	.0310	.0840	.0332	.0290	.0266
	GE	.0942	.1422	.0908	.1072	.0920	.1020	.1010
	RP	.1380	.8070	.1742	.7576	.3076	.2620	.3902
	E	.0724	.6780	.0556	.8514	.2058	.5408	.4914
-.5	G	.0504	.9986	.1692	.9986	.4602	.0102	.1696
	GE	.0774	.9976	.1582	.9972	.4552	.4454	.4154
	RP	.0752	.9998	.1980	1	.5824	.6404	.7460
	E	.0632	.9616	.0830	.9964	.5372	.9918	.9704
0	G	.0458	1	.2820	1	.7898	.5404	.7520
	GE	.0732	1	.2402	1	.8074	.8596	.8706
	RP	.0772	1	.2288	1	.7640	.8496	.9054
	E	.0682	.8594	.0608	.9582	.2610	.5618	.5562
.5	G	.0384	.9990	.1696	.9982	.4118	.0010	.1102
	GE	.0642	.9990	.1444	.9988	.4700	.4680	.4882
	RP	.0750	.9908	.1132	.9880	.5226	.3256	.7500
	E	.0710	.6118	.0582	.8106	.2006	.3462	.3650
.6	G	.0358	.9884	.1162	.9772	.2858	.0012	.0592
	GE	.0640	.9882	.1144	.9832	.3218	.2800	.3086
	RP	.0802	.9536	.1030	.9262	.5164	.2580	.7744
	E	.0838	.3250	.0626	.4640	.1492	.2032	.2214
.7	G	.0260	.9076	.0814	.8196	.1610	.0036	.0334
	GE	.0714	.9042	.0866	.8448	.1998	.1634	.1802
	RP	.0784	.8022	.0926	.7010	.5754	.2902	.8060
	E	.1034	.1552	.0810	.2004	.1324	.1620	.1596
.8	G	.0206	.6146	.0466	.4406	.0708	.0046	.0166
	GE	.0726	.6118	.0796	.4488	.1122	.1154	.1136
	RP	.0896	.4928	.0932	.3264	.6766	.3950	.8782
	E	.1752	.1264	.1618	.1368	.1612	.1870	.1680
.9	G	.0106	.1558	.0094	.0714	.0150	.0054	.0086
	GE	.1074	.1844	.0968	.1190	.0980	.1182	.1072
	RP	.1168	.1982	.1174	.1338	.8702	.6788	.9662

Table 6.2: Rejection rates at level .05 of a process defined by (6.12). Sample size $n = 100$.

q	Test	N(0,1)	log N	t_{10}	χ^2_1	χ^2_{10}	$U(0,1)$	$\beta(2,1)$
	E	.0744	.3720	.0584	.2162	.0712	.0918	.0850
-.9	G	.0708	.8838	.0840	.6202	.1142	.0462	.0754
	GE	.0780	.8604	.0924	.5400	.1116	.0866	.0952
	RP	.0810	.9990	.2260	.9928	.6924	.4630	.6918
	E	.0594	1	.1334	1	.7730	.9924	.9922
-.5	G	.0472	1	.4580	1	.9960	.9656	.9976
	GE	.0476	1	.3784	1	.9912	.9514	.9914
	RP	.0490	1	.5090	1	.9998	.9946	1
	E	.0566	1	.3292	1	.9982	1	1
0	G	.0480	1	.7428	1	1	1	1
	GE	.0510	1	.6756	1	1	1	1
	RP	.0554	1	.6188	1	1	1	1
	E	.0654	1	.1476	1	.8808	.9918	.9960
.5	G	.0454	1	.4340	1	.9972	.9704	.9988
	GE	.0516	1	.3816	1	.9924	.9504	.9962
	RP	.0618	1	.2656	1	.9610	.7440	.9634
	E	.0566	.9998	.1026	1	.7084	.8286	.9090
.6	G	.0470	1	.3336	1	.9582	.4678	.8858
	GE	.0570	1	.2692	1	.9388	.6944	.8870
	RP	.0610	1	.1794	1	.8604	.4730	.9006
	E	.0708	.9996	.0786	1	.4704	.4042	.5810
.7	G	.0474	1	.1970	1	.7592	.0644	.4040
	GE	.0598	1	.1670	1	.7332	.3640	.5768
	RP	.0702	1	.1282	1	.6986	.2616	.8786
	E	.0776	.9780	.0710	.9638	.2500	.1948	.2564
.8	G	.0744	.9998	.0976	.9980	.3908	.1524	.2628
	GE	.0702	.9998	.1102	.9978	.3972	.1848	.2960
	RP	.0710	.9986	.0910	.9908	.6834	.2484	.9208
	E	.1156	.5708	.0944	.4674	.1526	.1430	.1560
.9	G	.0232	.8356	.0370	.5404	.0764	.0138	.0336
	GE	.0802	.8708	.0838	.6378	.1490	.1092	.1390
	RP	.0860	.7996	.0770	.5510	.8430	.4818	.9772

Table 6.3: Rejection rates at level .05 of a process defined by (6.12). Sample size $n = 500$.

q	Test	N(0,1)	log N	t_{10}	χ^2_1	χ^2_{10}	$U(0,1)$	$\beta(2,1)$
	E	.0648	.7836	.0578	.4572	.0826	.0888	.0942
-.9	G	.0902	.9934	.1206	.8932	.2448	.0760	.1358
	GE	.0880	.9856	.1002	.8560	.2190	.1004	.1450
	RP	.0940	1	.3344	.9998	.8686	.5876	.8056
	E	.0530	1	.2574	1	.9764	1	1
-.5	G	.0436	1	.6778	1	1	1	1
	GE	.0450	1	.6040	1	1	1	1
	RP	.0378	1	.7498	1	1	1	1
	E	.0490	1	.5946	1	1	1	1
0	G	.0546	1	.9364	1	1	1	1
	GE	.0486	1	.9162	1	1	1	1
	RP	.0422	1	.8734	1	1	1	1
	E	.0550	1	.2534	1	.9966	1	1
.5	G	.0482	1	.6788	1	1	1	1
	GE	.0424	1	.6016	1	1	1	1
	RP	.0484	1	.4348	1	.9994	.9738	.9996
	E	.0566	1	.1718	1	.9580	.9800	.9974
.6	G	.0472	1	.5112	1	.9996	.9724	.9996
	GE	.0464	1	.4234	1	.9996	.9550	.9986
	RP	.0584	1	.2812	1	.9902	.7110	.9804
	E	.0594	1	.1162	1	.7720	.6338	.8632
.7	G	.0418	1	.3104	1	.9744	.3642	.8830
	GE	.0558	1	.2380	1	.9672	.5642	.8724
	RP	.0598	1	.1754	1	.8888	.3554	.9036
	E	.0690	.9998	.0720	1	.4342	.2288	.4108
.8	G	.0500	1	.1638	1	.6804	.0432	.3284
	GE	.0670	1	.1294	1	.6708	.2216	.4450
	RP	.0654	1	.0996	1	.7144	.1920	.9076
	E	.0902	.9152	.0880	.7690	.1836	.1170	.1686
.9	G	.0346	.9944	.0636	.9136	.1574	.0174	.0574
	GE	.0690	.9926	.0798	.9206	.2178	.1040	.1596
	RP	.0736	.9844	.0678	.8580	.8328	.3946	.9774

Table 6.4: Rejection rates at level .05 of a process defined by (6.12). Sample size $n = 1000$.

6.3.1 A stationary non-Gaussian process with Gaussian marginal

In this subsection, we discuss the behavior of the proposed procedure when used on a non-Gaussian process with Gaussian marginal. We have worked with the process introduced in Example 2.3 in Cuesta-Albertos and Matrán [22]. Its construction is explained here for the sake of completeness.

Let p be a prime number, and let Y_0, U and $\{Z_{m\cdot p}, m = 0, 1, \ldots\}$ be mutually independent random variables all uniformly distributed on $\{0, 1, \ldots, p-1\}$. Set

$$Z_{m\cdot p+k} \quad = \quad Z_{m\cdot p} \oplus (kY_0) \ , \ k = 0, \ldots, p-1, m = 0, 1, 2, \ldots$$

where $\oplus$ stands for sum modulus p. According to Cuesta-Albertos and Matrán [22] the sequence $W_n = Z_{n+U}$ is composed of pairwise independent random variables and it is stationary. Moreover, these random variables are not mutually independent because, by construction, for every $m \in \mathbb{N}$ we have that

$$Z_{m\cdot p} + Z_{m\cdot p+1} + \ldots + Z_{m\cdot p+p-1} = p(p-1)/2,$$

and so,

$$W_{mp-U} + W_{mp-U+1} + \ldots + W_{mp-U+p-1} = p(p-1)/2. \tag{6.13}$$

Therefore, the knowledge of the random variables $W_{n-U}, W_{n-U+1}, \ldots, W_{n-U+p-2}$ completely determines the value of $W_{n-U+p-1}$.

Now, given $k \in \{0, \ldots, p-1\}$, let q_k be the quantile of order k/p of the standard Gaussian distribution. For every $n \in \mathbb{N}$, let us define the random variable W_n^* conditionally to W_n as follows: If $W_n = k$, then draw the value of W_n^* with a standard Gaussian distribution conditioned to be in the interval (q_k, q_{k+1}), and independent of all the other random variables.

Since W_n is uniformly distributed on $\{0, 1, \ldots, p-1\}$, we obviously have that W_n^* is a standard Gaussian r.v.. Moreover, the sequence (W_n^*) inherits the remaining properties

of (W_n): this is a strictly stationary sequence of pairwise independent Gaussian random variables. However, if $n > p-1$ and we are aware of the values $W^*_{n-U}, \ldots, W^*_{n-U+p-2}$, we can recover the values $W_{n-U}, \ldots, W_{n-U+p-2}$ and, because of (6.13), we may deduce the value of $W_{n-U+p-1}$. With this information, we know to which interval $W^*_{n-U+p-1}$ belongs. Therefore, the random variables $(W^*_n)_n$ are not mutually independent and so, the process is not Gaussian. For examples of such processes, see Figure 6.3.

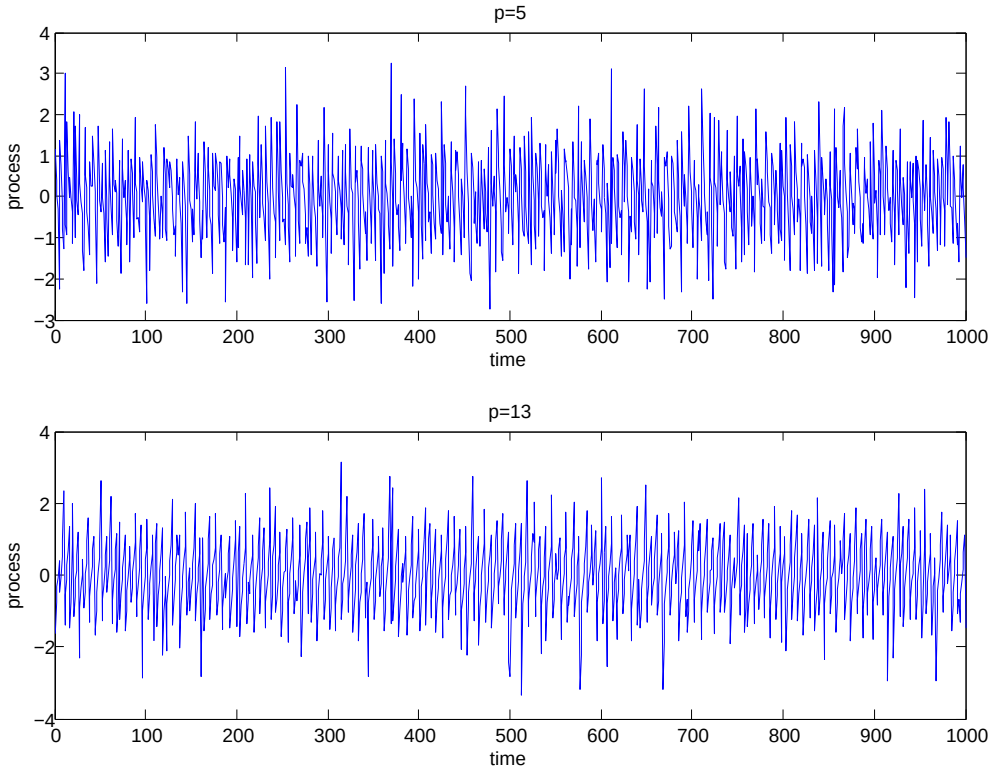

Figure 6.3: $\mathbf{W}^*$ process for $p = 5$ (upside graph) and $p = 13$ (downside graph).

We have simulated the previous process $5,000$ times for different values of p and sample sizes $n = 100, 500, 1000$. Then, we have applied the **RP** test at the level $\alpha = .05$. The rejection rates appear in Table 6.5.

	$p=2$	$p=3$	$p=5$	$p=7$	$p=11$	$p=13$	$p=17$
$n=100$	.1448	.1268	.1676	.1516	.1602	.1380	.1146
$n=500$	.3698	.3654	.4938	.5154	.5822	.5590	.5588
$n=1000$	.6382	.6386	.6814	.7250	.7802	.7608	.7700

Table 6.5: Rejection rates for different sample sizes applying the **RP** test to the $\mathbf{W}^*$ process at the level $\alpha = .05$.

For comparison, we show in Table 6.6 the rates of rejection when using the **E**, **G** and **GE** tests in the case $p = 5$. Since these tests check for the non-Gaussianity of the marginal, the rejection rates are not too high. However, it is worth paying some attention to the rejection rates in this table. To begin with, they are below the intended level (except GE with $n = 100$), but, more surprisingly, they show some decrease when the sample size increases. We think that this is due to the fact that these tests see the process $\mathbb{W}^*$ as *more Gaussian than a Gaussian process*. The reason is that when we generate observations of a Gaussian process, *approximately* a proportion of $1/p$ observations are in the interval (q_k, q_{k+1}), with $k \in \{0, \ldots, p-1\}$. However, the process $\mathbb{W}^*$ generates *exactly* a proportion of $1/p$ observations in each interval (q_k, q_{k+1}). Thus, it has a "more Gaussian" behavior than expected. Consequently, the rejection rates are lower than .05 and this fact becomes more apparent when n increases.

	$n=100$	$n=500$	$n=1000$
E	.0338	.0266	.0186
G	.0372	.0336	.0326
GE	.0520	.0336	.0206

Table 6.6: Rejection rates using the E, G and GE tests of the $\mathbb{W}^*$ process with $p = 5$, at the level $\alpha = .05$.

6.3.2 Increasing the number of projections

Although the rejection rates shown in Table 6.5 are above the nominal level, they are not so high, especially when the sample size is 100. A simple way to improve these rates is to increase the number of random projections using the correction described in Section 5.4.

From Table 6.7 it can be seen how an increase in the number of random projections employed noticeably improves the rates. In this table, half of the projections are taken using the $\beta(100, 1)$ distribution and the other half with the $\beta(2, 7)$ and in each case we compute half of the p-values with the **E** test and the other half with the **G** test. It is important to note that as in Section 6.3.1 we have simulated 5,000 times the process described there for samples sizes $n = 100, 500, 1000$ and apply the **RP**-test at level .05.

	$k = 2$	$k = 3$	$k = 5$	$k = 8$
$n = 100$	.1676	.1906	.2288	.2674
$n = 500$	.4938	.5772	.6988	.8064
$n = 1000$	.6814	.7688	.8498	.8628

Table 6.7: Rejection rates for different sample sizes applying the **RP** test with 2^k projections to the $\mathbb{W}^*$ process with $p = 5$.

6.3.3 Real data

Canadian lynx and Wolfer sunspot data

In this subsection, we work with the well-known Canadian lynx and Wolfer sunspot data in order to illustrate the behavior of the random projection test. These data are displayed in Figure 6.4.

The Canadian lynx data consists in the annual record of the number of lynxes trapped

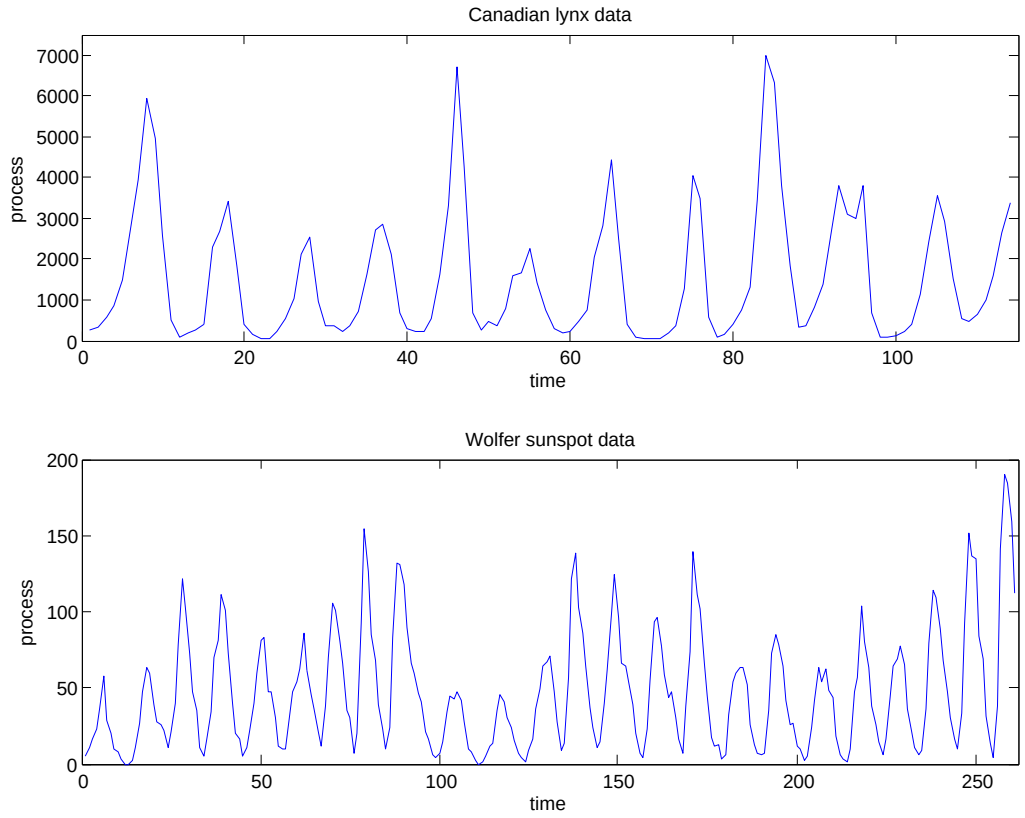

Figure 6.4: Canadian lynx (upside graph) and Wolfer sunspot data (downside graph).

in the Mackenzie River district of the North-West Canada for the period from 1821 to 1934 while the Wolfer sunspot data consists in the annual record of the sunspot activity in the period from 1700 to 1960. These data were used in Epps [26] and previously in Subba and Gabr [78], obtaining in both cases that the processes are not Gaussian.

We perform the random projection procedure to the lynx and sunspot data following the indications in Subsection 6.2.2. The obtained p-values are displayed in Table 6.8 together with those obtained in Epps [26] and in Subba and Gabr [78].

In these examples, we obtain p-values having approximatively the same magnitudes

	RP	Epps	S.R. & G
lynx	1.029×10^{-4}	1.402×10^{-5}	1.084×10^{-4}
sunspot	1.314×10^{-6}	7.356×10^{-6}	2.818×10^{-4}

Table 6.8: P-values using the **RP**-test and the tests proposed in Epps [26] and in Subba and Gabr [78] for the lynx and sunspot data.

as those of Epps [26] and Subba and Gabr [78].

Sea wave data

In this section, we analyze the Gaussianity of the heights of the sea level, which vary continuously with the waves. Until recently, the statistical procedures used to work with sea waves assumed the hypothesis of Gaussianity. It is known that when the sea is not calm, this hypothesis is not verified. However, this is not yet proved for a calm sea. It is believed by the experts that although the one-dimensional marginal of a calm sea is normal, this does not hold for higher order marginals, which makes the data suitable for our method.

The data we use here were measured with a datawell directional buoy on the East Coast of the USA which is referenced as 15401 Block Island, RI. Its deployment latitude and longitude are respectively 40 58.150' N and 71 07.543' W and the water depth is 48.16 meters. The location of the buoy can be seen in Figure 6.5.

The data have been downloaded from http://cdip.ucsd.edu. This web-site gives coastal data from the first observations made in 1975 to values just five minutes old; these measurements are being recorded by sensors in the water at this very moment.

We will work with the data of the first of November 2009 from 10:00 to 16:00, which makes a total of 27,648 data. Our first aim was to obtain a segment that was stationary.

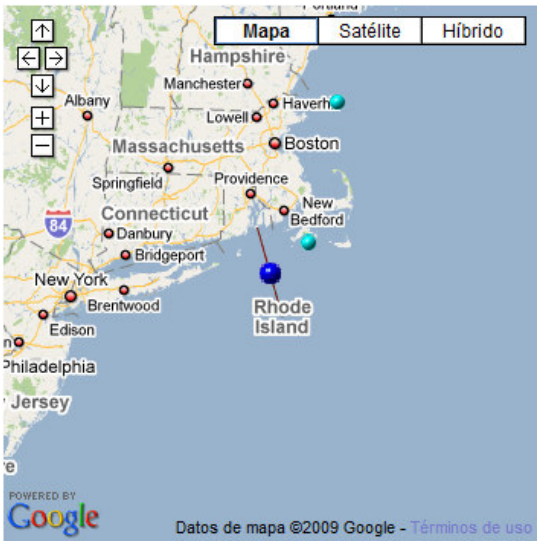

Figure 6.5: This picture represents in dark blue the buoy 15401 Block Island, RI. It has been taken from http://cdip.ucsd.edu.

For this we have used Soukissian's algorithm (see Soukissian and Samalekos [79]). For more information about segmentation algorithms for sea waves, see Hernández and Ortega [42]. Under the null hypothesis of Gaussianity, the first stationary segment we have obtained goes from 10:00 to 10:35. Therefore, among the 27,648 observations, we take the observations from 1 to 2,688. This segment can be visualized in Figure 6.6.

Secondly, we have applied the random projection procedure described in Subsection 6.2.2 and the **GE**-test, obtaining the p-values displayed in Table 6.9.

	RP	GE
p-value	.0890	.3825

Table 6.9: p-values using the **RP**-test and the **GE**-test for the sea wave data.

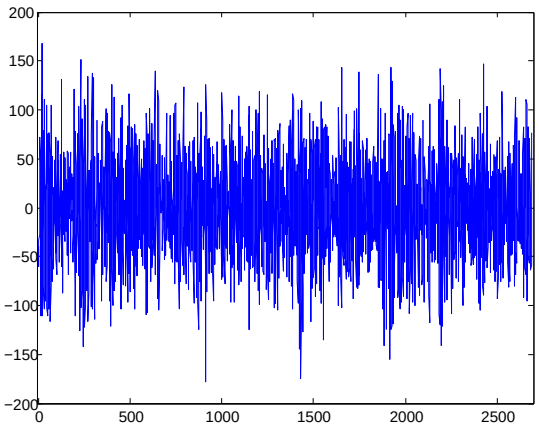

Figure 6.6: Height of the sea level measured the first of November of 2009 from 10:00 to 10:35, by buoy 15401 Block Island, RI.

It is worth noting that although with the p-values described in Table 6.9 neither of the two tests rejects the null hypothesis at level .05, the p-value obtained with the random projection procedure is much smaller.

Our work with this kind of data is in progress. However, the small example we have shown here leads us to consider that the random projection method should be useful for this aim. Finally, we would like to thank J.B. Hernández, J. León and J. Ortega for introducing us to the world of sea waves.

Chapter 7

Discussion

The random Tukey depth.

In this thesis, we introduce the random Tukey depth, which can be considered as a random approximation of the Tukey depth. The new depth is interesting because of the little effort required in its computation and because it can be extended to cover Hilbert valued data. This depth satisfies most of the properties of a depth according to the definition in Zuo and Serfling [83]. In addition, this depth can be consistently estimated from a random sample, both in the finite and the infinite dimensional settings. Moreover, it characterizes discrete distributions.

The attraction of the random Tukey depth lies in the fact that by taking only a few one-dimensional projections, it is possible to obtain results similar to those obtained with more involved depths. The number of required projections is surprisingly low indeed. Particularly, this is shown in the comparisons with the Tukey depth that we have carried out. Those studies do not show relevant differences between the results obtained with the Tukey depth and with the random Tukey depth. Thus, we conclude that, at least under the considered conditions, the random Tukey depth is an alternative which is worth considering because of its low computing time.

When this depth is applied to classification problems, the results are similar to those obtained with random forests but worse than those obtained with k-NN or kernel methods. However, the improvement which appears between the first and the second column in Table 5.9 makes us relatively optimistic about the results which could be obtained if an optimal procedure to select the distribution ν were applied.

Due to the generalization of the main results in Cuesta-Albertos et al. [15], which appears in Cuevas and Fraiman [24], it is possible to extend the results on the random Tukey depth to more general spaces.

Test of Gaussianity for stationary processes.

In Chapter 6, we have introduced the random projection test, **RP**-test, to check the Gaussianity of stationary processes. Given a sample, this test is based on a three-step procedure. First, a vector $\mathbf{h}$ must be drawn in a suitable Hilbert space which contains the process. Then, the sample is projected on the one-dimensional space spanned by $\mathbf{h}$. Finally, we take advantage of the fact that, with probability one, the initial process is Gaussian if the marginal of the projected one is Gaussian. Therefore, we only need to use a test to check the Gaussianity of the marginal of a stationary process. In the final step, we use a combination of the Epps and Lobato and Velasco tests.

The comparison of the **RP**-procedure with the Epps and Lobato and Velasco tests (as well as with the combination of these) in situations in which the marginal is not Gaussian is not bad, and there are cases in which the proposed test is clearly better. Moreover, the **RP** test is able to detect alternatives with Gaussian marginal, while the other tests are not designed to perform this task.

Appendix A

Computational codes

A.1 Preliminary results

A.1.1 Definition of data depth

The aim of the following program is to display the behavior of the data depths in dimension 2. It gives as output Figure 3.1 and Figure 4.1. Note that it calls `Depth.m`, which appears next in order to compute the Tukey and the random Tukey depth.

```
% This program generates a sample, computes the Mahalanobis, Tukey
% and two random Tukey depths of it and plots the results.
clear all
% Setting
p=2; nX=200; nV1=2; nV2=20;
X=randn(nX,p);  V1=rand(p,nV1);  V2=rand(p,nV2);
% Mahalanobis depth
S=inv(cov(X)); mX=mean(X); Mh=zeros(1,nX);
for  i=1:nX; xM=X(i,:)-mX; Mh(i)=inv(1+xM*S*xM'); end
% Tukey and random Tukey depths
IndiceTuc=0:pi/1000:pi; u=[cos(IndiceTuc) ; sin(IndiceTuc)];
```

```
T=Depth(X*u); RT1=Depth(X*V1); RT2=Depth(X*V2);
% Plotting
s=10; i=0:2*pi/s:2*pi; r=.1; xc=r*cos(i); yc=r*sin(i);
figure(1);
for j=1:nX; x=X(j,1)+xc; y=X(j,2)+yc; fill(x,y,RT1(j)); hold on; end
figure(2);
for j=1:nX; x=X(j,1)+xc; y=X(j,2)+yc; fill(x,y,RT2(j)); hold on; end
figure(3);
for j=1:nX; x=X(j,1)+xc; y=X(j,2)+yc; fill(x,y,Mh(j)); hold on; end
figure(4);
for j=1:nX; x=X(j,1)+xc; y=X(j,2)+yc; fill(x,y,T(j)); hold on; end
```

Depth.m

```
% Input data:  Prod = matrix containing the 1-dimensional projections
% of n points on k randomly chosen vectors.
% Output data: D = vector with the ranks associated to the depths of
% the n points.
function D=Depth(Prod)
[n k]=size(Prod);
% Vectors I1 and I2 are intended to compute the depths (ranks) in
% each projection. We give depth(rank) = 1 to the most outer point
%                          depth(rank) = 2 to the next one
%                          depth(rank) = integer part of n/2 to the
% deepest point
I1=1:n/2; I2=n:-1:(n/2+.1); PP=zeros(k,n); [SDat Indice]=sort(Prod);
for i=1:k; PP(i,Indice(I1,i))=I1; PP(i,Indice(I2,i))=n-I2+1; end
if k>1; D=min(PP)'; else D=PP'; end
```

A.2 The random Tukey depth

A.2.1 Definition and main properties

To carry out Example 4.1.4 we have used the following program, which calls `RandomDepthV.m`.

```
clear all
%Generating the sample, matrix A and vector in which to project
X=randn(3,2); A=randn(2,2); vi=randn(1,2); vi=vi/(norm(vi));
%Computing the depths of the points in the sample before and after
%multiplying by A
D=RandomDepthV(X,vi); DA=RandomDepthV((A*X')',vi);
```

`RandomDepthV.m`

```
% Input data:  X = a data set, for example X=randn(n,p), in which the
% number of rows (=n) is the sample size and the number of columns
% (=p) is the dimension.
%              vi = the vector in which to project, of dimension p.
% Output data: D = vector with the ranks associated to the depths of
% the points in the data set.
function D=RandomDepthV(X,vi)
n=length(X(:,1));
% Vectors I1 and I2 are intended to compute the depths (ranks) in
% each projection. We give depth(rank) = 1 to the most outer point
%                          depth(rank) = 2 to the next one
%                          depth(rank) = integer part of n/2 to the
% deepest point
D=zeros(n,1); I1=1:n/2; I2=n:-1:(n/2+.1); prod=X*vi';
[SDat Indice]=sort(prod); D(Indice(I1))=I1; D(Indice(I2))=n-I2+1;
```

A.3 Applications of the random Tukey depth

A.3.1 How many random projections?

The next program gives us as output Figure 5.1. It uses the function `funcion.m` which calls `RandomDepth.m` and `covan.m`.

```
clear all
% Paremeters for the figure
A=[0 0 1 2]; NX=[25 100 50 100 100 500 500]; P=[2 2 8 8 50 50 50];
veces=25; Ln=[25 25 100 100 100 500 500]; Sigma=[.9 0 0 0];
B=[1 1 1 1 1 1 0]; j=1; f=1;
% Plotting
while f≤7
    nX=NX(f); p=P(f); ln=Ln(f); b=B(f);
    for i=1:4
        r=funcion(A(i),nX,p,veces,ln,Sigma(i),b); subplot(7,4,j);
        j=j+1; plot(1:ln,r,'-')
    end
    f=f+1;
end
```

`funcion.m`

```
function r=funcion(a,nX,p,veces,numerovectores,sigma,b)
% Computes the Spearman correlation coefficients between:
%       -the M* depth of a sample, given by a and sigma, in R^p of
% size nX and
%       -each random Tukey depth calculated with a number of
%       projections among two and numerovectores, of the same sample.
% Imput data:  a = 0 for Gaussian marginals, 1 for double exponential
```

```
% marginals and 2 for Cauchy marginals
%              nX = sample size
%              p = space dimension
%              numerovectores = maximum number of projections we use.
% We start with two and continue until this number
%              sigma = covariance between marginals
%              veces = number of times we do the process
%              b = 0 to compute the Mahalanobis depth with the exact
% value of the covariance matrix and 1 to compute it with the sample
% one.
% Output data: r = a number of times, veces, Spearman correlation
% coefficients between the M* depth and each of the random Tukey
% depths
ln=numerovectores; r=zeros(ln,veces); pMh=zeros(1,nX);
S=(1-sigma)*eye(p)+sigma*ones(p,p);  ceros=zeros(1,p);
for j=1:veces
        %Distribution
    if a==0
        X=randn(nX,p)*(S^(.5)); mYY=mean(X);
        if b==0; Si=inv(S); mYY=ceros; else Si=inv(cov(X)); end
        % Mahalanobis depth
        for i=1:nX; xMh=X(i,:)-mYY; pMh(i)=inv(1+xMh*Si*xMh'); end
    elseif a==1
        X=exprnd(1,nX,p)*(S^(.5)); FX=find(rand(nX,p)>1/2);
        X(FX)=-X(FX); mYY=median(X);
        if b==0; Si=inv(S); mYY=ceros; else Si=inv(cov(X)); end
        % M^* depth
        for i=1:nX;
            xMh=X(i,:)-mYY; A=Si^(1/2)*xMh';
```

```
            pMh(i)=inv(1+abs(A(1))+abs(A(2)));
        end
    else
        X=trnd(1,nX,p)*(S^(.5)); mYY=median(X);
        if b==0; Si=inv(S); mYY=ceros; else Si=inv(covan(X)); end
        % M^* depth
        for i=1:nX;
            xMh=X(i,:)-mYY; A=Si^(1/2)*xMh';
            pMh(i)=inv(1+(1+(A(1))^2)*(1+(A(2))^2));
        end
    end
    for i=1:nX; xMh=X(i,:)-mYY; pMh(i)=inv(1+xMh*Si*xMh'); end
    RD=RandomDepth(X);
    % Random Tukey depths and corelations
    for i=2:ln
        RDp=RandomDepth(X); RD=min(RD,RDp);
        if max(RD)~=min(RD)
            r(i,j)=corr(RD,pMh','type','Spearman');
        end
    end
end
```

RandomDepth.m

```
% Input data:  X = a data set, for example X=randn(n,p), in which the
% number of rows (=n) is the sample size and the number of columns
% (=p) is the dimension.
% Output data: D = random Tukey depth in terms of ranks of X using
% one projection
function D=RandomDepth(X)
```

```
[n p] = size(X); D=zeros(n,1); I1=1:n/2; I2=n:-1:(n/2+.1);
% Vectors I1 and I2 are intended to compute the depths (ranks) in
% each projection. We give depth(rank) = 1 to the most outer point
%                          depth(rank) = 2 to the next one
%                          depth(rank) = integer part of n/2 to the
% deepest point
vi=randn(1,p); vi=vi/(norm(vi)); prod=X*vi';
[SDat Indice]=sort(prod); D(Indice(I1))=I1; D(Indice(I2))=n-I2+1;
```

covan.m

```
% Imput data: W = sample in which the number of rows is the sample
% size and the number of columns is the dimension.
% Output data: C = robust covariance matrix of W defined as in
% Maronna et al. [63, p.  206].
function C=covan(W)
[n pc]=size(W); Dm=zeros(pc,pc); Y=zeros(n,pc);
Uw=zeros(pc,pc); Z=zeros(n,pc); L=zeros(pc,pc);
for i=1:pc; Dm(i,i)=1/mad(W(:,i),1); end
for i=1:n; Y(i,:)=W(i,:)*Dm; end
for j=1:pc
    for k=1:pc
        if k==j
            Uw(j,j)=1;
        else
            Yj=Y(:,j); Yk=Y(:,k);
            Uw(j,k)=1/4*(mad(Yj+Yk,1)^2-mad(Yj-Yk,1)^2);
        end
    end
end
```

```
[E, nada]=eig(Uw);
for i=1:n; Z(i,:)=Y(i,:)*E; end
for i=1:pc; L(i,i)=mad(Z(:,i),1)^2; end
A=inv(Dm)*E; C=A*L*A';
```

Computation time

The program `Tiempo.m` gives us Table 5.1. It uses `RtMh.m` and `MhRt.m`.

`Tiempo.m`

```
% Output data: ResultRT = Time, in seconds, needed to compute the
% random Tukey depth of all the points in a sample; for different
% dimensions and sample sizes.
%              ResuktMH = Time, in seconds, needed to compute the
% random Tukey depth of all the points in a sample; for different
% dimensions and sample sizes.
function [ResultRT,ResultMH]=Tiempo
c=[2 8 50 ; 10 60 250]; d=[25 50 100 ; 100 100 500];
ResultRT=zeros(2,3); ResultMH=zeros(2,3);
for j=1:3;
    p=c(1,j); nV=c(2,j);
    for i=1:2
        nX=d(i,j);
        [MedioRT MHMedio]=MhRt(p,nV,nX);
        [RTMedio MedioMH]=RtMh(p,nV,nX);
        ResultRT(i,j)=mean([MedioRT RTMedio]);
        ResultMH(i,j)=mean([MHMedio MedioMH]);
    end
end
```

RtMh.m

```
% Calculate the average time for 100 samples needed to compute the
% random Tukey depth and the Mahalanobis depths. The random Tukey
% depth is computed first.
% Imput data:  p = dimension
%              nV = number of random vectors
%              nX = sample size
% Output data: RTEMedio = average time needed to compute the random
% Tukey depth
%              MedioMH = average time needed to compute the
% Mahalanobis depth
function [RTMedio MedioMH]=RtMh(p,nV,nX)
veces=100; I1=1:nX/2; I2=nX:-1:nX/2+.1; bMh=zeros(1,nX);
MH=zeros(1,veces); RTE=zeros(1,veces); PP=zeros(nV,nX);
for t=1:veces
    X=randn(p,nX);
    %Random Tukey
    tic
    V=randn(nV,p); for v=1:nV; Vv=V(v,:); V(v,:)=Vv/norm(Vv); end;
    P=V*X;
    for vv=1:nV
        [SDat Indice]=sort(P(vv,:));
        PP(vv,Indice(I1))=I1; PP(vv,Indice(I2))=nX-I2+1;
    end
    ProfRT=min(PP)./nX; RTE(t)=toc; Y=X';
    %Mahalanobis
    tic
    S=inv(cov(Y)); mY=mean(Y);
    for  i=1:nX; xMh=Y(i,:)-mY; bMh(i)=inv(1+xMh*S*xMh'); end;
```

```
    MH(t)=toc;
end
RTMedio=mean(RTE);MedioMH=mean(MH);
```

MhRt.m

```
% Calculate the average time for 100 samples needed to compute the
% random Tukey depth and the Mahalanobis depths. The Mahalanobis
% depth is computed first.
% Imput data:  p = dimension
%              nV = number of random vectors
%              nX = sample size
% Output data: RTEMedio = average time needed to compute the random
% Tukey depth
%              MedioMH = average time needed to compute the
% Mahalanobis depth
function [MedioRT MHMedio]=MhRt(p,nV,nX)
veces=100; I1=1:nX/2; I2=nX:-1:nX/2+.1; bMh=zeros(1,nX);
MH=zeros(1,veces); RTE=zeros(1,veces); PP=zeros(nV,nX);
for t=1:veces
    X=randn(p,nX); Y=X';
    % Mahalanobis
    tic
    S=inv(cov(Y)); mY=mean(Y);
    for  i=1:nX; xMh=Y(i,:)-mY; bMh(i)=inv(1+xMh*S*xMh'); end;
    MH(t)=toc;
    % Random Tukey
    tic
    V=randn(nV,p); for v=1:nV; Vv=V(v,:); V(v,:)=Vv/norm(Vv); end;
    P=V*X;
```

```
    for vv=1:nV
        [SDat Indice]=sort(P(vv,:));
        PP(vv,Indice(I1))=I1; PP(vv,Indice(I2))=nX-I2+1;
    end
   ProfRT=min(PP)./nX; RTE(t)=toc;
end
MHMedio=mean(MH); MedioRT=mean(RTE);
```

A.3.2 Multidimensional random Tukey depth. Testing homogeneity

With the programs in this section we obtain Figure 5.2 and Tables 5.2, 5.4, 5.3, 5.5, 5.6 and 5.7. Regarding Figure 5.2, it has been plotted using the following program which calls `Depth.m` , described in Section A.1.

```
% Generates two sample size with a change of scale, computes the
% Tukey random Tukey depths of the joint sample sample and plots the
% results.
% Setting:
p=2; nX=100; nV1=5; V1=rand(p,nV1);
X=randn(nX,p); Y=2*randn(2*nX,p); mX=median(X); mY=median(Y);
for j=1:nX; X(j,:)=X(j,:)-mX; Y(j,:)=Y(j,:)-mY; end; W=[X ; Y];
% Tukey and random Tukey depths
IndiceTuc=0:pi/1000:pi; u=[cos(IndiceTuc) ; sin(IndiceTuc)];
T=Depth(W*u); RT=Depth(W*V1);
% Plotting
s=10; i=0:2*pi/s:2*pi; r=.2; xc=r*cos(i); yc=r*sin(i);
st=3; i=0:2*pi/st:(2*pi-.1); r=.2; xct=r*cos(i); yct=r*sin(i);
figure(1);
for j=1:nX; x=W(j,1)+xc; y=W(j,2)+yc; fill(x,y,RT(j)); hold on; end
```

```
for j=(nX+1):(2*nX);
    x=W(j,1)+xct; y=W(j,2)+yct; fill(x,y,RT(j)); hold on
end
hold off
figure(2)
for j=1:nX; x=W(j,1)+xc; y=W(j,2)+yc; fill(x,y,T(j)); hold on; end
for j=(nX+1):(2*nX);
    x=W(j,1)+xct; y=W(j,2)+yct; fill(x,y,T(j)); hold on
end
```

For Tables 5.2, 5.4 and 5.3 we use `LiuGrande.m` which calls `Bootstrap.m`, `Depth.m` and `test.m`. In addition, Bootstrap.m calls `MedianaPonderadaM.m`. The code of `Depth.m` is in Section A.1.

`LiuGrande.m`

```
% Input data:  a = 0; Normal
%              a = 1; DExp
%              a = 2; Cauchy
%              b = 0; Tukey depth
%              b = 1; random Tukey depth
% Output data: MD = median of the number of vectors used in the
% random Tukey case and zero in the Tukey case.
%              Rate = Rejection Rates
function [MD,Rate]=LiuGrande(a,b)
% Parameters
TM=[20 30 100]; RY=[349 804 9377]; Sigma=[1 1.2 2]; p=2;
repeticiones=5000; LTM=length(TM); LS=length(Sigma); LTMS=LTM*LS;
if  b==1
    R=[1.1 1.2 1.3 1.5 1.7 2 2.5 3 4 5 7 10];
    nV=25; Boots=100; ob=.8*Boots;  met=4;
```

```
    rboots=zeros(repeticiones, Boots); k=zeros(1,Boots);
else
        pm=pi/1000; met=1;
    IndiceTuc=pm:pm:pi; u=[cos(IndiceTuc) ; sin(IndiceTuc)]';
end
MD=zeros(LTMS,met); Rate=zeros(LTMS,met); K0=zeros(repeticiones,met);
 for nn=1:LTM
    nX=TM(nn); Ry=RY(nn);
    for ss=1:LS
        sigma=Sigma(ss); rate=zeros(1,met);
        for repet=1:repeticiones
             % Distribution
             if a==0
                 X=randn(p,nX); Y=sigma.*randn(p,nX);
             elseif a==1
                 X=exprnd(1,p,nX); FX=find(rand(p,nX)>1/2);
                 Y=sigma.*exprnd(1,p,nX); FY=find(rand(p,nX)>1/2);
                 X(FX)=-X(FX); Y(FY)=-Y(FY);
             else
                 X=trnd(1,p,nX); Y=sigma.*trnd(1,p,nX);
             end
             % Centering
             mX=median(X,2); mY=median(Y,2);
             for j=1:nX; X(:,j)=X(:,j)-mX; Y(:,j)=Y(:,j)-mY; end
             W=[X,Y];
             if b==1
                V=randn(nV,p);
                % Selecting the number of vectors
                for i=1:Boots
```

```
                    [k(i) rboots(repet,i)]=Bootstrap(2*nX,nV,W,V,Ry,R);
                end
                ks=sort(k);
                k0=[mean(k) median(k) max(k) ks(round(ob))];
                K0(repet,:)=k0; rk=round(k0);
                % The test
                for rn=1:met
                    pa=Depth((V(1:rk(rn),:)*W)');
                    result=test(Ry,pa); rate(rn)=rate(rn)+result;
                end
            else
                pa=Depth((u*W)'); result=test(Ry,pa);
                rate=rate+result;
            end
        end
        Cn=(nn-1)*LS+ss; MD(Cn,:)=median(K0); Rate(Cn,:)=rate;
    end
end
Rate=Rate./repeticiones;
```

Bootstrap.m

```
% Input data:  N = sample size of the joint sample
%              nV = maximum number of vectors to be chosen
%              W = joint sample
%              V = vectors that are used to compute the random Tukey
% depth
%              Ry = constant that gives the critical region
%              R = grid
% Output data: k = number of vectors chosen by the bootstrap method
```

```
%              r = element of the grid that gave us the k
function [k,r]=Bootstrap(N,nV,W,V,Ry,R)
Nd=N/2; lR=length(R);
Xs=W(:,ceil(N*rand(1,Nd))); Ys=W(:,ceil(N*rand(1,Nd)));
mX=MedianaPonderadaM(Xs,Nd); mY=MedianaPonderadaM(Ys,Nd);
% Centering
for j=1:N/2; Xs(:,j)=Xs(:,j)-mX; Ys(:,j)=Ys(:,j)-mY; end
cte=0; cont=0;
while cte==0
    if cont<lR; cont=cont+1; r=R(cont); else r=r+10; end;
    Z=[Xs,r*Ys];
    for j=1:nV
         pa=Depth((V(1:j,:)*Z)');
         if test(Ry,pa)==1; k=j; cte=1; break; end
    end
end
```

test.m

```
% Input data:  Ry = constant that gives the critical region
%              pa = rank of each element of the sample associated to
% the depth
% Output data: rate = result of the Wilcoxon rank-sum test applied to
% the input data pa
function rate=test(Ry,pa)
N=length(pa); nX=N/2; cn=1;
% Tie-breaking random
for c=1:max(pa)
     F=find(pa==c); bn=length(F)+cn; permu=randperm(bn-1);
     b(F)=permu(find(permu≥cn)); cn=bn;
```

```
end
rate=0; if sum(b(nX+1:N))≤Ry; rate=1; end
```

MedianaPonderadaM.m

```
% Input data:  X = sample
%              Nd = sample size
% Output data: mX = ponderate median of X
function mX=MedianaPonderadaM(X,Nd)
mX=median(X,2);
for h=1:2
    m=mX(h); Xh=X(h,:); fc=find(Xh<m); fg=find(Xh>m);
    sl=Nd-length(fc)-length(fg);
    if sl>1
        xfc=Xh(fc); xfg=Xh(fg); c=(max(xfc)+m)/2;
        mX(h)=median([xfc xfg c+ rand(1,sl)*((min(xfg)+m)/2-c)]);
    end
end
```

For Tables 5.5, 5.6 and 5.7 we use the function `LiuChica.m`. It calls `BootstrapW.m` and `testWallis.m`. In addition, `Depth.m` and `MedianaPonderadaM.m` are also called. `Depth.m` appears in Section A.1 and `MedianaPonderadaM.m` just above here.

LiuChica.m

```
% Imput data:  b = 0; Tukey depth
%              b = 1; random Tukey depth
% Output data: MD = median of the number of vectors used in the
% random Tukey case and zero in the Tukey case.
%              Rate = Rejection Rates
function [MD,Rate]=LiuChica(b)
% Parameters
```

```
TM=[20 30]; Sigma2=[1 1.2 1.2 2]; Sigma3=[1 1.2 2 2]; p=2;
repeticiones=5000; LTM=length(TM); LS=length(Sigma2); LTMS=LTM*LS;
if b==1
    R=[1.1 1.2 1.3 1.5 1.7 2 2.5 3 4 5 7 10]; nV=25; Boots=100;
    met=4; rboots=zeros(repeticiones, Boots); k=zeros(1,Boots);
else
    IndiceTuc=0:pi/1000:pi; u=[cos(IndiceTuc) ; sin(IndiceTuc)]';
    met=1;
end
MD=zeros(LTMS,met); Rate=zeros(LTMS,met); K0=zeros(repeticiones,met);
for nn=1:LTM
    nX=TM(nn);
    for ss=1:LS
        sigma2=Sigma2(ss); sigma3=Sigma3(ss); rate=zeros(1,met);
        for repet=1:repeticiones
        % Distribution
        X=randn(p,nX); Y=sigma2.*randn(p,nX); Z=sigma3.*randn(p,nX);
        mX=median(X,2); mY=median(Y,2);  mZ=median(Z,2);
             % Centering
             for j=1:nX
                 X(:,j)=X(:,j)-mX; Y(:,j)=Y(:,j)-mY;
                 Z(:,j)=Z(:,j)-mZ;
             end
             W=[X,Y,Z];
             if b==1
                 V=randn(nV,p);
                 % Selecting the number of vectors
                 for i=1:Boots
                    [k(i) rboots(repet,i)]=BootstrapW(nX*3,nV,W,V,R);
```

```
                end
                ks=sort(k);
                k0=[mean(k) median(k) max(k) ks(round(.8*Boots))];
                K0(repet,:)=k0; rk=round(k0);
                % The test
                for rnn=1:met
                    pa=Depth((V(1:rk(rnn),:)*W)');
                    result=testWallis(pa);
                    rate(rnn)=rate(rnn)+result;
                end
            else
                pa=Depth((u*W)'); result=testWallis(pa);
                rate=rate+result;
            end
        end
        Cn=(nn-1)*LS+ss; MD(Cn,:)=median(K0); Rate(Cn,:)=rate;
    end
end
Rate=Rate./repeticiones;
```

BootstrapW.m

```
% Input data:  N = sample size of the joint sample
%              nV = maximum number of vectors to be chosen
%              W = joint sample
%              V = vectors that are used to compute the random Tukey
% depth
%              R = grid
% Output data: k = number of vectors chosen by the bootstrap method
%              r = element of the grid that gave us the k
```

```
function [k r]= BootstrapW(N,nV,W,V,R)
Nt=N/3; lR=length(R);
Xs=W(:,ceil(N*rand(1,Nt))); Ys=W(:,ceil(N*rand(1,Nt)));
Zs=W(:,ceil(N*rand(1,Nt)));
mX=MedianaPonderadaM(Xs,Nt); mY=MedianaPonderadaM(Ys,Nt);
mZ=MedianaPonderadaM(Zs,Nt);
% Centering
for j=1:N/3
    Xs(:,j)=Xs(:,j)-mX; Ys(:,j)=Ys(:,j)-mY; Zs(:,j)=Zs(:,j)-mZ;
end
cte=0; cont=0;
while cte==0
    if cont<lR; cont=cont+1; r=R(cont); else r=r+10; end
    Z=[Xs,r*Ys,Zs];
    for j=1:nV
         pa=Depth((V(1:j,:)*Z)');
         if testWallis(pa)==1; k=j; cte=1; break; end
    end
end
```

testWallis.m

```
% Input data:  pa = rank of each element of the sample associated to
% the depth
% Output data: rate = result of the Kruskal-Wallis test applied to
% the input data pa
function rate=testWallis(pa)
N=length(pa); nX=N/3; dnX=2*nX; Nm=N+1; cn=1;
% Tie-breaking random
for c=1:max(pa)
```

```
    F=find(pa==c); bn=length(F)+cn; permu=randperm(bn-1);
    b(F)=permu(find(permu>=cn)); cn=bn;
end
rate=0; Rbarra=[mean(b(1:nX)),mean(b(nX+1:dnX)),mean(b(dnX+1:N))];
T=(4/Nm)*sum((Rbarra-((N+1)/2)).^2); if T>=5.991; rate=1; end
```

A.3.3 Functional random Tukey depth. Functional classification

Through this Section we use the following program `Alturas.m` that loads the file `heights.mat`. It gives a matrix `X` formed by two groups. First group consists in 54 curves (rows) and second in 39 curves (rows), each measured at 31 times (columns).

`Alturas.m`

```
% This program loads the data in the file heights.mat, which contains
% the matrix F with 54 curves measured 31 times and the matrix M
% with 39 curves measured 31 times.
% Output data: X = matrix that contains all the curves
%              nA = number of curves of girls
%              nO = number of curves of boys
%              n = total number of curves
%              MatDist = the distance matrix of the curves in X
function [X,nA,nO,n,p,MatDist]=Alturas
load heights;
X=[F;M]; nA=length(F); nO=length(M); n=length(X(:,1));
MatDist=zeros(n);
for i=1:n; for j=1:n; MatDist(i,j)=norm(X(i,:)-X(j,:)); end; end
```

The following program `Graph.m` gives us Figures 5.3 and 5.4.

```
clear all
```

```
[X,nA,nO,n,p,Dist]=Alturas; F=X(1:nA,:); M=X(nA+1:n,:);
t=[1:0.25:2, 3:8, 8.5:0.5:18];
% Plots the height curves of boys and girls
plot(t,F), xlabel('Age in years', 'FontSize',16),
ylabel('Height in centimeters of girls', 'FontSize',16)
figure, plot(t,M), xlabel('Age in years', 'FontSize',16)
ylabel('Height in centimeters of boys', 'FontSize',16)
% Plots the height curves of boys and girls with each group centered
% by its component-wise median
Mm=median(M); Mt=zeros(nO,p); for i=1:nO; Mt(i,:)=M(i,:)-Mm; end
Fm=median(F); Ft=zeros(nA,p); for i=1:nA; Ft(i,:)=F(i,:)-Fm; end
figure, plot(t,Mt,'b'),
hold on
plot(t,Ft,'r'),
xlabel('Age in years', 'FontSize',16)
ylabel('Centered height of girls and boys', 'FontSize',16)
```

Program `ClassificationCurve.m` gives us Tables 5.8 and 5.9. To compute the rejection rates in the case $S_{0,0}$ we do `ClassificationCurve(0,0)` and in the case $S_{a,c}$ we do `ClassificationCurve`$([0,1],[0,1,5])$. This program needs of the following programs that appear bellow: `Alturas.m`, `restart1.m`, `DepthP.m` and `pesoMpon.m`.

`ClassificationCurve.m`

```
% Computes the rates of mistakes using several classification methods
% based on the random Tukey depth. The distribution
% used to select the curves in which the data is projected depend on
% the input, a and b. The data is loaded inside the program.
% Input data:  a=a vector with the possibilites for a in S_{a,c}
%              c=a vector with the possibilites for c in S_{a,c}
% Output data: M1=vector caontaining the average rates of mistakes
```

```
% when computing the methods M, AM, TAM, MD, DD and MI.
%              k=vector containing the median of the vectors used in
% each of the first five methods
function [M1 k]=ClassificationCurve(a,c)
[X,nA,nO,n,p,M]=Alturas; CrosV=25; veces=100;
randV=[1:4,6:2:12,15:5:29,30:10:59,60:20:100];
% randV=[1:4,6:2:12,15:5:29,30:10:59,60:20:99,100:25:199,200:50:350,
% 400:100:1000];
sI=zeros(1,p); sI(2:5)=.25; sI(6:11)=1; sI(12:p)=.5; dev=sqrt(sI);
sI(1)=.125; sI(5)=.625; sI(11)=.75;
uma=.8; nu=n-1; nm=n-2; aci=zeros(veces,5);
K=zeros(veces*n,5); MaxV=length(randV); MaxK=max(randV);
lena=length(a); lenc=length(c); total=lena*lenc;
for kk=1:veces
    T= ones(n,5);
    for v=1:n
        [nTA nTO TraA TraO]=restar1(nA,nO,1:n,v); Trai=[TraA TraO];
        XT=X(Trai,:); XTA=X(TraA,:); XTO=X(TraO,:); XV=X(v,:);
        Val2=randperm(nu); Valid2=Val2(1:CrosV);
        PTin=zeros(nu,MaxK,total); PVin=zeros(1,MaxK,total);
        ind=0; ABC=zeros(total,2); h=zeros(MaxK,p);
        for ia=1:lena
            for ic=1:lenc
                for i=1:MaxK;
                    h(i,:)=BrownC(p,XTA,XTO,a(ia),c(ic),dev).*sI;
                end
                ind=ind+1; PTin(:,:,ind)=XT*h';
                PVin(:,:,ind)=XV*h'; ABC(ind,:)=[a(ia),c(ic)];
```

```
    end
end
Max=zeros(1,5); MaxInCV=zeros(1,5); MaxIndk=zeros(1,5);
for inCV=1:total
    for Indk=1:MaxV
        k0=randV(Indk); ac=zeros(1,5);
        for ii2=1:CrosV
            i2=Valid2(ii2); PT2=PTin(:,:,inCV);
            PZ=PT2(i2,1:k0); PT2(i2,:)=[];
            [nTA2 nTO2 TraA2 TraO2]=restar1(nTA,nTO,Trai,i2);
            PTA2=PT2(1:nTA2,1:k0);
            PTO2=PT2(nTA2+1:nTA2+nTO2,1:k0);
            DepkA=DepthP(PTA2); DepkO=DepthP(PTO2);
            % M
            x=X(Trai(i2),:);
            [SA Aj]=sort(DepkA,'descend');
            [SO Oj]=sort(DepkO,'descend');
            umaA=1:floor(nTA2*uma); umaO=1:floor(nTO2*uma);
            MDisA=norm(x-mean(X(TraA2(Aj(umaA)),:)));
            MDisO=norm(x-mean(X(TraO2(Oj(umaO)),:)));
            if (i2<=nTA&&MDisA<MDisO)||(i2>nTA&&MDisO<MDisA)
                ac(1)=ac(1)+1;
            end
            % AM
            cteA=sum(DepkA); cteO=sum(DepkO);
            pesoA=pesoMpon(M, Trai(i2), TraA2,DepkA);
            pesoO=pesoMpon(M, Trai(i2), TraO2,DepkO);
            pesoA=pesoA/cteA; pesoO=pesoO/cteO;
            if (i2<=nTA&&pesoA<pesoO)||(i2>nTA&&pesoO<pesoA)
```

```
    ac(2)=ac(2)+1;
end
% TAM
[DepkAsort IndAsort]=sort(DepkA);
s=IndAsort(nTA2:-1:nTA2-nTO2+1);
cteA=sum(DepkA(s));
pesoA=pesoMpon(M,Trai(i2),TraA2(s),DepkA(s));
pesoA=pesoA/cteA;
if (i2≤nTA&&pesoA<pesoO)||(i2>nTA&&pesoO<pesoA)
    ac(3)=ac(3)+1;
end
% MD
PTA2masi2=[PTA2; PTin(i2,1:k0,inCV)];
PTO2masi2=[PTO2; PTin(i2,1:k0,inCV)];
DepkAmasi2=DepthP(PTA2masi2);
DepkOmasi2=DepthP(PTO2masi2);
DA2=DepkAmasi2(nTA2+1);
DO2=DepkOmasi2(nTO2+1);
if (i2≤nTA && DA2>DO2) || (i2>nTA && DO2>DA2)
    ac(4)=ac(4)+1;
end
% DD
[Ratio RatioZ]=DDt(k0,nTA2,nTO2,PTA2,PTO2,PZ);
[sr so]=sort(Ratio); rv=1; d=nTA2; D=nTA2;
for i=2:nm
    im=i-1;
    if so(im)≤nTA2;
        d=d-1;
        if sr(i)≠sr(im) && d<D; rv=i; D=d; end
```

```
                    else
                        d=d+1;
                    end
                end
                r=mean(sr(rv-1:rv));
                if (RatioZ<r && i2≤nTA2)||(RatioZ>r && i2>nTA2)
                    ac(5)=ac(5)+1;
                end
            end
            for i=1:5
                if ac(i)>Max(i)
                    Max(i)=ac(i);
                    MaxInCV(i)=inCV; MaxIndk(i)=Indk;
                end
            end
        end
    end
    po=(kk-1)*n+v;
    % M
    kM=randV(MaxIndk(1)); Ma=MaxInCV(1);
    PkTA=PTin(1:nTA,1:kM,Ma);
    PkTO=PTin(nTA+1:nTA+nTO,1:kM,Ma);
    DepkTA=DepthP(PkTA); DepkTO=DepthP(PkTO);
    [SAT ATj]=sort(DepkTA,'descend');
    [SOT OTj]=sort(DepkTO,'descend');
    xT=X(v,:);
    MDisA=norm(xT-mean(X(TraA(ATj(1:floor(nTA*uma))),:)));
    MDisO=norm(xT-mean(X(TraO(OTj(1:floor(nTO*uma))),:)));
    if MDisO < MDisA; T(v,1)=0; end;
```

```
% AM
kAM=randV(MaxIndk(2)); Ma=MaxInCV(2);
PkTA=PTin(1:nTA,1:kAM,Ma); DepkTA=DepthP(PkTA);
PkTO=PTin(nTA+1:nTA+nTO,1:kAM,Ma); DepkTO=DepthP(PkTO);
cteA=norm(DepkTA,1); cteO=norm(DepkTO,1);
pesoA=pesoMpon(M,v,TraA,DepkTA); pesoA=pesoA/cteA;
pesoO=pesoMpon(M,v,TraO,DepkTO); pesoO=pesoO/cteO;
if pesoA > pesoO; T(v,2)=0; end;
% TAM
kTAM=randV(MaxIndk(3)); Ma=MaxInCV(3);
PkTA=PTin(1:nTA,1:kTAM,Ma); DepkTA=DepthP(PkTA);
PkTO=PTin(nTA+1:nTA+nTO,1:kTAM,Ma); DepkTO=DepthP(PkTO);
[DepkTAsort IndTAsort]=sort(DepkTA);
SelecTA=IndTAsort(nTA:-1:nTA-nTO+1);
cteA=sum(DepkTA(SelecTA)); cteO=sum(DepkTO);
pesoA=pesoMpon(M,v,TraA(SelecTA),DepkTA(SelecTA));
pesoO=pesoMpon(M,v,TraO,DepkTO);
pesoA=pesoA/cteA; pesoO=pesoO/cteO;
if pesoA > pesoO; T(v,3)=0; end
% MD
kMD=randV(MaxIndk(4)); Ma=MaxInCV(4);
PkTA=PTin(1:nTA,1:kMD,Ma); PkTO=PTin(nTA+1:nTA+nTO,1:kMD,Ma);
PTAmasi1=[PkTA; PVin(1,1:kMD,Ma)];
PTOmasi1=[PkTO; PVin(1,1:kMD,Ma)];
DepkAmasi1=DepthP(PTAmasi1); DepkOmasi1=DepthP(PTOmasi1);
DepAi1=DepkAmasi1(nTA+1); DepOi1=DepkOmasi1(nTO+1);
if DepOi1 > DepAi1; T(v,4)=0; end
% DD
kR=randV(MaxIndk(5)); MR=MaxInCV(5); PZ=PVin(1,1:kR,MR);
```

```
        PkTA=PTin(1:nTA,1:kR,MR); PkTO=PTin(nTA+1:nu,1:kR,MR);
        [Ratio RatioZ]=DDt(kR,nTA,nTO,PkTA,PkTO,PZ);
        [sr so]=sort(Ratio); rv=1; d=nTA; D=nTA;
        for i=2:nu
            im=i-1;
            if so(im)≤nTA;
                d=d-1; if sr(i)≠sr(im) && d<D; rv=i; D=d; end
            else
                d=d+1;
            end
        end
        if RatioZ>mean(sr(rv-1:rv)); T(v,5)=0; end
        K(po,:)=[kM kAM kTAM kMD kR];
    end
    B=zeros(nO,5); B(find(T(nA+1:n,:)==0))=1; T(nA+1:n,:)=B;
    aci(kk,:)=sum(T);
end
M1=1-mean(aci)/n; k=median(K);
```

restar1.m

```
% Given a sample with the first nTA elements from the first group
% and an element, i2, of the sample, this program deletes this
% element from the sample.
% Input data:  nTA = number of elements of the group A in the
% initial sample
%              nTO = number of elements of the group O in the
% initial sample
%              Training = initial sample, formed by group A and O
%              i2 = element of the initial sample we want to delete
```

```
% from it
% Output data: nTA2 = number of elements from the group A in the
% sample after deleting i2
%              nTO2 = number of elements from the group O in the
% sample after deleting i2
%              TraA2 = elements from the group A in the sample after
% deleting i2
%              TraO2 = elements from the group O in the sample after
% deleting i2
function [nTA2 nTO2 TraA2 TraO2]=restar1(nTA,nTO,Training,i2)
TraA2 =Training(1:nTA); TraO2=Training(nTA+1:nTA+nTO);
if i2>nTA;  TraO2(i2-nTA)=[]; nTA2=nTA; nTO2=nTO-1;
else TraA2(i2)=[]; nTA2=nTA-1; nTO2=nTO;
end
```

BrownC.m

```
% Input data:  p = number of times at which the curves are measured
%              F = matrix with the curves of the group A. It has p
% columns
%              M = matrix with the curves of the group B. It has p
% columns
%              a = parameter of S_{a,c}
%              c = parameter of S_{a,c}
% Output data: B = S_{a,c} defined in Subsection 5.3.1
function Brown=BrownC(p,F,M,a,c,dev)
Brown=zeros(1,p); mF=median(F); mM=median(M); Brown(1)=c;
for i=2:p
   Brown(i)=Brown(i-1)+((mF(i)-mM(i))^a)*randn*dev(i);
end
```

DepthP.m

```
% Input data:  Prod = matrix containing the 1-dimensional projections
% of n points on k randomly chosen vectors.
% Output data: D = vector with the ranks associated to the depths of
% the n points.
function D=DepthP(Prod)
[n k]=size(Prod);
% Vectors I1 and I2 are intended to compute the depths (ranks) in
% each projection. We give depth(rank) = 1 to the most outer point
%                          depth(rank) = 2 to the next one
%                          depth(rank) = integer part of n/2 to the
% deepest point
I1=1:n/2; I2=n:-1:(n/2+.1); PP=zeros(k,n); [SDat Indice]=sort(Prod);
for i=1:k; PP(i,Indice(I1,i))=I1; PP(i,Indice(I2,i))=n-I2+1; end
if k>1; D=min(PP)'; else D=PP'; end; D=D/n;
% Taking into account that the maximum theoretical depth is 1/2, we
% divide by n and so, then the depths belong to the interval [1/n,1/2]
```

pesoMpon.m

```
% Input data:  MatDist = distance matrix of the total sample
%              Validat = element of the total sample
%              TraA = elements of the group A
%              DepkA = depth of the sample TraA
% Output data: pes = distance between Validat and the group A as a
% weighted mean of the distances between Validat and the members of
% the group where the weights are the depths of the points
function pes=pesoMpon(MatDist,Validat,TraA,DepkA)
```

```
nVal=length(Validat); nTra=length(TraA); pes=zeros(nVal,1);
for i2=1:nVal
    uno=Validat(i2);
    for i3=1:nTra
        otro=TraA(i3);
        pes(i2)=pes(i2)+(MatDist(uno,otro)*DepkA(i3));
    end
end
```

DDt.m

```
% Input data:  k = number of vectors used in the computation of
% random Tukey depth
%              nA = number of elements of group A in the training
% sample
%              nO = number of elements of group A in the training
% sample
%              prodA = matrix containing the 1-dimensional
% projections of nA training curves
%              prodO = matrix containing the 1-dimensional
% projections of nO training curves
%              prodZ = matrix containing the 1-dimensional
% projection of the test curve
% Output data: Ratio = Ratio of the DD procedure using the training
% data
%              RatioZ = Ratio of the DD procedure using the test
% data
function [Ratio RatioZ]=DDt(k,nA,nO,prodA,prodO,prodZ)
n=nA+nO;
[DrespA, DZrespA]=D([prodA ; prodZ],prodO,nA,nO,n,k,0,0);
```

```
[DrespO, DZrespO]=D([prodO ; prodZ],prodA,nO,nA,n,k,nA,nO);
Ratio=DrespO./DrespA; RatioZ=DZrespO/DZrespA;
```

D.m

```
% Input data:  prodA = matrix containing the 1-dimensional
% projections of nA training curves and the test curve
%              prodO = matrix containing the 1-dimensional
% projections of nO training curves
%              nA = number of elements of group A in the training
% sample
%              nO = number of elements of group A in the training
% sample
%              n = nA+nO
%              k = number of vectors used in the computation of
% random Tukey depth
%              (e,b) = constants depending on the position of group
% A in the training sample
% Output data: DrespA = random Tukey depth of the training sample
% with respect to the set of curves formed by the test sample and
% the elements of the training in the group A
%              DZrespA = random Tukey depth of the test sample with
% respect to the set of curves formed by the test sample and the
% elements of the training in the group A
function [DrespA, DZrespA]=D(prodA,prodO,nA,nO,n,k,e,b)
DrespA=zeros(1,n); nAZ=nA+1;
[SDat IA]=sort(prodA);
I1A=1:nAZ/2; I2A=nAZ:-1:(nAZ/2+.1); PP=zeros(k,nAZ);
for i=1:k; PP(i,IA(I1A,i))=I1A; PP(i,IA(I2A,i))=nAZ-I2A+1; end
if k>1; DA=min(PP)'; else DA=PP'; end;
```

```
DrespA((1+e):(nA+e))=DA(1:nA); DZrespA=DA(nAZ)/nAZ;
PA=zeros(k,nO);
for i=1:nO
    for j=1:k
        pOi=prodO(i,j); pA=prodA(:,j);
        infA=length(find(pA≤pOi));
        supA=length(find(pA≥pOi));
        PA(j,i)=min(infA,supA);
    end
end
ini=nA+1-b;
fini=n-b;
if k>1; DrespA(ini:fini)=min(PA)'; else DrespA(ini:fini)=PA'; end;
DrespA=DrespA/nAZ; DrespA(find(DrespA==0))=10^(-4);
```

Let us compute now Table 5.10. There, three methods are use, random forest, k-NN and Kernel. As we said in the introduction, the computations of the random forests have been done with a software downloaded from http://www.cs.waikato.ac.nz/ml/weka. Thus, here we write the MatLab codes por k-NN and Kernel.

For k-NN we use the function `dknn.m` which calls `Alturas.m` , `uknn.m` and `tknn.m`. Furthermore, `uknn.m` calls `knno.m`.

`dknn.m`

```
% Input data:  K = 0, we use cross-validation (CV) to select the
% number of nearest neighbors
%              K = 1, we do 1-NN
%              K = 3, we do 3-NN, ..
% Output data: ETotal = rate of mistakes for k-NN when classifying by
% using CV
%              kg = vector of length the number of curves whose
```

```
% components are the k's selected by CV when K=0 and are zero
% otherwise
function [Etotal kg]=dknn(K)
[J,chicas,chicos,m,p,d]=Alturas;
mm=m-1; NV=1:2:91; e=0; kg=zeros(1,m); k=K;
I(1:chicas)=ones(1,chicas); I(chicas+1:m)=zeros(1,chicos);
%In the following we classify the curves by leave-one-out CV
for j=1:m
    XTrain(1:j-1)=1:j-1; XTrain(j:mm)=j+1:m;
    YTrain(1:j-1)=I(1:j-1); YTrain(j:mm)=I(j+1:m);
    %The following line choose the k to be used in k-nn when K=0
    if K==0; k=uknn(XTrain,YTrain,d,NV); kg(j)=k; end
    if tknn(k,XTrain,YTrain,j,d)~=I(j); e=e+1; end
end
Etotal=e/m;
```

tknn.m

```
% K-Nearest-Neighbor-Classifier MatLab Code
% Input data:  k = value of k to do k-NN
%              TrainPattern = the elements we use as Training
%              TrainLabel = labels of the elements we use as
% Training
%              TestPattern = the element for which we want to predict
% its label
%              d = vector of distances between curves
% Output data: PreLabel = Predicted Label using k-nn
function PreLabel=tknn(k,TrainPattern,TrainLabel,TestPattern,d)
N=length(TrainPattern); dr=zeros(1,N);
% Determines distances of all TrainPattern points to the TestPattern
```

```
% point, creating a distance column vector with N rows
for i=1:N; dr(i)=d(TrainPattern(i),TestPattern); end
% The predicted label is the TrainLabel associated with nearest
% TrainPatterns points. This is done by determining the closest
% distances and their indices
[cldvalues clIndx]=sort(dr);
if sum(TrainLabel(clIndx(1:k)))>k/2; PreLabel=1; else PreLabel=0; end
```

uknn.m

```
% Input data:  XTrain = the sample
%              YTrain = the labels of the elements in the sample
%              d = matrix of distances
%              NV = the possible values for K
% Output data: K = is the K to be used in K-nn
function K=uknn(XTrain,YTrain,d,NV)
m=length(YTrain); e=zeros(1,length(NV));
for i=1:m
    XTrain1(1:i-1)=XTrain(1:i-1); XTrain1(i:m-1)=XTrain(i+1:m);
    YTrain1(1:i-1)=YTrain(1:i-1); YTrain1(i:m-1)=YTrain(i+1:m);
    PredictedLabels=knno(XTrain1,YTrain1,XTrain(i),d,NV);
    f=find(PredictedLabels≠YTrain(i)); e(f)=e(f)+1;
end
[a b]=min(e); K=NV(b);
```

knno.m

```
% Input data:  TrainPattern = elements we use as training
%              TrainLabel = labels of the training elements
%              TestPattern = element we want to label
%              d = matrix of distances
```

```
%              NV = possible values of k to do k-nn
% Output data: PredictLabels = Label the procedure gives to
% TestPattern for each of the elements of NV
function PredictLabels=knno(TrainPattern,TrainLabel,TestPattern,d,NV)
N=length(TrainPattern); dr=zeros(1,N);
%creates distance column vector with N rows
for i=1:N; dr(i)=d(TrainPattern(i),TestPattern); end
%determines closest distances and their indices
[cldvalues,clIndx]=sort(dr); n=0; PredictLabels=zeros(1,length(NV));
for k=1:NV
    n=n+1;
    if sum(TrainLabel(clIndx(1:k)))>k/2; PredictLabels(n)=1; end
end
```

For Kernel we use the function `ClasifKernel.m` with $H = 50$ and $a = 0$ for the fifth column of Table 5.10 and $a = 1$ for the sixth. This program calls `Aciert.m`. In addition, it calls `Alturas.m` and `restart1.m` that appear above.

`ClasifKernel.m`

```
% Computes the rates of mistakes using a classification method based
% on kernels. The data is loaded inside the program.
% Input data:  a = 0, when using the indicator kernel, K(u) = I_[0,1](u).
%              a = 1, when using the quadratic kernel,
% K(u) = (1-u^2)I_[0,1](u).
%              H is used in order to choose the window so that we
% apply leave-one-out cross-validation to a grid of H values.
% Output data: fallo = rate of mistakes.
function fallo = ClasifKernel(H,a)
[X,nA,nO,n,p,Dist]=Alturas; Ihh=(1:H)/H; aciertos=zeros(n,1);
for i1=1:n
```

```
    [nTA nTO TraA TraO]=restar1(nA,nO,1:n,i1); Training=[TraA TraO];
    nT=nTA+nTO; Dist2=Dist(Training,Training);
    hM=max(max(Dist2)); hm=min(min(Dist2+hM*eye(nT)));
    % As we have said, the kernel's window is chosen by
    % leave-one-out CV. Vecth is the list of all possible windows.
    Vecth=hm+(hM-hm)*Ihh; ac2=zeros(1,H);
    for i2=1:nT
        [nTA2 nTO2 TraA2 TraO2]=restar1(nTA,nTO,Training,i2);
        Trai2=[TraA2 TraO2]; nT2=nTA2+nTO2; uno=Training(i2);
        for i=1:H
            h=Vecth(i);
            ac2(i)=ac2(i)+Aciert(uno,i2,nT2,Trai2,Dist,h,nTA2,nTA,a);
        end
    end
    [maximo indexh0]=max(ac2); h0=Vecth(indexh0);
    % Note that we classify using the selected window, h0.
    aciertos(i1) = Aciert(i1,i1,nT,Training,Dist,h0,nTA,nA,a);
end
fallo=1-sum(aciertos)/n;
```

Aciert.m

```
% Input data:  uno = sample we test
%              i1 = label of the sample we test with respect to the
% total number of curves
%              nT = number of elements in the training sample
%              Training = training sample
%              Dist = Distance matrix of all the curves
%              h0 = window for the kernel
%              nTA = number of elements of the first class in the
```

```
% training sample
%               nA = number of elements in the first class between the
% training and the test sample.
%               a = 0, when using the indicator kernel, K(u) = I_[0,1](u).
%               a = 1, when using the quadratic kernel,
% K(u) = (1 - u^2) I_[0,1](u).
% Output data: ac = 1 if the test sample is correctly classified
%               ac = 0 if the test sample is wrongly classified
function ac = Aciert(uno,i1,nT,Training,Dist,h0,nTA,nA,a)
ac=0; pesoA=0; pesoO=0; w=0;
for i2=1:nT
    otro=Training(i2); D=Dist(uno,otro);
    if  D<h0;
        if a==1; w=D/h0; end
        if i2≤nTA; pesoA=pesoA+1-w^2; else pesoO=pesoO+1-w^2; end
    end
end
if (i1≤nA && pesoA>pesoO) || (i1>nA && pesoO>pesoA); ac=1; end
```

A.4 Test of Gaussianity for stationary processes

Here, we write the programs used in Section 6.3 to obtain Figures 6.1 and 6.2 and Tables 6.2, 6.3 and 6.4.

To compute Figure 6.1 we use the following program. It also produces Figure 6.3. This program calls `distribucionesp.m` .

```
% This program computes two figures. The first one is four
% possibilities of AR(1) and the second one two processes of the
```

```
% family of non-Gaussian processes with Gaussian families given in
% Cuesta-Albertos and Matrán [22].
clear all; n=1000; past=1000;
% First plot
% first subplot
figure; subplot(4,1,1); q=0; x=distribucionesp(0,1,n+past);
for i=2:n+past; x(i)=q*x(i-1)+x(i); end; x=x(past+1:past+n);
plot(1:n,x);
title('D_\epsilon=N(0,1) , q=0'); xlabel('time'); ylabel('AR(1)')
% second subplot
subplot(4,1,2); q=0; y=distribucionesp(0,7,n+past);
for i=2:n+past; y(i)=q*y(i-1)+y(i); end; y=y(past+1:past+n);
plot(1:n,y);
title('D_\epsilon=\beta(2,1) , q=0'); xlabel('time'); ylabel('AR(1)')
% third subplot
subplot(4,1,3); q=.9; x=distribucionesp(0,1,n+past);
for i=2:n+past; x(i)=q*x(i-1)+x(i); end; x=x(past+1:past+n);
plot(1:n,x);
title('D_\epsilon=N(0,1) , q=.9'); xlabel('time'); ylabel('AR(1)')
% fourth subplot
subplot(4,1,4); q=.9; y=distribucionesp(0,7,n+past);
for i=2:n+past; y(i)=q*y(i-1)+y(i); end; y=y(past+1:past+n);
plot(1:n,y);
title('D_\epsilon=\beta(2,1) , q=.9');
xlabel('time'); ylabel('AR(1)')
% Second plot
% first subplot
figure; subplot(2,1,1); x=distribucionesp(5,8,n); plot(1:n,x)
title('p=5'); xlabel('time'); ylabel('process')
```

```
% second subplot
subplot(2,1,2); x=distribucionesp(13,8,n); plot(1:n,x);
title('p=13'); xlabel('time'); ylabel('process')
```

distribucionesp.m

```
% Input data:  p = in case para=8. This specifies to which
% distribution we refer among the ones in the family given by 8
%              para = a number that refer to a distribution
%              c = sample size
% Output data: x = values generated from the distribution given by
% para and sample size given by c
function x=distribucionesp(p,para,c)
if para==1; x=randn(1,c);
elseif para==2; x=lognrnd(0,1,1,c);
elseif para==3; x=trnd(10,1,c);
elseif para==4; x=chi2rnd(1,1,c);
elseif para==5; x=chi2rnd(10,1,c);
elseif para==6; x=rand(1,c);
elseif para==7; x=betarnd(2,1,1,c);
elseif para==8
    U=randi(p,1)-1; mf=ceil((c+U)/p); Y=randi(p,1)-1;
    Zini=randi(p,mf)-1; mfp=mf*p; Z=zeros(1,mfp);
    for M=0:mf-1
        for K=0:p-1; Z(M*p+K+1)=mod(Zini(M+1)+K*Y,p); end
    end
    mfpU=mfp-U; W=zeros(1,mfpU); for i=1:mfpU; W(i)=Z(i+U); end
    Q=zeros(1,p+1); Q(1)=-inf; Q(p+1)=inf;
    for K=1:p-1; Q(K+1)=norminv(K/p,0,1); end; y=randn(1,mfpU);
```

```
    for m=1:mfpU; K=W(m)+1; We=y(m);
        while We≤Q(K) || We≥Q(K+1); We=randn(1); end; y(m)=We;
    end
    x=y(1:c);
end
```

Figure 6.2 is computed with the following program. This program calls **Gc.m**. In addition **Gc.m** calls the programs **distribucionesp.m**, reproduced above, and **GestadisticoVn.m**.

```
% This program computes the rejection rates under the null hypothesis
% of three AR(1) processes, one with q = 0, other with q = 0.5 and
% another with q = −0.9 using the Lobato and Velasco test for different
% values c and sample sizes.
clear all
% Computation of the rejection rates
Ru=zeros(3,31); Rd=zeros(3,31); Rt=zeros(3,31); s=[100 500 1000];
for i=1:3
    ss=s(i); Ru(i,:)=Gc(1,ss,0,5000);
    Rd(i,:)=Gc(1,ss,.5,5000); Rt(i,:)=Gc(1,ss,−.9,5000);
end
% First plot
figure; subplot(3,1,1)
plot(1:31,Ru(3,:),'b*−'); hold on;
plot(1:22,Ru(2,1:22),'r+−')
plot(1:10,Ru(1,1:10),'yo−')
plot(1:31,.05*ones(1,31),'g')
legend('n=1000', 'n=500', 'n=100');axis([0 35 0.01 0.054])
title('q=0'); xlabel('c'); ylabel('rejection rates')
```

```
% Second plot
subplot(3,1,2)
plot(1:31,Rd(3,:),'b*-'); hold on;
plot(1:22,Rd(2,1:22),'r+-')
plot(1:10,Rd(1,1:10),'yo-')
plot(1:31,.05*ones(1,31),'g')
legend('n=1000', 'n=500', 'n=100'); axis([0 35 0.01 0.054])
title('q=.5'); xlabel('c'); ylabel('rejection rates')
% Third plot
subplot(3,1,3)
plot(1:31,Rt(3,:),'b*-'); hold on;
plot(1:22,Rt(2,1:22),'r+-')
plot(1:10,Rt(1,1:10),'yo-')
plot(1:31,.05*ones(1,31),'g')
legend('n=1000', 'n=500', 'n=100'); axis([0 35 0.01 0.105])
title('q=-.9'); xlabel('c'); ylabel('rejection rates')
```

Gc.m

```
% Input data:  para = distribution we use to compute the AR(1)
% process
%              n = sample size
%              q = parameter of the AR(1) process
%              repetitions = number of times we run the test
% Output data: Rate = rejection rates at level 0.05. of the Lobato and
% Velasco's test for different values of c.
function Rate=Gc(para,n,q,repetitions)
past=1000; N=2; dN=2*N; rate=zeros(1,31); cc=chi2inv(1-.05,dN-2);
if n==100; d=10; elseif n==500; d=22; elseif n==1000; d=31; end
for rep=1:repetitions
```

```
    % Definition of the process
    x=distribucionesp(p,para,n+past);
    for i=2:n+past; x(i)=q*x(i-1)+x(i); end; x=x(past+1:past+n);
    % Computing the statistic
    for c=1:d
        T=GestadisticoVn(x,c); if T>=cc; rate(c)=rate(c)+1; end
    end
end
Rate=rate/repetitions;
```

GestadisticoVn.m

```
% Input data:  y = process we want to test whether is Gaussian
%              c = constant use to know until when the sum involved
% in the Lobato and Velasco test is computed
% Output data: G = statistic of the Lobato and Velasco test for the
% process y using the value given in c
function G=GestadisticoVn(y,c)
n=length(y); me=mean(y); mu2=var(y)*(n-1)/n;
mu3=sum((y-me).^3)/n; mu4=sum((y-me).^4)/n;
hn=ceil(c*sqrt(n)-1); gamma=zeros(1,hn);
for j=1:hn; yt=y(1:n-j); gamma(j)=sum((yt-me).*(y(1+j:n)-me))/n; end
hnm=hn+1; gat=zeros(1,hn); for j=1:hn; gat(j)=gamma(hnm-j); end
F3=abs(2*sum(gamma.*(gamma+gat).^2)+mu2^3);
F4=abs(2*sum(gamma.*(gamma+gat).^3)+mu2^4);
G=n*(mu3^2/(6*F3)+(mu4-3*mu2^2)^2/(24*F4));
```

For the computation of Tables 6.2, 6.3 and 6.4 we use the following program. It calls `GoE.m`, `GE.m` and `testrandom.m`. In turn, they call `GestadisticoVn.m` that can be found above and `Sub.m` that is below. `Sub.m` needs of `Quadratic.m` and `amoebam.m`.

```
Q=[-.9 -.5 0 .5 .6 .7 .8 .9]; lq=length(Q); N=[100 500 1000];
repetitions=5000; E=zeros(lq,7,3); G=zeros(lq,7,3);
ge=zeros(lq,7,3); TR=zeros(lq,7,3);
for j=1:lq
    q=Q(j);
    for para=1:7
        for i=1:3
            n=N(i);
            E(j,para,i)=GoE(2,0,para,n,q,repetitions);
            G(j,para,i)=GoE(1,0,para,n,q,repetitions);
            ge(j,para,i)=GE(0,para,n,q,repetitions);
            TR(j,para,i)=testrandom(4,0,para,n,q,repetitions);
        end
    end
end
% Table 6.2
E(:,:,1)
G(:,:,1)
ge(:,:,1)
TR(:,:,1)
% Table 6.3
E(:,:,2)
G(:,:,2)
ge(:,:,2)
TR(:,:,2)
% Table 6.4
E(:,:,3)
G(:,:,3)
ge(:,:,3)
```

```
TR(:,:,3)
```

GoE.m

```
% Input data:  Test = 1 if we want to apply the Lobato and Velasco's
% test and any other value if we want to apply the Epps' test
%              p = in case para=8, this specifies to which
% distribution we refer among the ones in the family given by 8
%              para = distribution we use to compute the AR(1)
% process
%              n = sample size
%              q = parameter of the AR(1) process
%              repetitions = number of times we run the test
% Output data: Rate = rejection rate at level 0.05. of the Lobato
% and Velasco's test (if Test=1) or the Epps' test (if Test≠1)
function Rate=GoE(Test,p,para,n,q,repetitions)
past=1000; N=2; dN=2*N; rn=floor(n^.4);
rate=0; cc=chi2inv(1-.05,dN-2);
for rep=1:repetitions
    % Definition of the process X
    x=distribucionesp(p,para,n+past);
    for i=2:n+past; x(i)=q*x(i-1)+x(i); end; x=x(past+1:past+n);
    % Statistic
    if Test==1;
        T=GestadisticoVn(x,1);
    else
        deviSt=std(x)*(n-1)/n;
        T=Sub([1 2]/deviSt,x,deviSt,rn,n,dN,N);
    end
    if T≥cc; rate=rate+1; end
```

```
end
Rate=rate/repetitions;
```

GE.m

```
% Input data:  p = in case para=8, this specifies to which
% distribution we refer among the ones in the family given by 8
%               para = distribution we use to compute the AR(1)
% process
%               n = sample size
%               q = parameter of the AR(1) process
%               repetitions = number of times we run the test
% Output data: Rate = rejection rate at level 0.05. of the combination
% using FDR of Epps' test and Lobato and Velasco's test
function Rate=GE(p,para,n,q,repetitions)
v=2; past=1000; N=2; dN=2*N; rn=floor(n^.4); rate=0; Cs=0;
for i=1:v; Cs=Cs+1/i; end; cc=zeros(1,v);
for i=1:v; cc(i)=chi2inv(1-.05*i/(v*Cs),dN-2);end; T=zeros(1,v);
for rep=1:repetitions
    % Definition of the process X
    x=distribucionesp(p,para,n+past);
    for i=2:n+past; x(i)=q*x(i-1)+x(i); end; x=x(past+1:past+n);
    % Statistics
    deviSt=std(x)*(n-1)/n;
    T(1)=GestadisticoVn(x,1);
    T(2)=Sub([abs(randn) 2*abs(randn)]/deviSt,x,deviSt,rn,n,dN,N);
    F=sort(T,'descend');
    for i=v:-1:1; if F(i)≥ cc(i); rate=rate+1; break; end; end
end
Rate=rate/repetitions;
```

testrandom.m

```
% Input data:  v = is the number of projections we use. Half of them
% for the Epps' test and the other half for the Lobato and Velasco's
% test
%              p = in case para=8, this specifies to which
% distribution we refer among the ones in the family given by 8
%              para = distribution we use to compute the AR(1)
% process
%              n = sample size
%              q = parameter of the AR(1) process
%              repetitions = number of times we run the test
% Output data: Rate = rejection rates at level 0.05. of the random
% projection test
function Rate=testrandom(v,p,para,n,q,repetitions)
past=1000; N=2; dN=2*N; rn=floor(n^.4); rate=0; Cs=0;
for i=1:v; Cs=Cs+1/i; end; cc=zeros(1,v);
for i=1:v; cc(i)=chi2inv(1-.05*i/(v*Cs),dN-2);end; T=zeros(1,v);
for rep=1:repetitions
    % Definition of the process X
    x=distribucionesp(p,para,n+past);
    for i=2:n+past; x(i)=q*x(i-1)+x(i); end; x=x(past+1:past+n);
    % Definition of the process Y
    for i=1:v
       if mod(i,2)==1; A=100; B=1; elseif mod(i,2)==0; A=2; B=7; end
       ch=1; HH=betarnd(A,B,1,n); C=n;
       while ch>10^(-15) && C>1;
           a=ch*HH(C); ch=ch-a;
           if C==n; HH(C)=sqrt(a); else HH(C)=sqrt(a)/(n-C); end
           C=C-1;
```

```
        end
        HH(C)=sqrt(ch)/(n-C); h=HH(C:n); k=length(h); y=zeros(1,n);
        for j=1:(k-1); y(j)=x(1:j)*h(k-j+1:k)'; end
        for j=k:n; y(j)=x(j-k+1:j)*h'; end
        % Statistic
        dev=std(y)*(n-1)/n;
        if i≤v/2;
            T(i)=GestadisticoVn(y,1);
        elseif i>v/2;
            T(i)=Sub([abs(randn) 2*abs(randn)]/dev,y,dev,rn,n,dN,N);
        end
    end
    F=sort(T,'descend');
    for i=v:-1:1; if F(i)≥ cc(i); rate=rate+1; break; end; end
end
Rate=rate/repetitions;
```

Sub.m

```
% Input data:  lambda = points at which it is verified whether the
% characteristic function of the process y is equal to the
% characteristic function of a Gaussian distribution
%              y = process at which we compute the statistic
%              deviSt = standard deviation of y
%              rn = number needed in the computation of the
% estimator of the spectral density at zero
%              n = size of y
%              dN = two times N
%              N = number of elemnts of lambda
% Output data: Tr = statistic of the process y using the Epps' test
```

```
function Tr=Sub(lambda,y,deviSt,rn,n,dN,N)
% Definition of gn, the emprical characteristic function computed on
% lambda
gmatrix=zeros(n,dN);
for i=1:n; ly=lambda*y(i); co=cos(ly); si=sin(ly);
    for j=1:N; j2=j*2; gmatrix(i,j2-1:j2)=[co(j), si(j)]; end
end
gn=mean(gmatrix);
% Definition of Gm, the generalized inverse of two times the
% spectral density function at zero
dpifcero=zeros(dN,dN);
for j=1:n; zj=gmatrix(j,:)-gn; dpifcero=dpifcero+zj'*zj; end
de2=zeros(dN,dN);
for r=1:rn
    de1=zeros(dN,dN);
    for j=1:n-r; de1=de1+(gmatrix(j,:)-gn)'*(gmatrix(j+r,:)-gn); end
    de2=de2+de1*(1-r/rn);
end
dpifcero=(dpifcero+2*de2)/n; Gm=pinv(dpifcero); me=mean(y);
sts=deviSt/sqrt(n); ts2=sqrt(2/n); Va=deviSt^2;
P=[me-sts Va*(1-ts2); me+sts Va*(1-ts2); me Va*(1+ts2)];
Y=zeros(3,1);
for i=1:3; Y(i)=Quadratic(P(i,:),gn,lambda,Gm,N,dN); end
Tr=amoebam(P,Y,n,gn,lambda,Gm,N,dN);
```

Quadratic.m

```
% Input data:  m = vector containing a mean and a variance
%              gn = emprical characteristic function computed on
```

```
% lambda
%              lambda = points at which it is verified whether the
% characteristic function of the process y is equal to the
% characteristic function of a Gaussian distribution
%              Gm = generalized inverse of two times the spectral
% density function at zero
%              N = number of elemnts of lambda
%              dN = two times N
% Output data: q = quadratic form needed to compute the statistic of
% Epp's test
function q = Quadratic(m,gn,lambda,Gm,N,dN)
mu=m(1); sigma=m(2);
ml=mu*lambda;
e=exp(-sigma*(lambda.^2)/2);
re=e.*cos(ml); im=e.*sin(ml); gms=zeros(1,dN);
for j=1:N; j2=j*2; gms((j2)-1:j2)=[re(j), im(j)]; end
g=gn-gms;
q=g*Gm*g';
```

amoebam.m

```
% The following program is the translation to MatLab of the program
% amoeba that can be found in Press et al. [71]
% Input data:  P = 3 times 2 matrix containing three pairs of
% inicializations, mean and variance
%              Y = vector of length 3 containig the value of the
% quadratic form involved in the Epps' test
%              n = size of the process we are testing
%              gn = emprical characteristic function of the process
% computed on lambda
```

```
%              lambda = points at which it is checked whether the
% characteristic function of the process y is equal to the
% characteristic function of a Gaussian distribution
%              Gm = generalized inverse of two times the spectral
% density function at zero
%              N = number of elemnts of lambda
%              dN = two times N
% Output data: mf = computes the minimizer of the quadratic function
% nearest the sample mean and variance of the process
function mf=amoebam(P,Y,n,gn,lambda,Gm,N,dN)
NDIM=2; FTOL=.0001; NMAX=20; ALPHA=1; BETA=0.5; GAMMA=2; ITMAX=500;
PR=zeros(1,NMAX); PRR=zeros(1,NMAX); MPTS=NDIM+1; ITER=0; IXXX=1;
while IXXX==1
    ILO=1; if Y(1)>Y(2); IHI=1; INHI=2; else IHI=2; INHI=1; end
    for I=1:MPTS
        if Y(I)<Y(ILO); ILO=I; end
        if Y(I)>Y(IHI); INHI=IHI; IHI=I;
        elseif Y(I)>Y(INHI) && I ≠IHI; INHI=I;
        end
    end
    RTOL=2.*abs(Y(IHI)−Y(ILO))/(abs(Y(IHI))+abs(Y(ILO)));
    if RTOL<FTOL; mf=n*min(Y); break; end
    if ITER==ITMAX; mf=n*min(Y); break; end
    ITER=ITER+1; PBAR=zeros(1,NDIM);
    for I=1:MPTS;
        if I≠IHI; for J=1:NDIM; PBAR(J)=PBAR(J)+P(I,J); end; end
    end
    for J=1:NDIM
        PBAR(J)=PBAR(J)/NDIM;
```

```
    PR(J)=(1.+ALPHA)*PBAR(J)-ALPHA*P(IHI,J);
end
YPR=Quadratic(PR,gn,lambda,Gm,N,dN);
if YPR≤Y(ILO)
    for J=1:NDIM; PRR(J)=GAMMA*PR(J)+(1.-GAMMA)*PBAR(J); end
    YPRR=Quadratic(PRR,gn,lambda,Gm,N,dN);
    if YPRR < Y(ILO);
        for J=1:NDIM; P(IHI,J)=PRR(J); end
        Y(IHI)=YPRR;
    else
        for J=1:NDIM; P(IHI,J)=PR(J); end; Y(IHI)=YPR;
    end
elseif YPR≥Y(INHI)
    if YPR < Y(IHI);
        for J=1:NDIM; P(IHI,J)=PR(J); end
        Y(IHI)=YPR;
    end
    for J=1:NDIM; PRR(J)=BETA*P(IHI,J)+(1.-BETA)*PBAR(J); end
    YPRR=Quadratic(PRR,gn,lambda,Gm,N,dN);
    if YPRR<Y(IHI);
        for J=1:NDIM; P(IHI,J)=PRR(J); end
        Y(IHI)=YPRR;
    else
        for I=1:MPTS;
            if I≠ILO;
                for J=1:NDIM;
                    PR(J)=0.5*(P(I,J)+P(ILO,J)); P(I,J)=PR(J);
                end
                Y(I)=Quadratic(PR,gn,lambda,Gm,N,dN);
```

```
                end
            end
        end
    else for J=1:NDIM; P(IHI,J)=PR(J); end; Y(IHI)=YPR;
    end
end
```

A.4.1 A stationary non-Gaussian process with Gaussian marginal

In this subsection we have the codes to compute Tables 6.5 and 6.6. The code for Figure 6.3 appears at the beginning of this section. The program for Table 6.5 is the following one. It calls `testrandom.m`, reproduced above in the section.

```
P=[2 3 5 7 11 13 17]; N=[100 500 1000]; JC=zeros(3,7);
for i=1:7
    p=P(i); for j=1:3; JC(j,i)=testrandom(4,p,8,N(j),0,5000); end
end
% Table 6.5
JC
```

The following program computes Table 6.6. It calls `GoE.m` and `GE.m`, which are stated above in the section.

```
N=[100 500 1000]; repetitions=5000;
E=zeros(1,3); G=zeros(1,3); ge=zeros(1,3);
for i=1:3
    n=N(i); E(i)=GoE(2,0,8,n,0,repetitions);
    G(i)=GoE(1,0,8,n,0,repetitions); ge(i)=GE(0,8,n,0,repetitions);
end
```

```
% Table 6.6
E
G
ge
```

A.4.2 Increasing the number of projections

The following program is used to compute Table 6.7. It calls `testrandom.m` that can be found above in the section.

```
K=[2^3 2^5 2^8]; Rk=zeros(3,3); N=[100 500 1000];
for i=1:3
    k=K(i); for j=1:3; Rk(j,i)=testrandom(k,5,8,N(j),0,5000); end
end
% Last three columns of Table 6.7
Rk
```

A.4.3 Real data

Canadian lynx and Wolfer sunspot data

Here we deal with Figure 6.4 and Table 6.8. Regarding Figure 6.4 we use the following program.

`figureReal2.m`

```
% This program computes a Figure with two plots. In the first one is
% plotted the Canadian lynx data and the second one with the Wolfer
% sunspot data
% Input data:  x = Canadian lynx data, it is a row vector
%              y = Wolfer sunspot data, it is a row vector
```

```
function figureReal2(x,y)
% first subplot
subplot(2,1,1); plot(1:length(x),x); axis([0 115 0 7500])
title('Canadian lynx data'); xlabel('time'); ylabel('process')
% second subplot
subplot(2,1,2); plot(1:length(y),y); axis([0 262 0 200])
title('Wolfer sunspot data'); xlabel('time'); ylabel('process')
```

In order to compute the first column of Table 6.8 we use the program `RealData.m` where for the first column we take the Canadian lynx data and for the second one the Wolfer sunspot data. The other two columns of Table 6.8 are taken from their respective papers. `RealData.m` calls `GestadisticoVn.m` and `Sub.m` that are above in the section.

`RealData.m`

```
% Input data:  x = is the process we want to test, it is given in a
% row vector
% Output data: Pvalue = p-value obtained by doing the random
% projection test for the process x
function Pvalue=RealData(x)
v=4; n=length(x); N=2; dN=2*N; rn=floor(n^.4);  T=zeros(1,v);
Cs=0; for i=1:v; Cs=Cs+1/i; end;
cc=zeros(1,v); for i=1:v; cc(i)=chi2inv(1-.05*i/(v*Cs),dN-2);end;
for i=1:v
   if mod(i,2)==1; A=100; B=1; elseif mod(i,2)==0; A=2; B=7;end
   ch=1; HH=betarnd(A,B,1,n); C=n;
   while ch>10^(-15) && C>1;
       a=ch*HH(C); ch=ch-a;
       if C==n; HH(C)=sqrt(a); else HH(C)=sqrt(a)/(n-C); end
       C=C-1;
   end
```

```
    HH(C)=sqrt(ch)/(n-C); h=HH(C:n); k=length(h); y=zeros(1,n);
    for j=1:(k-1); y(j)=x(1:j)*h(k-j+1:k)'; end
    for j=k:n; y(j)=x(j-k+1:j)*h'; end
    % Statistics
    dev=std(y)*(n-1)/n;
    if i≤v/2
        T(i)=GestadisticoVn(y,1);
    elseif i>v/2
        T(i)=Sub([abs(randn) 2*abs(randn)]/dev,y,dev,rn,n,dN,N);
    end
end
F=sort(T,'descend'); p=zeros(1,4);
for i=1:v; p(i)=(1-chi2cdf(F(i),2))/i; end; Pvalue=min(p)*25/3;
```

Sea waves data

The following program computes Figure 6.6 and Table 6.6. It calls `RealData.m` (above), `RealDataGE.m` (below) and `soukissian.m`. We have obtained `soukissian.m` by courtesy of J.B. Hernández.

```
% Output data: Pvalue = $p$-value obtained by doing the RP-test
%              PvalueGE = $p$-value obtained by doing the GE-test
function [Pvalue PvalueGE]=Olas
% With Datos we obtain a matrix D whose fourth column contains the
% 27,648 data measured between 10:00 and 16:00 the first of November
% 2009.
Datos
lD=length(D);
% Cm is the number of observations in 5 minutes
```

```
Cm=lD/72; dCm=Cm*2; n=ceil((lD-3*Cm)/dCm); Hs=zeros(n,1);
ini=1; fini=3*Cm;
for i=1:35
    Hs(i)=4*std(D(ini:fini,4)); ini=fini-Cm; fini=fini+dCm;
end
% Apply de Soukissian Algorithm
[S,R,Hs]=soukissian(Hs,n); Souki=[S.ini' ; S.fin']'
S.est
% Once selected the stationary segment, we compute the $p$values
sCm=1:7*Cm; O=D(sCm,4)'; Pvalue=RealData(O); PvalueGE=RealDataGE(O);
% Plot of the selected the stationary segment
plot(sCm,D(sCm,4))
```

RealDataGE.m

```
% Input data:  x = is the process we want to test, it is given in a
% row vector
% Output data: Pvalue = p-value obtained by doing the GE-test to the
% process x
function Pvalue=RealDataGE(x)
v=4; n=length(x); N=2; dN=2*N; rn=floor(n^.4);  T=zeros(1,v);
Cs=0; for i=1:v; Cs=Cs+1/i; end;
cc=zeros(1,v); for i=1:v; cc(i)=chi2inv(1-.05*i/(v*Cs),dN-2);end;
for i=1:v
   % Statistics
   dev=std(x)*(n-1)/n;
   if i≤v/2
       T(i)=GestadisticoVn(x,1);
   elseif i>v/2
       T(i)=Sub([abs(randn) 2*abs(randn)]/dev,x,dev,rn,n,dN,N);
```

```
   end
end
F=sort(T,'descend'); p=zeros(1,4);
for i=1:v; p(i)=(1-chi2cdf(F(i),2))/i; end
Pvalue=min(p)*25/3;
```

Bibliography

[1] Abraham, C., Biau, G. and Cadre, B. (2006). On the kernel rule for function classification. *Ann. Inst. Statist. Math.* **58**, 619–633. 78

[2] An, H.Z., Chen, Z.G. and Hannan, E.J. (1982). Autocorrelation, autoregression and autoregressive approximation. *Ann. Statist.* **10**(3), 926–936. 38

[3] Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis.* John Wiley & Sons. 36

[4] Azencott, R and Dacunha-Castelle, D. (1986). *Series of Irregular Observations: Forecasting and Model Building.* Springer. 11, 22

[5] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B* **57**(1), 289–300. 14, 39

[6] Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Ann. Statist.* **29**(4), 1165–1188. 14, 15, 39

[7] Biau, G., Bunea, F. and Wegcamp, M.H. (2005). Functional classification in Hilbert spaces. *IEEE Transact. Informat. Theo.* **51**, 2163–2172. 78

[8] Bowman, K.O. and Shenton, L.R. (1975). Omnibus test contours for departures from normality based on$\sqrt{b_1}$ and b_2. *Biometrika* **62**(2), 243–250. 37

[9] Breiman, L. (2001). Random forests. *Machine Learning* **45**(1), 5–32. 78

[10] Bugni, F.A., Hall, P., Horowitz, J.L. and Neumann G.R. (2009). Goodness-of-fit tests for functional data. *Econom. J.* **12**, S1–S18. 3, 19

[11] Cuesta-Albertos, J.A., Cuevas, A. and Fraiman, R. (2009). On projection-based tests for spherical and compositional data. *Stat. Comput.* **19**(4), 367–380. 3, 19

[12] Cuesta-Albertos, J.A., del Barrio, T., Fraiman, R. and Matrán, C. (2007). The random projection method in goodness of fit for functional data. *Computat. Statist. Data Anal.* **51**(10), 4814–4831. 3, 11, 12, 19, 23, 24, 33, 89

[13] Cuesta-Albertos, J.A. and Febrero-Bande, M. (2009). Multiway ANOVA for Functional Data. *Preprint.* 3, 19

[14] Cuesta-Albertos J.A., Fraiman R., Galves A., Garcia J. and Svarc, M. (2007). Classifying speech sonority functional data using a projected Kolmogorov-Smirnov approach. *J. Appl. Stat.* **34**(5-6), 749–761. 3, 19

[15] Cuesta-Albertos J.A., Fraiman R. and Ransford T. (2007). A sharp form of the Cramér-Wold theorem. *J. Theoret. Probab.* **20**, 201–209. 1, 2, 3, 11, 12, 17, 18, 19, 23, 32, 46, 47, 62, 126

[16] Cuesta-Albertos, J.A., Fraiman, R. and Ransford, T. (2006) Random projections and goodness-of-fit tests in infinite-dimensional spaces. *Bull. Braz. Math. Soc.*, **37**(4), 1–25. 3, 19

[17] Cuesta-Albertos J.A., Gamboa F. and Nieto-Reyes, A. (2009). A random-projection based procedure to test if a stationary process is Gaussian. *Submitted to Scand. J. Statist.* 15, 26

[18] Cuesta-Albertos J.A., Gamboa F. and Nieto-Reyes, A. (2009). Gaussianity in Stationary Processes: A Random Projection Approach. *Proceedings of the 16th European Young Statisticians Meeting.* Eds. Ciumara R. and Badin L., 5–9. 15, 26

[19] Cuesta-Albertos J.A. and Nieto-Reyes, A. (2008). The random Tukey depth. *Comput. Statist. Data Anal.* **52**(11), 4979–4988. 15, 26

[20] Cuesta-Albertos J.A. and Nieto-Reyes, A. (2008). The Tukey and the random Tukey depths characterize discrete distributions. *J. Multivariate Anal.* **99**(10), 2304–2311. 15, 26

[21] Cuesta-Albertos J.A. and Nieto-Reyes, A. (2008). A random functional depth. In *Functional and Operational Statistics.* Eds. S. Dabo-Niang and F. Ferraty. Springer, 121–126. 15, 26

[22] Cuesta-Albertos, J.A. and Matrán, C. (1991). On the asymptotic behavior of sums of pairwise independent random variables. *Statist. Probab. Lett.* **11**(3), 201–210. 116, 164

[23] Cuevas, A., Febrero-Bande, M. and Fraiman, R. (2007). Robust estimation and classification for functional data via projection-based depth notions. *Comput. Statist.* **22**(3), 481–496. 3, 5, 10, 19, 21, 26, 32, 80

[24] Cuevas, A. and Fraiman, R. (2009). On depth measures and dual statistics. A methodology for dealing with general data. *J. Multivariate Anal.* **100**(4), 753–766. 2, 3, 18, 19, 32, 46, 126

[25] Doob, J.L. (1953). *Stochastic Processes.* John Wiley & Sons. 94, 99, 100, 103

[26] Epps, T. W. (1987). Testing that a stationary time series is Gaussian. *Ann. Statist.* **15** (4), 1683–1698. x, 11, 13, 23, 35, 36, 37, 97, 98, 104, 105, 120, 121

[27] Febrero-Bande, M., Galeano, P. and González-Manteiga, W. (2008). Outlier detection in functional data by depth measures with application to identify abnormal NOx levels. *Environmetrics* **19**(4), 331–345. 3, 19

[28] Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis: Theory and Practice.* Springer. 76

[29] Ferraty, F. and Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. *Computat. Statist. Data Anal.* **44**, 161–173. 78

[30] Fox, J.T. and Gandhi A. (2009). Identifying heterogeneity in discrete choice, selection and other economic models. *Preprint.* 3, 19

[31] Fraiman, R. and Muniz, G. (2001). Trimmed means for functional data. *Test.* **10**(2), 419–440. 32

[32] Frankl, P. and Maehara, H. (1988). The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Combin. Theory Ser. B.* **44**(3), 355–362. 1, 17

[33] Gaposhkin, V. F. (1980). Almost sure convergence of estimates for the spectral density of a stationary process. *Theory Probab. Appl.* **25**, 169–176. 36

[34] Gasser, T. (1975). Goodness-of-fit tests for correlated data. *Biometrika* **62**(3), 563–570. 102

[35] Gebelein, H. (1941). Das statistische Problem der Korrelation als Variations- und Eigenwertproblem und sein Zusammenhang mit der Ausgleichsrechnung. *Z. Angew. Math. Mech.* **21**, 364–379. 97

[36] Gershenfeld, N (2000). *The Nature of Mathematical Modeling.* Cambridge: Cambridge Univ. Press. 10, 22

[37] Ghosh, A. K. and Chaudhuri, P. (2005). On maximum depth and related classifiers. *Scandinavian Journal of Statistics.* **32**, 327–350. 80

[38] Grenander, U. and Rosenblatt, M. (1984). *Statistical Analysis of Stationary Time Series.* Chelsea Publishing Company. 100

[39] Hand, D.J. (2006). Classifier technology and the illusion of progress. *Statist. Sci.* **21**(1), 1–14. 2, 18

[40] Hannan, E.J. (1970). *Multiple Time Series.* John Wiley & Sons. 10, 100, 103

[41] Hassairi, A. and Regaieg, O. (2007). On the Tukey depth of an atomic measure. *Stat. Methodol.* **4**(2), 244–249. 8, 22

[42] Hernández, J.B. and Ortega, J. (2007). A comparison of segmentation procedures and analysis of the evolution of spectral parameters. *Proc. ISOPE 2007.* 1836–1842. 122

[43] Hettmansperger, T.P. (1984). *Statistical Inference Based on Ranks.* John Wiley & Sons. 74

[44] Jarque, C.M. and Bera, A.K. (1987). A test for normality of observations and regression residuals. *Internat. Statist. Rev.* **55**, 163-172. 37

[45] Johnson, W.B. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. *Conference in Modern Analysis and Probability; Amer. Math. Soc.; Contemp. Math.* **26**, 189–206. 1, 17

[46] Kavalieris, L. (2008). Uniform convergence of autocovariances. *Statist. Probab. Lett.* **78**(6), 830–838. 14, 38

[47] Koshevoy, G. A. (2002). The Tukey depth characterizes the atomic measure. *J. Multivariate Anal.* **83**, 360–364 . 8, 22, 58

[48] Laha, R.G. and Rohatgi, V.K. (1979). *Probability Theory.* John Wiley & Sons. 95, 96

[49] Li, J., Cuesta-Albertos, J.A. and Liu, R.Y. (2009). Nonparametric classification procedures based on DD-plot. *Preprint* 10, 15, 26, 80

[50] Li, J. and Liu, R.Y. (2008). Multivariate spacings based on data depth: I. Construction of nonparametric multivariate tolerance regions. *Ann. Statist.* **36**(3), 1299–1323. 61

[51] Liu, R.Y. (1995). Control charts for multivariate processes. *J. Amer. Statist. Assoc.* **90**(432), 1380–1387. 61

[52] Liu, R.Y. (1990). On a notion of data depth based on random simplices. *Ann. Statist.* **18**, 405–414. 6, 29

[53] Liu, R.Y., Parelius, J.M. and Singh, K. (1999). Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Ann. Statist.* **27**(3), 783–858. 5, 21, 30

[54] Liu, R.Y., Serfling,R. and Souvaine, D.L., editors. (2006). Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications. *Amer. Math. Soc. DIMACS Series* **72**. 5, 20, 21

[55] Liu, R.Y. and Singh, K. (1993). A quality index based on data depth and multivariate rank tests. *J. Amer. Statist. Assoc.* **88**(421), 252–260. 61

[56] Liu, R.Y. and Singh, K. (1997). Notions of limiting p values based on data depth and bootstrap. *J. Amer. Statist. Assoc.* **92**(437), 266–277. 61

[57] Liu, R.Y. and Singh, K. (2006). Rank tests for nonparametric description of dispersion. *Amer. Math. Soc. DIMACS Series* **72**, 17–35. 9, 25, 67, 68, 69, 73, 75

[58] Lobato, I.N. and Velasco, C. (2004). A simple test of normality for time series. *Econometric Theory.* **20**(4), 671–689. 11, 13, 23, 24, 37, 101, 102, 104, 105, 107, 109, 111

[59] Loève, M. (1977). *Probability Theory I.* Springer. 95

[60] López-Pintado, S. and Romo, J. (2006). Depth-based classification for functional data. *Amer. Math. Soc. DIMACS Series* **72**, 103–119. 9, 26, 31, 77, 78, 79, 80, 82, 83, 84, 85

[61] López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *J. Amer. Statist. Assoc.* **104**(486), 718–734. 9, 31

[62] Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proc. Natl. Inst. Science* **12**, 49–55. 30

[63] Maronna, R.A., Martin, R.D. and Yohai, V.J. (2006). *Robust Statistics.* John Wiley & Sons. 65, 133

[64] Mielniczuk, J. (2000). Some properties of random stationary sequences with bivariate densities having diagonal expansions and nonparametric estimators based on them. *J. Nonparametr. Statist.* **12**(2), 223–243. 97

[65] Mosler, K. and Hoberg, R. (2006). Data analysis and classification with the zonoid depth. *Amer. Math. Soc. DIMACS Series* **72**, 49–59. 5, 9, 21, 66

[66] Moulines, E. and Choukri, K. (1996). Time-domain procedures for testing that a stationary time-series is Gaussian. *IEEE Trans. Sig. Proc.* **44**(8), 2010–2025. 11, 23

[67] Nelder, J.A. and Mead, R. (1965). A simplex method for function minimization. *Comput. J.* **7**, 308–313. 105

[68] Opazo, L., Raddatz, C. and Schmukler S.L. (2009). The long and the short of emerging market debt. *Preprint.* 3, 19

[69] Parelius, J. (1997). Multivariate Analysis Based on Data Depth. Ph.D. dissertation. Dept. Statistics, Rutgers Univ., New Jersey. 30

[70] Pitman, J. (2006). *Combinatorial Stochastic Processes. Lectures from the 32nd Summer School on Probability Theory held in Saint-Flour.* Springer. 91

[71] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007). *Numerical Recipes. The Art of Scientific Computing.* Cambridge University Press. 105, 175

[72] Ramsay, J.O. and Silverman, B.W. (1997). *Functional Data Analysis.* Springer. 77, 78

[73] Ramsay, J.O. and Silverman, B.W. (2002). *Applied Functional Data Analysis: Methods and Case Studies.* Springer. 77

[74] Ramsay, J.O., Hooker, G. and Graves, S. (2009). *Functional Data Analysis with R and MATLAB.* Springer. 77

[75] Rudin, W. (1966). *Real and Complex Analysis.* Mc Graw-Hill. 96

[76] Shapira, L., Avidan, S. and Shamir, A. (2009). Mode-detection via median-shift. *Preprint.* 15, 27

[77] Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging.* Springer. 11, 22

[78] Subba Rao, T. and Gabr, M.M. (1980). A test for linearity of stationary time series. *J. Time Ser. Anal.* **1**, 145–158. x, 11, 23, 120, 121

[79] Soukissian, T.H. and Samalekos, P.E. (2006). Analysis of the Duration and Intensity of Sea States Using Segmentation of Significant Wave Height Time Series. *Proc. ISOPE 2006.* **3**, 107–113. 122

[80] Tukey, J.W. (1975). Mathematics and picturing of data. *Proc. of ICM, Vancouver* **2**, 523–531. 4, 5, 20, 21, 30

[81] Vempala, S.S. (2004). The random projection method. *Amer. Math. Soc. DIMACS Series* **65**. 1, 2, 17, 18

[82] Witten I.H. and Frank E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann. 15, 26

[83] Zuo, Y. and Serfling, R. (2000). General notions of statistical depth function. *Ann. Statist.* **28**(2), 461–482. 5, 6, 21, 24, 29, 42, 44, 45, 125

[84] Zuo, Y. (2003). Projection-based depth functions and associated medians. *Ann. Statist.* **31**(5), 1460–1490. 5, 21

[85] Zuo, Y. (2006). Multidimensional trimming based on projection depth. *Ann. Statist.* **34**(5), 2211–2251. 5, 6, 21