Introducción a la detección de anomalías por métodos de aprendizaje máquina

Hugo Salvador Gonzalo

Trabajo de fin de grado de Matemáticas
Universidad de Zaragoza

Director del trabajo: José Tomás Alcalá Nalvaiz
12 de septiembre de 2022

Abstract

Outlier detection is a key step in any data analysis process. It is a data cleaning technique which is used in multitude of fields, such as medicine, cyber security, statistics or financial fraud, among others. The objective of this project is to review the main mathematical properties of two different algorithms, as well as describe their best statistical aspects that allow us to detect outliers with an effective performance. Finally, we consider two real data sets and analyze how to apply both models to them using the latest R software libraries related to this topic. We ended up comparing the performances of both of these methods.

The content of this project is divided in 3 chapters. In chapter 1, we introduce a definition of the term Machine Learning, as well as the presentation of the different sorts of algorithms that make it up, which are supervised, unsupervised and reinforcement. Subsequently, outlier detection is defined as the process of pattern recognition in data that do not belong to a expected behaviour. Furthermore, the main families of the outlier detection algorithms are presented.

In chapter 2, the different reasons justifying the decision of choosing these algorithms are shown, as well as the algorithms themselves. First, the introduction of the Principal Component Analysis (PCA) model defined as the process of computing the principal components and their posterior use to change the basis of data. Followed by its functioning basis as a dimension reduction algorithm and also its outlier detection application. Then the Isolation Forest model is introduced, together with its functioning based on decision trees, its similarity with the classification model Random Forest, and its statistical basis that allows the model to achieve a great performance as outlier detection algorithm.

In chapter 3, two real data sets are introduced, followed by a summary of the strategies designed to be used in the practical application of the proposed methods. The objective of this implementation is to construct both models in an optimized way, allowing us to determinate in a justified manner which is the best model in each scenario. For that, we propose multiple applications of each technique with different parameters and only using the best one of them. As we achieve this, we use the area under the *ROC* curve, as well as the achieved specificity and sensitivity as statistical indicators to evaluate the degree of efficiency of both models.

In addition, the R code proposed to perform the case study can be consulted in the appendix.

Índice general

Capítulo 1

Aprendizaje máquina

1.1. Introducción

El aprendizaje máquina o Machine Learning [1] es una disciplina científica, subcampo de las ciencias de la computación y una rama de la Inteligencia Artificial cuyo objetivo es el desarrollo de técnicas que permitan aprender a determinados sistemas (informáticos). Se dice que un agente aprende cuando su desempeño mejora con la experiencia y mediante el uso de datos, es decir, dicha habilidad no forma parte de su genotipo o rasgos de nacimiento. El sistema (informático) que aprende es un algoritmo que revisa los datos y, realizando un modelo a partir de esos datos, es capaz de predecir comportamientos futuros.

El aprendizaje automático puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos. Por lo tanto se trata de un proceso de inducción del conocimiento.

El desarrollo del Machine Learning, como campo separado de la Inteligencia Artificial, se inicia a finales de los setenta y principios de los ochenta con la creación de múltiples algoritmos. Estos algoritmos pueden agruparse en función de su salida u output [1]:

- **Aprendizaje supervisado.** El algoritmo produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Dicho de otra forma, es una técnica para deducir una función a partir de datos de entrenamiento, los cuales consisten en pares de objetos, normalmente vectores, donde una componente del par son los datos de entrada y el otro, los resultados deseados. (La salida de la función puede ser un valor numérico, como en regresión, o una etiqueta de clase, como en clasificación).

  El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir, para cualquier objeto de entrada válida, el valor correspondiente después de analizar los datos de entrenamiento. Para conseguirlo, es necesario generalizar a partir de los datos dados a situaciones no presentadas anteriormente. Este tipo de aprendizaje automático puede utilizarse para descubrir la estructura subyacente de los datos.

- **Aprendizaje no supervisado.** Se trata de los métodos en los que un modelo se ajusta a las observaciones y trata de encontrar una estructura en los datos. Al contrario que en el aprendizaje supervisado, no hay un conocimiento a priori respecto a la muestra. Debido a esto trata las observaciones como un conjunto de variables aleatorias y se construye un modelo de densidad.

  Los algoritmos de aprendizaje supervisado reconocen patrones de datos sin referencia a resultados conocidos. Además permiten realizar tareas de procesamiento de datos más complejas en comparación con el aprendizaje supervisado. Sin embargo pueden ser más imprevisibles en comparación con otros métodos.

- **Aprendizaje por refuerzo.** En estos casos el algoritmo aprende observando el mundo que le rodea en base a recompensas y penalizaciones, derivadas del éxito o del fracaso respectivamente.

Su información de entrada es el feedback o la retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Su objetivo principal es aprender la función de valor que le ayude al agente a maximizar la señal de recompensa, optimizando así sus políticas de forma que comprenda el comportamiento del entorno y tome buenas decisiones para el logro de sus objetivos formales.

1.2. Detección de anomalías

En un problema de clasificación convencional, el objetivo es encontrar un clasificador que separe de forma óptima dos (o más) clases. Este tipo de problemas se conocen comúnmente como problemas de clasificación multiclase. En ellos, el término de entrada del problema es un conjunto de entrenamiento etiquetado formado por un número similar de casos de cada clase. Sin embargo, hay tipos de problemas en los que este supuesto de (aproximar) distribución uniforme de ejemplos no se sostiene. El ejemplo prototípico de estos problemas es la detección de anomalías.

Definición. *La detección de anomalías es el proceso de identificación de patrones en los datos que no se ajustan a un comportamiento previsible.*

Según esta definición tomada de [3], una anomalía es un evento raro normalmente encontrado con escasez en los datos de entrenamiento. Además las anomalías en su mayoría pueden ser detectadas cuando son observadas teniendo en cuenta el contexto, es decir, comparándolas con la mayoría de los puntos regulares. Por lo tanto, la detección de anomalías proporciona un ejemplo de la denominada *clasificación de una clase* o *one-class classification* [4]. El quid de esta consiste en, dados los datos originados por una sola clase, los cuales están posiblemente contaminados por una pequeña cantidad de atípicos, encontrar la frontera de dicha clase.

Durante la década de los 80, Dorothy E. Denning [5] trabajó en la detección de intrusos en tiempo real. Este trabajo sirvió como base para futuros métodos y aplicaciones de anomalías. El desarrollo de estas nuevas técnicas se recopila en libros como *Outlier analysis* [7] de Charu C. Aggarwal, creando un cuerpo científico de este nuevo campo y consolidándolo como una ciencia más.

La detección de anomalías tiene una relevancia significativa y suele proveer de información crítica factible en numerosos dominios de aplicación. Un punto anómalo en una imagen del campo de la astronomía podría indicar el descubrimiento de una nueva estrella; un patrón en el tráfico en la red podría significar un acceso no autorizado. Dichas aplicaciones necesitan de algoritmos de detección de anomalías con un gran rendimiento en la detección y una rápida ejecución.

Los principales algoritmos de detección de anomalías se agrupan en familias, entre las que destacan las siguientes (recopiladas del capítulo 1 de [7]): modelos probabilísticos y estadísticos, modelan los datos en forma de una distribución de probabilidad; modelos lineales, modelan los datos a lo largo de subespacios de menor dimensión mediante el uso de relaciones lineales; modelos basados en la proximidad, modelan las anomalías aislandolas del resto de los datos en base a su similaridad o a métricas; modelos de teoría de la información, representan las anomalías como ampliaciones del código mínimo necesario para representar el resto de los datos; y modelos de alta dimensión.

Capítulo 2

Algoritmos

En este trabajo se abordan dos algoritmos pertenecientes a las familias de modelos lineales y de alta dimensión, respectivamente. Esta elección se debe a la similaridad de su funcionamiento con varios conceptos estudiados a lo largo del grado en las asignaturas de Estadística Matemática y Técnicas de Regresión en el caso del Análisis de componentes principales o la teoría de Grafos en la cual se basan los árboles de decisión y, en particular, los Isolation Forests.

2.1. Análisis de componentes principales

2.1.1. Introducción

El Análisis de Componentes Principales (ACP) es un método estadístico, perteneciente a la familia del aprendizaje no supervisado, que permite simplificar la complejidad de espacios muestrales con muchas dimensiones a la vez que conserva su información. Su objetivo es analizar si es posible representar razonablemente esta información con un número menor de variables, las cuales son combinaciones lineales de la originales.

Suponer que existe una muestra de n individuos cada uno con p variables, es decir, el espacio muestral tiene p dimensiones. El ACP permite encontrar un número de combinaciones lineales ($l < p$) que otorgan aproximadamente la misma cantidad de información que las p variables originales. Donde antes se necesitaban p valores para caracterizar a cada individuo ahora bastan l valores, es decir, permite "condensar" la información aportada por múltiples variables en solo unas pocas componentes.

La técnica de componentes principales encuentra sus orígenes en los ajustes ortogonales por mínimos cuadrados. Su utilidad permite representar de forma óptima en un espacio de dimensión pequeña observaciones de un espacio general p-dimensional. En este sentido nos otorga un primer paso en la identificación de posibles variables no observadas que generan variabilidad en los datos. Además, permite transformar las variables originales correladas en variables incorreladas, facilitando la interpretación de los datos.

2.1.2. Planteamiento

Suponer que se dispone de los valores de p variables en n elementos de una población representados en una matriz $n \times p$ denominada X, con media muestral cero, cuyas filas y columnas representan los elementos y las variables, respectivamente.

Para resolver el problema sobre cómo encontrar un espacio de dimensión más reducida que represente adecuadamente los datos puede ser abordado desde distintas perspectivas:

Enfoque descriptivo.

Se desea encontrar un subespacio de dimensión menor que p tal que, al proyectar sobre él los puntos, conserven su estructura con la menor dispersión posible. Considerando un punto $x_i \in \mathbb{R}^p$ y una dirección

$a_1 = (a_{11}, \ldots, a_{1p})^T$, definida por un vector a_1 de norma unidad, la proyección del punto x_i sobre esta dirección es

$$z_i = a_{11}x_{i1} + \cdots + a_{1p}x_{ip} = a_1^T x_i, i \in \{1, \ldots, n\}$$

y el vector que representa dicha proyección es $z_i a_1$. Llamando r_i a la distancia entre el punto x_i y su proyección sobre a_1, la resolución del proceso implica

$$minimizar \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} |x_i - z_i a_1|^2 \tag{2.1}$$

con $|u|$ norma euclídea del vector u.

Aplicando el Teorema de Pitágoras para cada punto se tiene

$$\sum_{i=1}^{n} x_i^T x_i = \sum_{i=1}^{n} r_i^2 + \sum_{i=1}^{n} z_i^2.$$

Dado que el primer término es constante, minimizar $\sum_{i=1}^{n} r_i^2$, es equivalente a maximizar $\sum_{i=1}^{n} z_i^2$. Y, por ser las variables z_i de media nula, maximizar la suma de sus cuadrados equivale a maximizar su varianza.

Este problema está relacionado con el siguiente, llamando $d_{ij}^2 = x_i^T x_j$ a los cuadrados de las distancias originales entre los puntos, y $\widehat{d}_{ij}^2 = (z_i - z_j)^2$ a las distancias entre los puntos proyectados sobre una recta, se pretende minimizar

$$D = \sum_i \sum_j (d_{ij}^2 - \widehat{d}_{ij}^2).$$

Y, dado que las distancias originales son fijas, minimizar D es sinónimo de maximizar las distancias entre los puntos proyectados, $\sum_i \sum_j \widehat{d}_{ij}^2$.

Lema 2.1. *Maximizar las distancias al cuadrado entre los puntos proyectados es equivalente a maximizar la varianza de la variable dada por las proyecciones de los puntos.*

Demostración. Sea z_i la proyección de la observación x_i sobre la dirección a_1 previamente definida. z_i tendrá media cero por ser las x de media cero. La suma de las distancias al cuadrado entre los puntos proyectados es

$$D_p = \sum_{i=1}^{n} \sum_{h=i+1}^{n} \widehat{d}_{ij}^2.$$

Se puede observar que cada término z_i aparece al cuadrado $n-1$ veces (ya que cada punto se compara con los otros $n-1$, y que habrá tantos dobles productos como parejas de puntos, es decir, $\binom{n}{2} = n(n-1)/2$). Por tanto

$$D_p = (n-1)\sum_{i=1}^{n} z_i^2 - 2\sum_{i=1}^{n}\sum_{h=i+1}^{n} z_i z_h = n\sum_{i=1}^{n} z_i^2 - \left(\sum_{i=1}^{n} z_i^2 + 2\sum_{i=1}^{n}\sum_{h=i+1}^{n} z_i z_h\right)$$

y desarrollando se obtiene

$$\sum_{i=1}^{n} z_i^2 + 2\sum_{i=1}^{n}\sum_{h=i+1}^{n} z_i z_h = z_1(z_1 + \cdots + z_n) + z_2(z_1 + \cdots + z_n) + \cdots + z_n(z_1 + \cdots + z_n) = \sum_{i=1}^{n} z_i \sum_{i=1}^{n} z_i = 0$$

Por tanto, maximizar las distancias entre los puntos equivale a maximizar $n\sum_{i=1}^{n} z_i^2$ que es el criterio de maximizar la varianza de la nueva variable (2.1), obtenido anteriormente. □

Enfoque geométrico.

Considerando la nube de puntos de la figura 2.1, se puede observar que los puntos se sitúan creando una elipse, pudiendo describir su orientación dando la dirección del eje mayor de dicha elipse y la posición de cada punto por su proyección sobre esta dirección. Puede demostrarse que este eje es la recta que minimiza distancias ortogonales, volviendo al problema ya resuelto. En un subespacio de mayor

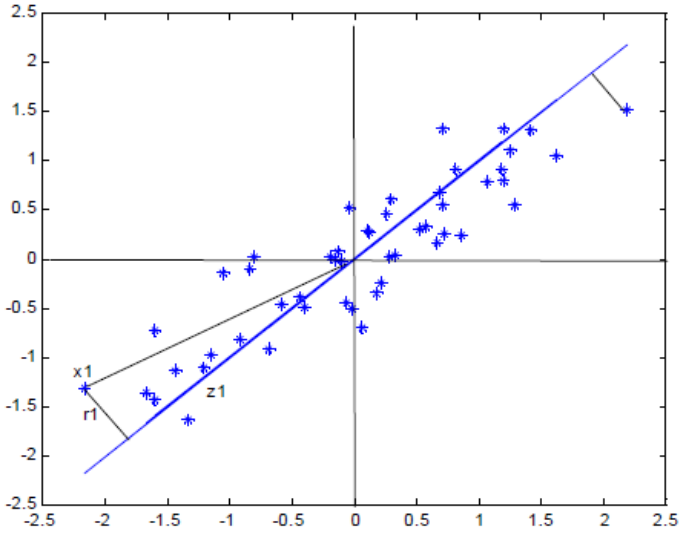


Figura 2.1: Figura 5.1 correspondiente a [8]

dimensión, se tratará con elipsoides y la mejor aproximación será la proporcionada por el eje mayor de los elipsoides.

Además, es importante presentar el concepto de *distancia de Mahalanobis*, ya que se trata de una métrica muy útil cuando es necesario trabajar con datos estandarizados. Para ello, hay que conocer la matriz de covarianzas de la muestra S_x, $p \times p$, donde su elemento (i, j) es igual a la covarianza entre las dimensiones i y j.

Definición. *Se define la distancia de Mahalanobis entre un punto, x_i, y su vector de medias, $\bar{x}$, por*

$$d_M^2(x_i, \bar{x}, S_x) = [(x_i - \bar{x})^T S_x^{-1} (x_i - \bar{x})]. \tag{2.2}$$

Nota 1. El cálculo de la distancia de Mahalanobis requiere la inversión de la matriz de covarianzas S_x.

2.1.3. Obtención de las componentes

Considerar la matriz de covarianzas muestral S_x, la cual se representa como

$$S_x = \frac{X^T X}{n}.$$

Se define su raíz cuadrada $S_x^{1/2}$, por la condición

$$S_x = S_x^{1/2} (S_x^{1/2})^T.$$

La matriz $S_x^{1/2}$ es simétrica y definida positiva y se puede diagonalizar de la siguiente forma

$$S_x = PDP^T$$

donde D es una matriz diagonal que contiene los valores propios de S_x y P es una matriz ortogonal que contiene los vectores propios. Sea $D^{1/2}$ la matriz diagonal cuyos términos corresponden a las raíces cuadradas de los elementos de D, siendo estos positivos. Definiendo la raíz cuadrada obtenemos

$$S_x^{1/2} = PD^{1/2}P^T.$$

La estandarización denominada *multivariante* se define como

$$Y = XS_x^{-1/2} = XPD^{-1/2}P^T.$$

Las propiedades más importantes respectivas a esta estandarización son

1. Centralización de la matriz
2. Su matriz de varianzas y covarianzas es la matriz identidad.
3. La distancia de Mahalanobis de los datos coincide con la distancia euclídea de los datos estandarizados.

Demostración. (*Propiedad* 2.) Sea

$$y_i^T y_i = (x_i - \bar{x})^T S_x^{-1/2} S_x^{-1/2} (x_i - \bar{x}) = d_M^2(x_i, \bar{x}, S_x),$$

con y variable de media cero y matriz de varianzas y covarianzas identidad, ya que

$$S_y = S_x^{-1/2} S_x S_x^{-1/2} = I.$$

Con esta transformación pasamos de variables correladas con matriz de varianzas y covarianzas muestral S_x, a variables incorreladas con matriz de varianzas y convarianzas muestral identidad. □

Esta estandarización se denomina *multivariante*, ya que utiliza todas las covarianzas para estandarizar cada variable.

La transformación aplicando componentes principales conduce a variables incorreladas con distinta varianza, puede interpretarse como rotar los ejes de la elipse que trazan los puntos para que coincidan con sus ejes naturales (el subespacio, que es una combinación lineal de los $(p-1)$ mayores autovectores, otorga un sistema de ejes en el cual los datos pueden representarse aproximadamente con poca pérdida).

Otra forma de expresar esto es que, considerando la población p-dimensional, un hiperplano normal relativo al menor autovector de S provee un hiperplano $(p-1)$-dimensional que aproxima los datos con el menor error de mínimos cuadrados. Es posible generalizar este argumento con los l mayores autovectores para definir un subespacio l-dimensional correspondiente de representación (aproximada). Aquí, l puede ser cualquier valor en $\{1,\ldots,p-1\}$. Observar que un dato atípico consistirá en un punto de los datos para el cual el error de esta aproximación será alto.

Por tanto en el ACP el primer paso es el de transformar los datos a un sistema de representación de nuevos ejes. Los autovectores ortonormales otorgan los ejes de las direcciones a lo largo de las cuales los datos deberían proyectarse. Las propiedades clave del ACP que serán relevantes para la detección de anomalías son las siguientes:

Propiedad 1. El ACP otorga un conjunto de autovalores que satisfacen las siguientes propiedades

- Los datos se transforman respecto al sistema de ejes correspondiente a los autovectores ortogonales, la varianza de los datos transformados a lo largo de cada eje (autovector) es igual a la del autovalor correspondiente. Las covarianzas de los datos transformados en la nueva representación son 0.
- Las varianzas de los datos transformados a lo largo de los autovectores asociados a autovalores pequeños son bajas, las desviaciones resultantes de los datos transformados de los valores medios a lo largo de estas direcciones podrían representar atípicos.

Por tanto, si el j-ésimo autovalor es muy pequeño, el valor de x'_{ij} en esta nueva representación transformada no varía demasiado respecto a los diferentes valores de i (con j fija). Los puntos atípicos en esta componente principal serán fácilmente detectables ya que se alejan de su valor medio 0.

Estas direcciones también son conocidas como *direcciones principales* y las nuevas variables definidas por estas se denominan *componentes principales*, las cuales son incorreladas y retienen la mayoría de la varianza de los datos. En un escenario real es común para una gran parte de los autovalores el ser muy pequeños, lo que significa que la mayoría de los datos se alinean a lo largo de un *subespacio de baja dimensión*. (Esto es muy conveniente desde la perspectiva de la detección de atípicos, ya que las observaciones que no respetan este ajuste pueden considerarse como atípicos).

Lema 2.2 (Cálculo de la primera componente)**.** *La primera componente principal es la combinación lineal de las variables originales que tiene varianza máxima.*

Demostración. Consideremos que los valores de esta primera componente en los n individuos se representarán por un vector z_1, dado por

$$z_1 = Xa_1.$$

Por ser z_1 de media nula, su varianza será:

$$S_{z_1}^2 = \frac{1}{n} z_1^T z_1 = \frac{1}{n} a_1^T X^T X a_1 = a_1^T S_x a_1$$

donde S_x es la matriz de varianzas y covarianzas de las observaciones. Es obvio que podemos maximizar la varianza aumentando el módulo del vector a_1. Para que la maximización de la varianza tenga solución habrá imponer una restricción al módulo del vector a_1, la cual será $a_1^T a_1 = 1$. Se introducirá esta restricción mediante el multiplicador de Lagrange:

$$M = a_1^T S_x a_1 - \lambda (a_1^T a_1 - 1)$$

y se maximizará esta expresión derivando respecto a las componentes de a_1 e igualando a cero. Por tanto

$$\frac{\partial M}{\partial a_1} = 2S_x a_1 - 2\lambda a_1 = 0$$

cuya solución es:

$$S_x a_1 = \lambda a_1,$$

luego λ es un valor propio de la matriz S_x y a_1 es vector propio respecto del valor propio λ. Multiplicando esta ecuación por a_1^T a izquierda,

$$a_1^T S_x a_1 = \lambda a_1^T a_1 = \lambda$$

obtenemos que el valor propio λ es igual a la varianza de z_1. Dado que esta es la cantidad a maximizar, λ es el mayor valor propio de S_x y su vector asociado a_1 contiene los coeficientes asociados a cada variable en la primera componente principal. □

Una vez obtenida la primera componente, se busca obtener la segunda, de forma que los vectores a_1 y a_2 que definirán el plano sean ortogonales, es decir, $a_1 \perp a_2$. Además, se establecerá como función objetivo la máxima suma de las varianzas z_1 y z_2 para calcular el mejor plano de proyección de las variables X.

Corolario 2.3 (Generalización)**.** *Análogamente al proceso realizado previamente puede demostrarse que el espacio de dimensión r que mejor representa a los puntos viene dado por los vectores propios asociados a los r mayores autovalores de S_x. Llamando Z a la matriz cuyas columnas son los valores de las p componentes en los n individuos, se puede generalizar la relación de estas variables con las originales mediante*

$$Z = XP$$

donde $P^T P = I$. En resumen, calcular las componentes principales equivale a aplicar una transformación ortogonal P a las variables originales X para obtener nuevas variables Z, incorreladas entre sí.

Este proceso puede interpretarse como la elección de unos nuevos ejes coordenados que coincidan con los ejes originales de los datos.

Lema 2.4. *Las componentes principales son predictores óptimos de las* **X**.

Demostración. Se comenzará demostrando que si se quiere aproximar la matriz $\mathbf{X}$ de rango p por otra matriz $\mathbf{X}_r$ de rango $r < p$, la aproximación óptima es $\mathbf{X}\mathbf{P}_r\mathbf{P}_r^T = \mathbf{Z}_r\mathbf{P}_r^T$, donde la matriz $\mathbf{P}_r$ es $p \times r$ y sus columnas son los vectores propios asociados a los r mayores valores propios de la matriz $\mathbf{S}_x$.

Para aproximar la matriz $\mathbf{X}$ se buscan unas variables $[\mathbf{z}_1, \ldots, \mathbf{z}_r]$ que sean combinaciones lineales de las variables originales y que tengan la propiedad de preverlas de manera óptima. Se particulariza para el caso en que $r = 1$ (replicable para los casos en que $r > 1$). Se busca un vector $\mathbf{a}_1$ de forma que la nueva variable estará definida como

$$\mathbf{z}_1 = \mathbf{X}\mathbf{a}_1$$

lo cual permitirá prever con mínimo error los valores observados para el conjunto de variables que forman las columnas de la matriz $\mathbf{X}$. Así, el valor previsto para la variable x_j en el individuo i, $\hat{x}_{ij}$, conocido el valor de la variable z_1 para ese individuo, z_{1i}, será

$$\hat{x}_{ij} = b_j z_{1i}$$

de forma que el error de predicción será $e_{ij} = x_{ij} - \hat{x}_{ij}$.

En primer lugar se calcula el vector $\mathbf{a}_1$ para que minimice estos errores de predicción. Es conocido que el coeficiente de regresión b_j viene dado por

$$b_j = \frac{\sum_{i=1}^n x_{ij} z_{1i}}{\sum_{i=1}^n z_{1i}^2}$$

y como $1/n \sum z_{1i}^2 = 1/n\, \mathbf{a}_1^T \mathbf{X}^T \mathbf{X} \mathbf{a}_1 = \mathbf{a}_1^T \mathbf{S}_\mathbf{x} \mathbf{a}_1$, la varianza de z_1 puede crecer indefinidamente si no se impone restricción alguna. Para evitar esto se impondrá la condición de que sea unitaria, es decir,

$$\mathbf{a}_1^T \mathbf{S}_\mathbf{x} \mathbf{a}_1 = 1 = 1/n \sum z_{1i}^2. \tag{2.3}$$

Entonces

$$b_j = 1/n \sum x_{ij} z_{1i} = 1/n \mathbf{x}_j^T \mathbf{X} \mathbf{a}_1 = \mathbf{v}_j^T \mathbf{a}_1 \tag{2.4}$$

donde $\mathbf{v}_j$ es el vector fila j de la matriz $\mathbf{S}$ de varianzas y covarianzas muestral. Impongamos la condición mínimo cuadrática para obtener $\mathbf{a}_1$:

$$\frac{1}{n} \sum_{i=1}^n e_{ij}^2 = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \mathbf{v}_j^T \mathbf{a}_1 z_{1i})^2.$$

Desarrollando se obtiene

$$\frac{1}{n} \sum_{i=1}^n (x_{ij} - \mathbf{v}_j^T \mathbf{a}_1 z_{1i})^2 = \frac{1}{n} \sum_{i=1}^n x_{ij}^2 + \frac{1}{n} \mathbf{a}_1^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{a}_1 \sum_{i=1}^n z_{1i}^2 - 2\mathbf{v}_j^T \mathbf{a}_1 \frac{1}{n} \sum_{i=1}^n x_{ij} z_{1i}$$

y utilizando (2.3) y (2.4), se obtiene

$$\frac{1}{n} \sum_{i=1}^n e_{ij}^2 = \frac{1}{n} \sum_{i=1}^n x_{ij}^2 - \mathbf{a}_1^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{a}_1.$$

Aplicando el mismo razonamiento al resto de variables x y sumando todas ellas se obtiene la expresión del error cuadrático total de la aproximación

$$M = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p e_{ij}^2 = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p x_{ij}^2 - \sum_{j=1}^p \mathbf{a}_1^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{a}_1$$

como el primer término es la traza de $\mathbf{S}_x$ que es fija, minimizar M equivale a maximizar

$$\mathbf{a}_1^T \sum_{j=1}^p \mathbf{v}_j \mathbf{v}_j^T \mathbf{a}_1 = \mathbf{a}_1^T \mathbf{S}_x \mathbf{S}_x^T \mathbf{a}_1 = \mathbf{a}_1^T \mathbf{S}_x^2 \mathbf{a}_1$$

por ser $\mathbf{S}_x$ simétrica. Por lo tanto, debido a la restricción 2.3:

$$L = \mathbf{a}_1^T \mathbf{S}_x^2 \mathbf{a}_1 - \lambda(\mathbf{a}_1 \mathbf{S}_x \mathbf{a}_1 - 1)$$

y derivando se tiene

$$\frac{\partial L}{\partial \mathbf{a}_1} = 2\mathbf{S}_x^2 \mathbf{a}_1 - \lambda 2\mathbf{S}_\mathbf{x}\mathbf{a}_1 = 0$$

$$\mathbf{S}_x^2 \mathbf{a}_1 = \lambda \mathbf{S}_\mathbf{x}\mathbf{a}_1$$

de donde se concluye que $\mathbf{a}_1$ debe ser un vector propio asociado al valor propio λ de la matriz $\mathbf{S}_x$. □

2.1.4. Efectos y detección de atípicos

Las consecuencias de una única observación atípica pueden ser graves, entre ellas destacan la distorsión de las medias y de las desviaciones típicas de las variables y la destrucción de las relaciones existentes entre ellas. Sin embargo, el efecto del atípico depende de su tamaño, medido por su distancia euclídea al centro de los datos, y de su posición, ya que los datos más afectados de la matriz *S* dependen de la posición de dicha observación en el espacio.

Cuando existe más de un atípico, puede producirse un efecto conocido como *enmascaramiento*, el cual consiste en que las observaciones atípicas se ocultan entre sí. De esta forma, aunque se elimine uno de ellos, los demás continuarán distorsionando el cálculo de las medias y varianzas, haciendo muy difícil su identificación.

El procedimiento para detectar grupos de atípicos consiste en eliminar de la muestra todo elemento sospechoso, evitando así el enmascaramiento y calculando el vector de medias y la matriz de covarianzas sin alteraciones. El primer paso para identificar las observaciones sospechosas es detectar aquellas que sean claramente atípicas respecto a una variable. Para ello, una regla simple y generalizada es considerar sospechosas aquellas observaciones que satisfagan

$$\frac{|x_i - med(x)|}{Meda(x)} > 4,5$$

donde *med(x)* es la mediana de las observaciones, que es un estimador robusto del centro de los datos, y *Meda(x)* es la mediana de las desviaciones absolutas $|x_i - med(x)|$, que es una medida robusta de la dispersión. Por tanto, este método puede tratarse como una estandarización robusta de los datos.

Sin embargo, esta detección no será capaz de identificar muchos atípicos multivariantes ya que con frecuencia dichos atípicos corresponden a situaciones con efectos pequeños sobre todas las variables en lugar de un efecto importante en una de ellas, como un error sistemático de observación general.

En la ilustración 2.2 se ilustra la efectividad del ACP al destapar atípicos en forma de diagrama de dispersión tridimensional. En este caso, los autovectores han sido ordenados respecto a sus autovalores (varianzas), de forma que la mayor parte de la varianza estaría contenida en el subespacio de menor dimensión formado por los dos autovectores correspondientes a los dos mayores autovalores, aunque una cantidad significativa de varianza sería capturada eligiendo únicamente el primer autovector. Si las distancias de los datos originales a la linea unidimensional correspondiente al primer autovector (y pasando a través de la media de los datos) son calculadas, el punto *x* de los datos en la figura 2.2 sería inmediatamente catalogado como atípico. En el caso de datos de alta dimensión, la mayoría de la varianza de los datos puede estar contenida en un menor subespacio *k*-dimensional.

Los residuos correspondientes a los datos pueden ser calculados examinando las distancias de proyección a este hiperplano *k*-dimensional que atraviesa la media de los puntos de los datos. Los puntos más distantes al hiperplano pueden identificarse como atípicos. Aunque se puede utilizar esta distancia como medición de atípicos, también es posible mejorar la puntuación mediante la normalización de la siguiente forma.

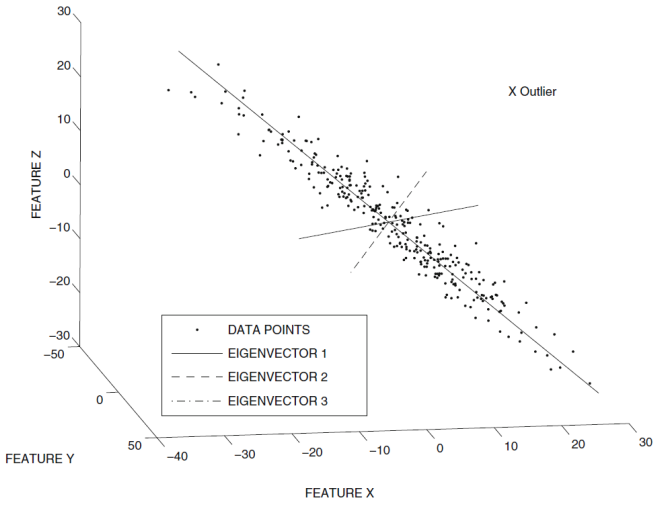


Figura 2.2: Figure 3.4 correspondiente a [7]

La distancia euclídea a este hiperplano puede descomponerse en la suma de cuadrados de las $(d-k)$ distancias a lo largo de los menores autovectores. Cada una de estas $(d-k)$ distancias cuadradas debería dividirse por los autovalores correspondientes (para la estandarización de la varianza) y los valores escalados deberían ser sumados para obtener el resultado final.

Nota 2. La intuición detrás de este escalado es que la varianza a lo largo de un autovector es su autovalor y, por lo tanto, una gran desviación a lo largo de un menor autovalor debería ser compensada en mayor medida.

2.1.5. Conexiones con el método de Mahalanobis

En el método anterior, se eliminan los k mayores autovectores de una forma *dura* y se calcula una suma con pesos de estas distancias al cuadrado a lo largo de las restantes $(p-k)$ direcciones para su posterior uso como puntuación de anomalías. Un caso más simple sería el uso de un método *suave* para la clasificación (ponderación) de las distancias a lo largo de todos los distintos autovectores en vez de seleccionar un conjunto particular de autovectores de forma dura.

Este cálculo se consigue evaluando la distancia normalizada de los puntos al centroide a través de la dirección de cada componente principal. Sea a_j el j-ésimo autovector con una varianza (autovalor) de λ_j a lo largo de dicha dirección. La puntuación de anomalía normalizada total de un punto x al centro $\bar{x}$ viene dado por la suma de cuadrados de estos valores

$$Score(x) = \sum_{j=1}^{p} \frac{|(x-\bar{x})^T a_j|^2}{\lambda_j} \tag{2.5}$$

Notar la presencia de λ_j en el denominador, lo cual otorga una ponderación suave. Se puede dar un algoritmo simple para calcular esta puntuación de las filas de la matriz X, $n \times p$, de datos:

Es importante notar que la mayoría de la contribución a la puntuación de anomalías de la ecuación (2.5) vendrá dada por desviaciones a través de las componentes principales con pequeños valores de λ_j, cuando un punto se desvía significativamente en alguna de estas direcciones. Este paso está contenido en el siguiente algoritmo de estandarización [1]. Este paso reconoce que las componentes principales son incorreladas e implementan una forma de ponderación suave de las dimensiones transformadas mediante estandarización en vez de seleccionar un subconjunto de dimensiones transformadas de una forma dura. En poblaciones normales, la suma de cuadrados de las distancias euclídeas del punto desde el centro a

Algorithm 1 Índice de anomalías por ACP

1: Calcular la matriz de covarianzas S_x de la matriz de datos originales y se diagonaliza como $S_x = PDP^T$.
2: Transformar los datos X a un nuevo sistema decorrelado de ejes a $X' = XP$.
3: Estandarizar cada columna de X' a la varianza unidad dividiendo por su desviación estándar.
4: Para cada fila de X', obtener su distancia euclídea desde el centroide de X' y utilizarla como su puntuación de anomalías.

lo largo de las dimensiones transformadas tiene una distribución χ^2 con p grados de libertad. El valor del residuo agregado se compara a la distribución acumulada para la distribución χ^2 para determinar un valor de probabilidad respectivo al nivel de incongruencia.

La distancia de Mahalanobis entre x_i y $\bar{x}$ calculada en (2.2) es exactamente la misma que la puntuación descrita en (2.5) exceptuando un mejor entendimiento del análisis de los autovectores en cómo esta puntuación se descompone a través de las distintas direcciones de correlación.

2.1.6. Sensibilidad al ruido

El ACP es generalmente más estable a la presencia de algunos atípicos que los métodos analíticos de variables dependientes. Esto se debe a que el ACP calcula los errores con respecto al hiperplano óptimo, en lugar de respecto a una variable particular. Cuando se añaden más atípicos a los datos, el hiperplano óptimo generalmente no cambia drásticamente. Sin embargo, en algunos entornos, la presencia de atípicos puede causar ciertos desafíos. En dichos casos, existen varias técnicas para realizar un ACP robusto. De forma que este método puede ser utilizado para determinar los atípicos obvios en una primera fase. En la llamada segunda fase, estos atípicos pueden eliminarse, y la matriz de covarianzas puede ser construida más robustamente con los datos restantes. Las puntuaciones son entonces recalculadas a partir de la matriz de covarianzas ajustadas. Este método puede también ser aplicado de manera iterativa y, en cada iteración, los atípicos obvios serán eliminados y se construirá un modelo ACP más refinado. Las puntuaciones de atípicos finales serán los niveles de desviación en la última iteración.

2.2. Isolation forest

2.2.1. Introducción

Se propone un método no supervisado llamado Isolation Forest, cuyo propósito es el de la identificación de anomalías en casos en los que los datos no están etiquetados (es decir, cuando no se conoce la clasificación real anomalía-no anomalía de las observaciones).

La implementación del Isolation Forest consiste en la creación de un tipo de *Árbol de decisión* llamado *Isolation Tree* el cual identifica las anomalías aislando los valores atípicos de los datos y está inspirado en el algoritmo de clasificación y regresión conocido como *Random Forest*, aunque al seguir este el aprendizaje supervisado, se encontrarán diferencias en la forma de clasificar los datos. Los isolation forests son un caso especial de ERC-Forest (extremely randomized clustering forest) [11] para agrupamiento o clustering.

A continuación se expone una breve introducción relativa a los árboles de decisión y al método previamente nombrado Random Forest. El trabajo original donde se recoge esta técnica es [12]y, junto con [7], han servido para elaborar gran parte del contenido de esta sección.

2.2.2. Arboles de decisión

Definición. *Un árbol de decisión es una herramienta de clasificación que utiliza un modelo basado en un grafo de árbol. En él, a partir de un conjunto de datos, se fabrican diagramas de construcciones lógicas,*

similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva.

El resultado de un árbol de decisión suele presentarse como un grafo de árbol invertido denominado diagrama de árbol. Dicho diagrama de árbol consiste en una estructura basada en un diagrama de flujo compuesto de tres tipos de nodos: nodos raíz, nodos internos y nodos hoja ó terminales, unidos a su vez por ramas. En él se parte del nodo raíz, el cual agrupa todos los datos a la vez que presenta un primer "test". Este se halla unido a los nodos internos, cada uno de ellos presenta un test para un atributo o predictor. Esta unión se realiza mediante ramas de forma que cada rama simboliza un resultado del test. El diagrama finaliza al alcanzar los nodos hoja, cada nodo hoja representa una etiqueta de clase. Las rutas de raíz a hoja representan reglas de clasificación.

Su objetivo consiste en encontrar el mejor predictor. Para ello se realizan dos tipos de búsqueda. En primer lugar, para cada predictor se consideran todas las posibles particiones de sus valores y, a continuación, se elige la mejor. Esta elección se basa en que, para cada partición de un predictor dado, se calcula una suma de cuadrados para ambas divisiones resultantes y se suman. Dicha suma será menor o igual que la suma de cuadrados original de la variable respuesta. Se dice que una partición es la “mejor” para cada predictor cuando es la que más reduce la suma de cuadrados.

En segundo lugar, una vez hallada la mejor partición de cada predictor, se determina la mejor partición global. Para ello se sigue el mismo criterio previamente utilizado en la suma de cuadrados. Al seleccionar la mejor partición, y siguiendo dicho criterio, se elige implícitamente el mejor predictor.

Una vez realizada la búsqueda de dos pasos, la partición obtenida se utiliza para dividir el conjunto de datos en el nodo raíz, de forma que habrá dos subconjuntos de datos. El siguiente paso consistirá en utilizar el mismo criterio para cada subconjunto de los datos, encontrando la mejor partición y los mejores predictores. Este proceso puede realizarse recursivamente hasta que no haya una reducción significativa del error de la suma de cuadrados.

Notar que el resultado del proceso consiste en un particionamiento recursivo de los datos, que puede ser representado en un marco de funciones base, las cuales son variables indicadoras definidas por las mejores particiones. Una vez determinadas, se obtienen sus coeficientes de regresión y su bondad de ajuste, obtenidos de una regresión de la variable respuesta sobre las funciones base.

2.2.3. Random Forest

El método Random Forest pertenece a la familia de algoritmos de consenso. En él se comienza con la construcción de un gran número de árboles de decisión individuales que operan como un conjunto. Cada árbol individual contiene muestras (que siguen distribución) bootstrap de un conjunto de datos, y la construcción de cada árbol se realiza a partir de una muestra aleatoria de casos, y cada partición por una muestra aleatoria de predictores. Por último, el proceso de clasificación se realiza atendiendo a la clase mayoritaria obtenida del conjunto total de árboles. A este conjunto de árboles es a lo que se conoce como *Random Forest*.

Como se ha señalado anteriormente, Random Forest utiliza árboles de decisión como pilar de su construcción. Una ventaja de este método es su flexibilidad y solidez. Además, son capaces de trabajar con grandes cantidades de predictores, incluso más que observaciones, lo cual puede llevar a una reducción del sesgo. Se puede extraer una mayor cantidad de información del proceso de ajuste en comparación con el uso de modelos convencionales, como los modelos de regresión. Otra sutil ventaja es que se pueden evaluar distintos conjuntos de predictores para diferentes particiones, por lo cual se pueden aplicar distintos modelos en caso necesario.

El objetivo de encontrar un papel para predictores altamente especializados es una tarea para árboles muy grandes y densos, lo cual suele parecer una gran estrategia. Sin embargo, los árboles grandes pueden llevar a resultados inestables cuando hay un número sustancial de predictores poco relacionados con la respuesta y altamente correlacionados entre ellos. Esto conduce a un problema de multicoliealidad de

forma que la inestabilidad es demasiado grande como para ser igualada lo suficientemente rápido. Es por esto por lo que, en la práctica, pueda ser más eficiente en determinadas ocasiones trabajar con árboles de menor tamaño.

2.2.4. Funcionamiento del Isolation Forest

Un isolation forest consiste en una combinación de isolation trees, los cuales son árboles binarios propios, es decir, cada nodo de los árboles tiene exactamente cero o dos nodos hijo. Cada árbol individual se genera con distintas combinaciones de particiones realizadas aleatoriamente. Asumiendo que todos los datos del conjunto son distintos entre si, cada caso se aislará en un nodo externo una vez el árbol se haya desarrollado por completo. En este caso el número de nodos externos será igual al número de datos, n, y el número de nodos internos de $n-1$, y el número total de nodos del árbol será $2n-1$. Cada isolation tree crea grupos jerárquicos de los datos y cada uno de ellos está definido por intervalos de valores elegidos por las particiones. El volumen de estos grupos se reduce en la mitad en cada partición.

Dicho particionamiento se realiza hasta que ocurra una de las siguientes situaciones:

(i) el árbol alcance la altura límite

(ii) el tamaño de la muestra es igual a 1

(iii) todos los datos de la muestra tienen el mismo valor

El método se aprovecha de las propiedades de las anomalías, como son su escasez o sus distintivos valores en determinados atributos, lo cual las hace más susceptibles a ser aisladas del resto de los datos. Las ramas de los árboles que contienen los valores atípicos son notablemente menos profundas, ya que estos datos están situados en regiones aisladas y con escasez de casos. De esta forma la distancia de la hoja a la raíz se utiliza como medición de atípicos:

Definición. *La longitud de ruta $h(x)$ de un punto x se mide por el número de nodos que este recorre en un isolation tree, desde el nodo raíz hasta que termina su recorrido en un nodo terminal.*

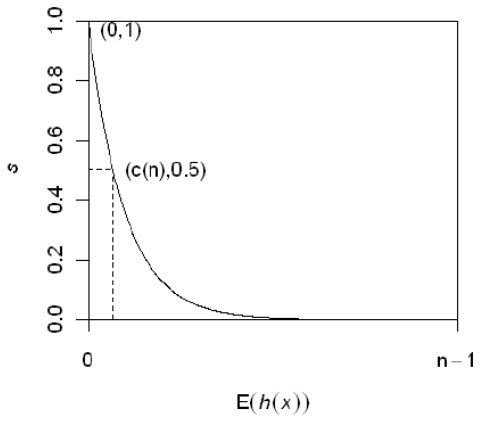
De esta forma, el Isolation Tree identificará las anomalías como puntos con una longitud de ruta notablemente más corta. El último paso del modelo consistirá en promediar las longitudes de ruta de cada punto de los datos.

Nota 3. La ruta o camino recorrido de la raíz a la hoja define un subespacio de dimensión variable. (Los isolation forests están muy relacionados con la subspace outlier detection). Cada rama corresponde a un subespacio local de una región de los datos, dependiendo de los atributos seleccionados para realizar las particiones. De esta forma, las rutas pequeñas corresponden a subespacios de menores dimensiones donde se aislan los datos atípicos. Cuantas menos dimensiones sean necesarias para aislar un punto, más probable será que la anomalía sea fuerte en ese punto. Dicho de otra forma, los isolation forests trabajan bajo el supuesto implícito de que es más probable ser capaz de aislar atípicos en subespacios de menor dimensión creados mediante particiones aleatorias.

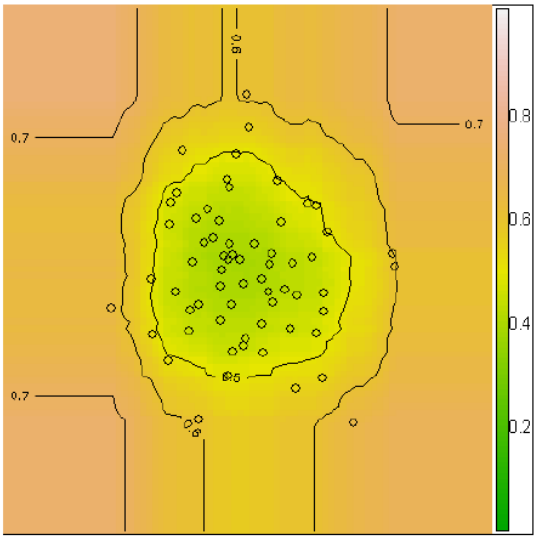
Cualquier método de detección necesita de una puntuación de anomalías. La dificultad de derivar dicha puntuación de $h(x)$ es que, mientras que la máxima altura del isolation tree crece en orden de n, la altura promedio crece en orden de $\log n$.

Debido a que los isolation trees tienen una estructura similar a los árboles binarios de búsqueda [13] (Binary Search Tree, BST, en inglés), la estimación de la media de $h(x)$ para nodos externos es la misma que la búsqueda fallida en un BST [14]. Es por esto por lo que se utiliza el análisis de BST para estimar la longitud de ruta media del isolation tree. Dado un conjunto de datos de n casos, el cálculo de la longitud de ruta media de búsqueda fallida en BST está representada por

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}, \tag{2.6}$$



(a) Figure 2 correspondiente a [12]



(b) Figure 3 correspondiente a [12]

donde $H(i)$ es el número armónico y puede ser aproximado por $\ln(i)+\gamma$, con $\gamma \approx 0{,}5772\ldots$ (constante de Euler). Como $c(n)$ es la media de $h(x)$ dado n, se utiliza para normalizar $h(x)$. La puntuación de anomalías s de una instancia x se define como

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{2.7}$$

donde $E(h(x))$ es la esperanza de $h(x)$ de una colección de isolation trees. En la ecuación 2.7 :

- cuando $E(h(x)) \longrightarrow c(n)$, $s \longrightarrow 0{,}5$;
- cuando $E(h(x)) \longrightarrow 0$, $s \longrightarrow 1$;
- y cuando $E(h(x)) \longrightarrow n-1$, $s \longrightarrow 0$.

Notar que s es monótono a $h(x)$. En la figura 2.3a, se ilustra la relación entre $E(h(x))$ y s, y las siguientes condiciones donde $0 < s \leq 1$ para $0 < h(x) \leq n-1$. Usando la puntuación de anomalías s se pueden considerar las siguientes valoraciones

(a) si las instancias devuelven una s muy cercana a 1, entonces serán claramente anomalías

(b) si las instancias obtienen una s mucho menor que 0,5, entonces serán consideradas como instancias normales

(c) si todas las instancias devuelven $s \approx 0{,}5$, entonces la muestra al completo no tiene anomalías distinguibles

Se puede obtener una curva de nivel de la puntuación de anomalías usando una muestra en una cuadrícula a través de una colección de isolation trees, facilitando un análisis detallado del resultado de detección. La figura 2.3b muestra un ejemplo de dicha curva de nivel, permitiendo visualizar e identificar anomalías en el espacio de instancias.

El isolation forest, como modelo que utiliza isolation trees, se caracteriza por

(a) identificar las anomalías como puntos con longitudes de ruta más cortas,

(b) constar de múltiples árboles actuando como “expertos” para enfocar distintas anomalías, y

(c) no tener la necesidad de aislar todos los casos normales

Contrario a los métodos existentes donde el tamaño de las muestras de datos son más convenientes, el método de aislamiento funciona mejor cuando el tamaño de la muestra es pequeño. Las muestras de gran tamaño reducen la habilidad del isolation forest para aislar anomalías con claridad. Así, el submuestreo proporciona un ambiente favorable para el buen funcionamiento del algoritmo.

Los problemas más comunes a los que se ha de enfrentar el Isolation Forest son: el efecto de *swamping*, identificación errónea de casos normales como anomalías, y el de *masking*, existencia de gran número de anomalías lo cual oculta su presencia.

Estos problemas han sido estudiados extensamente en detección de anomalías. Cuando los casos normales se sitúan muy próximos a las anomalías, el número de particiones requeridas para la conseguir separación de anomalías aumenta; lo cual dificulta la distinción entre anomalías y casos normales. Cuando un grupo de anomalías es largo y denso, también aumenta el número de particiones necesario para aislar anomalías. Bajo estas circunstancias, las evaluaciones que utilizan estos árboles tienen longitudes de ruta más altas, haciendo más difícil de detectar las anomalías. Notar que tanto swamping como masking resultan de los casos de detección de anomalías con demasiados datos. Las características exclusivas propias de los isolation trees que permiten al isolation forest construir un modelo parcial mediante submuestreo, aliviando a su vez los efectos de swamping y masking, son

1. el submuestreo controla el tamaño de los datos, lo cual ayuda al isolation forest a aislar mejor ejemplos de anomalías

2. cada isolation tree puede especializarse, igual que cada submuestra influye diferentes conjuntos de anomalías o incluso sin anomalías.

Otra posible mejora que ofrece este algoritmo con el objetivo de mejorar su rendimiento consiste en que el árbol no alcance su altura máxima. Esto se consigue deteniendo el crecimiento de un nodo tan pronto como dicho nodo contenga instancias duplicadas o exceda una determinada altura. Con el fin de estimar la longitud de ruta de los puntos en dichos nodos, se utilizará $c(n)$ que, como ya hemos visto, calcula la media de $h(x)$, debido a que los puntos en esos nodos no han sido materializados al no alcanzar el árbol su altura máxima. La terminación temprana es una mejora centrada en la eficiencia y se puede elegir que los árboles alcancen su altura máxima si se considera necesario.

El proceso de detección de anomalías con Isolation Forest consta de dos etapas. En la primera, denominada fase de entrenamiento, se construyen los isolation trees mediante submuestras del conjunto de datos de entrenamiento (entorno al 70 % de los datos originales). En la segunda, denominada fase de testeo, se analizan todos los datos utilizando los isolation trees para obtener una puntuación de anomalía en cada caso.

En la fase de entrenamiento, los isolation trees se construyen realizando un particionamiento recursivo aleatorio de los datos de entrenamiento sin reemplazamiento hasta que todos los datos estén aislados o se haya alcanzado una altura impuesta. Notar que el límite de la altura l se selecciona automáticamente por el tamaño de las submuestras ψ: $l = ceiling(\log_2 \psi)$ que es aproximadamente la altura promedio de un árbol.

En el algoritmo iForest [2] hay dos parámetros de entrada, los cuales son el tamaño de las submuestras ψ y el número de árboles.

El tamaño de las submuestras ψ controla el tamaño del conjunto de datos de entrenamiento. Empíricamente, se ha demostrado que dar a ψ los valores 2^8 o 256 suele otorgar suficiente información para aplicar la detección de anomalías en una amplia variedad de datos.

A su vez, el número de árboles t controla en tamaño del conjunto (o bosque). Ha sido demostrado que las longitudes de ruta suelen converger correctamente con $t = 100$.

Algorithm 2 $iForest(X, t, \psi)$

Input: X - datos cargados, t - número de árboles, ψ - tamaño de la submuestra
Output: conjunto de t *iTrees*

Initialite *Forest*
seleccionar la altura límite $l = ceiling(\log_2 \psi)$
for $i = 1$ to t **do**
    $X' \leftarrow sample(X, \psi)$
    $Forest \leftarrow Forest \cup iTree(X', 0, l)$
return *Forest*

Algorithm 3 $iTree(X, e, l)$

Input: X - datos cargados, e - altura actual del árbol, l - altura límite
Output: un *iTree*

if $e \geq l$ or $|X| \leq 1$ **then**
    return $exNode\{Size \leftarrow |X|\}$
else
    sea Q una lista de atributos en X
    seleccionar aleatoriamente un atributo $q \in Q$
    seleccionar aleatoriamente un punto de separación p entre los valores *max* y *min* del atributo q en X
    $X_l \leftarrow filter(X, q < p)$
    $X_r \leftarrow filter(X, q \geq p)$
    return $inNode\{Left \leftarrow iTree(X_l, e+1, l),$
    $Right \leftarrow iTree(X_r, e+1, l),$
    $SplitAtt \leftarrow q,$
    $SplitValue \leftarrow p\}$

Una vez terminada la fase de entrenamiento se devuelve una colección de árboles lista para comenzar la fase de entrenamiento.

En la fase de testeo, se derivará una puntuación de anomalías obtenida de la longitud de ruta esperada $E(h(x))$ para cada punto de los datos. Utilizando la funcion *PathLength* se obtendrá una única longitud de ruta $h(x)$ contando el número de nodos e desde la raíz hasta el nodo terminal en el que se encuentre el dato x.

Algorithm 4 $PathLength(x, T, e)$

Input: x - un punto de los datos, T - un iTree, e - altura actual del árbol
Output: longitud de ruta de x

1: **if** T es un nodo externo **then**
    return $e + c(T.size)$ ▷ $c(.)$ está definido en 2.6
2: $a \leftarrow T.splitAtt$
3: **if** $x_a < T.splitValue$ **then**
    return $PathLength(x, T.left, e+1)$
4: **else** $\{x_a \geq T.splitValue\}$
    return $PathLength(x, T.right, e+1)$

Este método de agrupamiento es particularmente efectivo y suele proveer de buenos resultados. La versión básica del procedimiento isolation tree, una vez completado, está libre de parámetros. Esta característica constituye un ventaja significativa en problemas no supervisados como la detección de atípicos. La complejidad computacional de caso promedio (cantidad de algún recurso computacional utilizado por el algoritmo, promediado sobre todas las entradas posibles) es del orden de $O(n\log(n))$ y la complejidad espacial (es la cantidad de memoria requerida para resolver una instancia del problema computacional en función de las características de la entrada) es del orden de $O(n)$ para cada isolation tree. Siempre puede mejorarse la eficiencia computacional y al mismo tiempo suele mejorarse la exactitud mediante submuestreo.

Nota 4. El isolation forest es un método eficiente, lo cual es notable considerando el hecho de que la mayoría de los métodos respectivos a subespacios son altamente costosos computacionalmente.

Aunque el submuestreo produce beneficios respectivos a la diversidad en el isolation forest, como ya se ha comentado, también suele haber efectos negativos respectivos al sesgo en los detectores base. Estos efectos negativos afloran como resultado de la forma en que se clasifican los puntos externos a la muestra tomada en los isolation forests. Notar que los puntos externos a la muestra podrían aplicarse a regiones vacías en la submuestra. Desde la perspectiva del isolation tree (construido a partir de la submuestra), esta región vacía podria ser fusionada con una región normal del espacio, y por lo tanto la puntuación correspondería a la profundidad de esta región normal. Sin embargo, en la mayoría de los casos, las regiones vacías suelen corresponder a los bordes de regiones normales. Los bordes de regiones normales también tienden a ser menos densos y recibir puntuaciones de menor profundidad, lo cual es conveniente (sigue un ejemplo de cómo afecta, lo cual puede ser interesante).

2.2.5. Mejoras para la selección de subespacios

El modelo se mejora mediante una preselección adicional de características con una medida de curtosis. La curtosis de un conjunto de valores de características $x_1, \ldots, x_n$ se calcula estandarizando dichos valores, de forma que se obtiene $z_1, \ldots, z_n$ con media cero y desviación típica unidad, de forma que su relación se expresa, para $i = 1, \ldots, n$, como:

$$z_i = \frac{x_i - \mu}{\sigma},$$

donde μ es la media y σ es la desviación típica de $x_1, \ldots, x_n$. Una vez hecho esto, se calcula la curtosis de la siguiente forma

$$K(z_1, \ldots, z_n) = \frac{\sum_{i=1}^{n} z_i^4}{n}.$$

Las características que son uniformes obtendrán gran nivel de curtosis. Además, la computación de la curtosis se puede ver como una medida de selección de características para la detección de anomalias. En [12] se preselecciona un subconjunto de atributos basándose en los valores univariantes de curtosis y entonces construye el random forest tras desechar aquellos atributos. Notar que esto resulta en selección de subespacios globales; sin embargo, el método de particionamiento aleatorio aún es capaz de explorar diferentes subespacios locales, aunque aleatoriamente (como feature bagging).

2.2.6. Comparación con otros métodos

La mayoría de modelos existentes enfocados en la detección de anomalías construyen un perfil de instancias normales, entonces se identifican los puntos que no se adaptan a dicho perfil, los cuales se consideran anomalías. En referencia a [7], algunos ejemplos notables que utilizan esta técnica son métodos probabilísticos y estadísticos, métodos basados en proximidad y métodos lineales.

Las dos mayores desventajas respecto a esta técnica son:

(i) El detector de anomalías está optimizado para perfilar instancias normales, pero no para detectar anomalías; como consecuencia, los resultados de la detección podrían no ser tan buenos como se espera, causando demasiados falsos positivos (teniendo instancias normales clasificadas como anomalías) o demasiadas anomalías sin detectar.

(ii) Muchos de los métodos existentes están restringidos a datos de baja dimensión y conjuntos de datos de tamaño pequeño debido a su alta complejidad computacional.

Además de la diferencia clave del aislamiento frente a la evaluación por perfil, el isolation forest se distingue de los existentes modelos en lo siguiente:

- Sus características de aislamiento permiten construir modelos parciales y aprovechar el submuestreo a un alcance que no es posible para los modelos existentes. Como una parte extensa de un isolation tree que aisla puntos normales no es necesaria para la detección de anomalías; no es necesario construirlo.
- No utiliza medidas de distancia o de densidad para detectar anomalías. Esto elimina el mayor coste computacional del cálculo de la distancia en todos los métodos basados en distancias o en densidad.
- Tiene una complejidad temporal lineal con constante baja y poca memoria requerida.
- Tiene la capacidad de aumentar proporcionalmente para tratar grupos de datos extremadamente grandes y problemas de anta dimensión con un gran número de atributos irrelevantes.

Capítulo 3

Aplicación práctica de los algoritmos

En este capítulo se comentan los conjuntos de datos utilizados para la parte práctica del trabajo, se explican las librerias de *R* utilizadas en la aplicación de ambos algoritmos y se evalúan los resultados obtenidos.

3.1. Presentación de los datos

La evaluación de ambos algoritmos se realizará mediante la experimentación con dos conjuntos de datos distintos, ambos han sido obtenidos de [15]. Los conjuntos de datos utilizados son:

- **Pendigits.** Se trata de un conjunto de clasificación multiclase formado por 16 atributos numéricos y creado a partir de la escritura de 250 digitos por parte de 44 autores. Los distintos atributos contienen las coordenadas de 8 puntos de cada una de las escrituras realizadas. Este contiene una muestra de 6870 datos de los cuales 156 (2,27 %) son anomalías.
- **Cardiotocography (Cardio).** Este conjunto está formado por 21 atributos numéricos obtenidos de medidas de cardiotocografías, ratios de frecuencia cardíaca fetal y contracciones uterinas, clasificadas por obstetras experimentados. Contiene una muestra de 1831 datos de los cuales 176 (9,6 %) son anomalías.

3.2. Implementación y estrategias de identificación

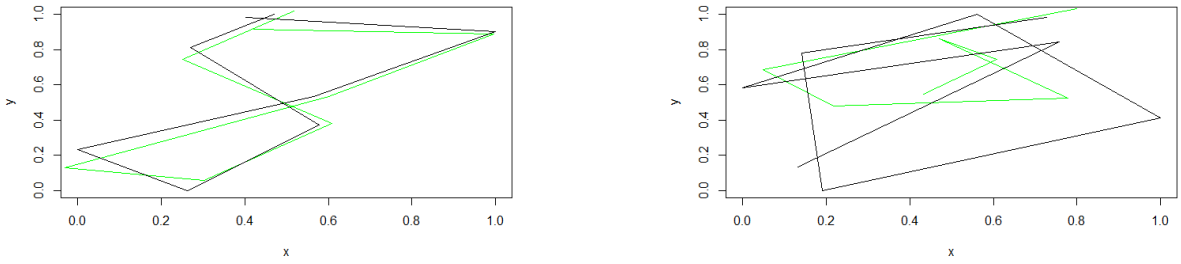
3.2.1. Análisis de componentes principales

Para la aplicación del PCA se utilizará la función $prcomp()$ de *R*. Una vez aplicada dicha función se aplican las funciones $reconstruct\_prcomp()$ y $error\_reconstruccion\_prcomp()$ previamente definidas, las cuales han sido obtenidas de [16] y optimizadas mediante un análisis previo. La primera función revierte la reducción de la dimensión realizada por el PCA basándose en las componentes principales indicadas mientras que la segunda calcula el error de reconstrucción producido por dicha reversión. Dicho error de reconstrucción será utilizado como medidor de anomalías.

Al no conocer el número óptimo de componentes principales pero sí contar con las etiquetas de atípicos se utilizará el índice de Youden [17], junto el área bajo la curva *ROC* (en inglés, Receiver operating characteristic) [18] para evaluar el rendimiento y la calidad del modelo.

3.2.2. Isolation Forest

Para la aplicación del Isolation Forest se utiliza la librería *isotree* de *R* y, en particular, su función $isolation.forest()$, la cual construye un isolation forest atendiendo al tamaño de muestra y al número de árboles deseados. Dichos parámetros serán optimizados previo a su aplicación, utilizando una pequeña



(a) Dato “normal”

(b) Dato anómalo

Figura 3.1: En negro los datos del conjunto previos a la aplicación del modelo PCA; en verde posteriormente a la reconstrucción del modelo.

submuestra aleatoria del conjunto de datos y un cantidad asequible de árboles para que el proceso se realice lo más rápidamente posible.

Además, debido a la complejidad de realización de las fases de entrenamiento y testeo en el código, se opta por un modelo de Isolation Forest más directo, el cual se crea sobre una submuestra y se aplica a todos los datos del conjunto, obteniendo su longitud de ruta.

La obtención de la longitud de ruta de cada uno de los datos servirá como medición de anomalías; además un análisis del área bajo la curva *ROC* mediante el índice de Youden servirá para obtener la bondad de la aplicación del modelo. Por último, se clasificarán los datos como anomalías o no anomalías.

3.3. Conjunto de datos *Pendigits*

En primer lugar se carga el conjunto de *Pendigits*. Se dibujan dos puntos del conjunto un dato “normal” en 3.1a y un dato anómalo en 3.1b previos a la aplicación del modelo PCA y una vez realizada la reconstrucción. En dichos dibujos puede observarse que la reconstrucción del dato normal es muy similar al dato original mientras que la reconstrucción del dato anómalo no se parece demasiado al original, esto se debe a que el error de reconstrucción de una anomalía tiende a ser mayor que el de un dato normal. Se buscará un resultado similar al realizar la reconstrucción para facilitar la identificación de las anomalías.

A continuación se aplica el modelo PCA y se calculan las posibles reconstrucciones con diferentes combinaciones de componentes principales, las se pueden observar en las tablas 3.1a y 3.1b. Una rápida observación permite observar que el mejor valor del área bajo la curva *ROC* obtenido es 0,9337 y las componentes utilizadas son tanto las 12 primeras como las 5 últimas. Debido a que la mayoría de los datos atípicos no suelen tener atributos anómalos en las direcciones principales, sino que se recogen en las últimas componentes, se decide elegir las 5 últimas para realizar la reconstrucción.

Una vez realizada dicha reconstrucción y calculados los errores de reconstrucción para todo punto de los datos se puede observar la diferencia en las funciones de densidad dependiendo de si los datos son anómalos o no en la Figura 3.2. En ella se puede observar que los datos clasificados por el modelo como normales obtienen, en promedio, menores errores de reconstrucción que los datos clasificados como anómalos, consiguiendo así lo buscado.

Se pueden comprobar los resultados del modelo comparando los datos clasificados con los etiquetados en la matriz de confusión 3.2.

A continuación se aplica el modelo Isolation Forest con distintas combinaciones de cantidad de árboles y tamaños de muestra con el objetivo de optimizar el resultado lo máximo posible, dichos resultados pueden revisarse en la Tabla 3.3. A pesar de conseguir muy buenos resultados con la mayoría de combinaciones, el resultado óptimo se consigue con una submuestra de tamaño 500 y un número de árboles

Componentes	AUC	Specificity	Sensitivity
1	0.8882	0.7140	0.8846
1:2	0.8907	0.8261	0.9230
1:3	0.9068	0.8647	0.8910
1:4	0.894	0.8339	0.8846
1:5	0.8753	0.8199	0.8782
1:6	0.9036	0.8529	0.8782
1:7	0.8915	0.8388	0.8782
1:8	0.9172	0.8777	0.8846
1:9	0.9176	0.8550	0.9166
1:10	0.9142	0.8419	0.9230
1:11	0.9305	0.8500	0.9743
1:12	0.9337	0.8544	0.9871
1:13	0.9336	0.8509	0.9871
1:14	0.9329	0.8497	0.9871
1:15	0.9326	0.8442	0.9935

(a) Primeras componentes de *Pendigits* (orden ascendente).

Componentes	AUC	Specificity	Sensitivity
16	0.9316	0.8430	0.9935
16:15	0.9326	0.8442	0.9935
16:14	0.9329	0.8497	0.9871
16:13	0.9336	0.8509	0.9871
16:12	0.9337	0.8544	0.9871
16:11	0.9305	0.8500	0.9743
16:10	0.9142	0.8419	0.9230
16:9	0.9176	0.8550	0.9166
16:8	0.9172	0.8777	0.8846
16:7	0.8915	0.8388	0.8782
16:6	0.9036	0.8529	0.8782
16:5	0.8753	0.8199	0.8782
16:4	0.894	0.8339	0.8846
16:3	0.9068	0.8647	0.8910
16:2	0.8907	0.8261	0.9230

(b) Últimas componentes de *Pendigits* (orden descendente).

Tabla 3.1: Posibles combinaciones de componentes principales para realizar la reconstrucción en el conjunto *Pendigits*.

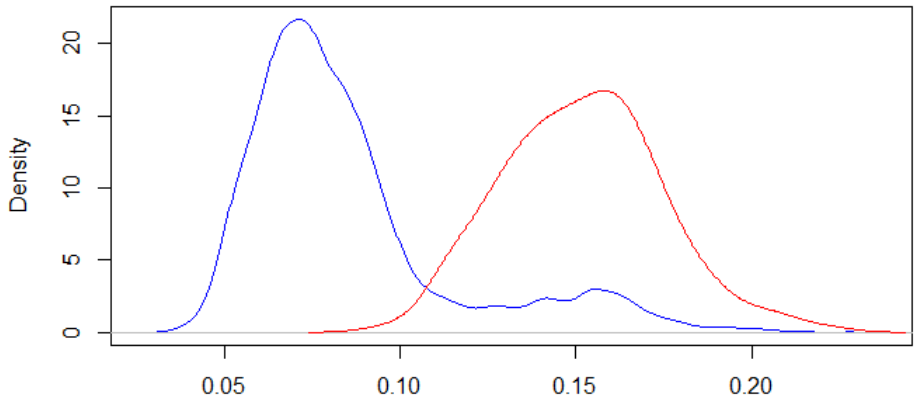


Figura 3.2: Densidades de los errores de reconstrucción. En azul la densidad de los errores de los datos no anómalos; en rojo la densidad de los errores de los datos anómalos.

igual a 250. Se obtiene un gráfico de las funciones de distribución de las longitudes de ruta respectivas a los datos anómalos y no anómalos. En la Figura 3.3 puede observarse que las longitudes de ruta de los datos clasificados como anómalos son, en promedio, menores que las de los datos clasificados como normales.

Para comprobar el resultado de dicha aplicación se comprueba el área bajo la curva *ROC*, la cual da un valor de 0.9574, un valor considerablemente elevado. Además, el modelo con estos parámetros obtiene unos valores de $specificity$ de 0.8810 y $sensitivity$ de 0.9936.

Por último, una vez aplicado el modelo óptimo se presenta la matriz de confusión 3.4 con los resultados obtenidos contrastados con los datos etiquetados.

Se puede observar comparando los datos obtenidos de ambos modelos, aunque ambos son bastante buenos, que el área bajo la curva *ROC* es notablemente superior en el caso del Isolation Forest. Además, el Isolation Forest detecta un número de anomalías prácticamente igual, sin embargo detecta una cantidad notablemente menor de falsos positivos. De esta forma, se puede concluir que el Isolation Forest es objetivamente mejor modelo para este conjunto de datos.

Etiquetado \ Predicción	0	1	Totales
0: No anómalo	5737	2	5739
1: Anómalo	977	154	1131
Totales	6714	156	6870

Tabla 3.2: Clasificación de atípicos respectivos a la reconstrucción con las 5 últimas componentes principales respectiva al conjunto *Pendigits*.

Nº. árboles \ Tamaño de muestra	250	500	750	1000
100	0.9325	0.9521	0.9421	0.9532
250	0.9486	0.9574	0.945	0.9521
500	0.9536	0.955	0.9476	0.95
750	0.9536	0.9536	0.9478	0.9507
1000	0.955	0.9515	0.9481	0.9492

Tabla 3.3: Distintas aplicaciones del modelo Isolation Forest realizadas en el conjunto *Pendigits*. La tabla recoge el AUC resultante de la combinación del tamaño muestral con el número de árboles utilizados en el Isolation Forest.

3.4. Conjunto de datos *Cardio*

De la misma forma que en el anterior conjunto de datos, en primer lugar se construye el modelo PCA aplicado al conjunto *Cardio* y se construyen las posibles reconstrucciones, las cuales pueden ser observadas en 3.5a y 3.5b. Se puede observar que el mejor valor de área bajo la curva *ROC* es 0.9535 respectivo a las 7 primeras componentes y a las 15 últimas. Por lo comentado anteriormente se eligen las 15 últimas componentes principales para realizar la reconstrucción.

Así, realizando la reconstrucción óptima con las 15 últimas componentes, se calculan sus respectivos errores y se dibujan las funciones de densidad de los datos clasificados como normales y como atípicos de forma separada. En esta figura 3.4 se observa que los errores de reconstrucción de los datos anómalos son , en promedio, superiores a los no anómalos.

Pueden compararse los datos clasificados por el modelo con los etiquetados en la matriz de confusión 3.6.

A continuación se aplica el modelo Isolation Forest con distintas cantidades de árboles y tamaños de muestra tratando de optimizar el resultado del modelo, se pueden observar dichos resultados Tabla 3.7.

Los resultados obtenidos son, en general, bastante buenos aunque el resultado óptimo ha sido conseguido con una submuestra de tamaño 450 y un número de 250 árboles. Pueden observarse, en el gráfico 3.5, las funciones de distribución de las longitudes de ruta respectivas a los datos clasificados como anómalos y no anómalos. Notar que las longitudes de ruta de los datos clasificados como normales son, en promedio, mayores que las de los datos clasificados como anomalías.

Se comprueba la bondad del ajuste del modelo verificando el valor del área bajo la curva *ROC*, que es 0.9336, con unos valores de $specificity$ de 0.7523 y $sensitivity$ de 0.9943. Posteriormente se calcula la matriz de confusión 3.8, lo cual es posible gracias a disponer del etiquetado de atípicos.

Se puede observar que el área bajo la curva *ROC* del modelo PCA es ligeramente superior al del Isolation Forest, sin embargo la $sensitivity$ perteneciente al Isolation Forest es muy cercana a 1, de forma que el Isolation Forest detectará un mayor porcentaje de atípicos a base de detectar una mayor cantidad de falsos positivos. De esta forma los datos clasificados como no anómalos se verán menos alterados que los clasificados por el modelo PCA, lo cual es preferible en la detección de anomalías.

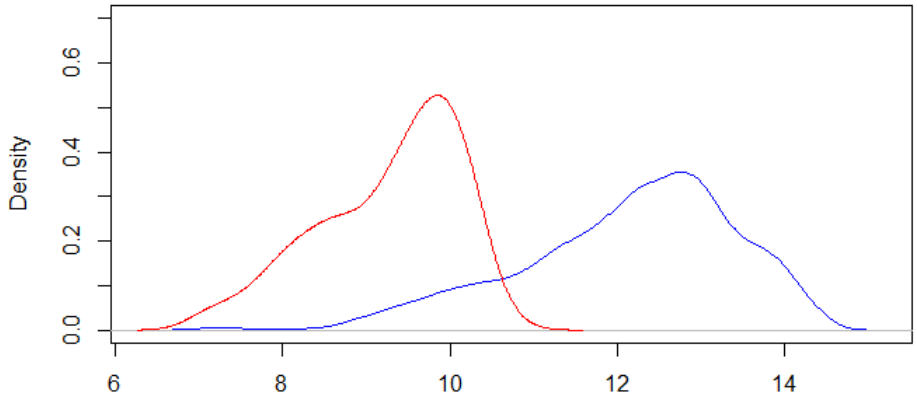


Figura 3.3: Densidades de las longitudes de ruta. En azul la densidad de las longitudes de los datos no anómalos; en rojo la densidad de las longitudes de los datos anómalos.

Etiquetado \ Predicción	0	1	Totales
0: No anómalo	5915	1	5916
1: Anómalo	799	155	854
Totales	6713	156	6870

Tabla 3.4: Clasificación de atípicos perteneciente al modelo Isolation Forest con una submuestra de tamaño 500 y un número de árboles de 250 respectiva al conjunto *Pendigits*.

Componentes	AUC	Specificity	Sensitivity
1	0.9481	0.8586	0.9375
1:2	0.9246	0.8694	0.8522
1:3	0.9326	0.8422	0.8977
1:4	0.9455	0.8700	0.9261
1:5	0.9531	0.8592	0.9318
1:6	0.9516	0.8465	0.9431
1:7	0.9535	0.8664	0.9204
1:8	0.9522	0.8132	0.9772
1:9	0.9513	0.8223	0.9659
1:10	0.9478	0.8108	0.9715
1:11	0.9518	0.8338	0.9715
1:12	0.9489	0.8211	0.9715
1:13	0.9511	0.8416	0.9602
1:14	0.9509	0.8320	0.9772
1:15	0.9509	0.8277	0.9772
1:16	0.9505	0.8223	0.9829
1:17	0.95	0.8205	0.9829
1:18	0.9501	0.8193	0.9829
1:19	0.9501	0.8157	0.9829
1:20	0.9501	0.8151	0.9829

(a) Primeras componentes de *Cardio* (orden ascendente).

Componentes	AUC	Specificity	Sensitivity
21	0.95	0.8151	0.9829
21:20	0.9501	0.8151	0.9829
21:19	0.9501	0.8157	0.9829
21:18	0.9501	0.8193	0.9829
21:17	0.95	0.8205	0.9829
21:16	0.9505	0.8223	0.9829
21:15	0.9509	0.8277	0.9772
21:14	0.9509	0.8320	0.9772
21:13	0.9511	0.8416	0.9602
21:12	0.9489	0.8211	0.9715
21:11	0.9518	0.8338	0.9715
21:10	0.9478	0.8108	0.9715
21:9	0.9513	0.8223	0.9659
21:8	0.9522	0.8132	0.9772
21:7	0.9535	0.8664	0.9204
21:6	0.9516	0.8465	0.9431
21:5	0.9531	0.8592	0.9318
21:4	0.9455	0.8700	0.9261
21:3	0.9326	0.8422	0.8977
21:2	0.9246	0.8694	0.8522

(b) Últimas componentes de *Cardio* (orden descendente).

Tabla 3.5: Posibles combinaciones de componentes principales para realizar la reconstrucción en el conjunto *Cardio*.

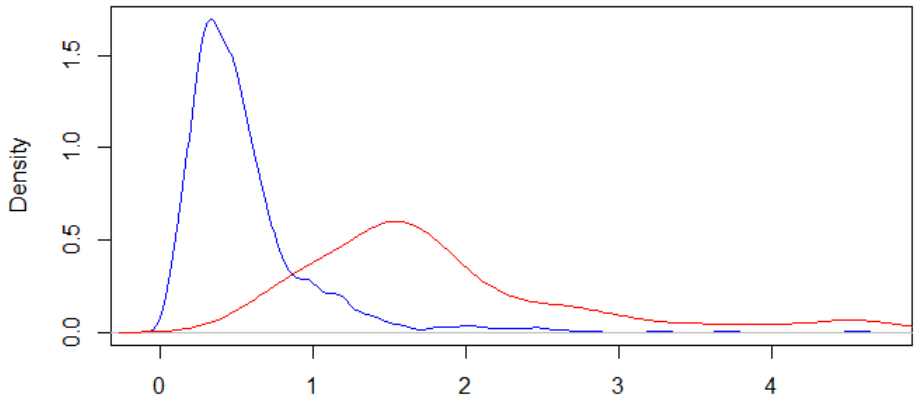


Figura 3.4: Densidades de los errores de reconstrucción de los datos clasificados como anómalos (rojo) y los errores de los datos clasificados como normales (azul).

Etiquetado \ Predicción	0	1	Totales
0: No anómalo	1434	14	1448
1: Anómalo	221	162	383
Totales	1655	176	1831

Tabla 3.6: Clasificación de atípicos respectivos a la reconstrucción con las 15 últimas componentes principales respectiva al conjunto *Cardio*.

Nº. árboles \ Tamaño de muestra	150	250	350	450	550
100	0.9251	0.9209	0.921	0.9316	0.9304
250	0.9321	0.9258	0.9245	0.9336	0.9292
350	0.9288	0.9223	0.9215	0.9249	0.9285
450	0.927	0.9265	0.9271	0.9263	0.9289

Tabla 3.7: Distintas combinaciones del modelo Isolation Forest aplicadas al conjunto *Cardio*. La tabla recoge el AUC obtenida de la combinación del tamaño muestral y del número de árboles utilizados en el Isolation Forest.

Etiquetado \ Predicción	0	1	Totales
0: No anómalo	1245	1	1246
1: Anómalo	410	175	585
Totales	1655	176	1831

Tabla 3.8: Clasificación de atípicos respectiva al modelo Isolation Forest con una submuestra de 450 datos y un número de 250 árboles perteneciente al conjunto *Cardio*.

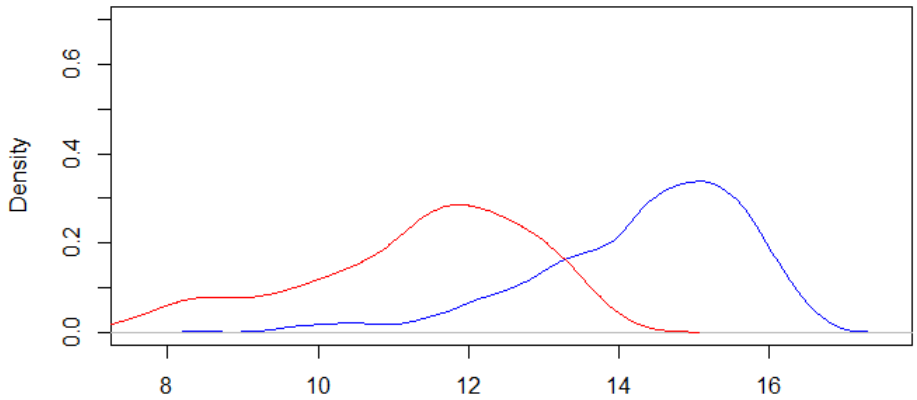


Figura 3.5: Densidades de las longitudes de ruta de los datos clasificados como anómalos (rojo) y los errores de los datos clasificados como normales (azul).

Bibliografía

[1] Machine learning. (19 de agosto de 2022). En *Wikipedia, la enciclopedia libre*. https://en.wikipedia.org/wiki/Machine_learning

[2] FOOTE, K. D. (2021, 3 DE DICIEMBRE). *A Brief History of Machine Learning*. https://www.dataversity.net/a-brief-history-of-machine-learning/.

[3] CHANDOLA, V., BANERJEE, A. Y KUMAR, V. (2009). *Anomaly Detection: A Survey.* ACM Comput. Surv. Volume 41. https://dl.acm.org/doi/10.1145/1541880.1541882

[4] PAUWELS, E. Y AMBEKAR, O. (2011). *One Class Classification for Anomaly Detection: Support Vector Data Description Revisited.* (pp. 25-39). Conference: Advances in Data Mining. Applications and Theoretical Aspects - 11th Industrial Conference, ICDM 2011, New York, NY, USA. https://www.researchgate.net/publication/221338562_One_Class_Classification_for_Anomaly_Detection_Support_Vector_Data_Description_Revisited.

[5] Dorothy E. Denning. (25 de junio de 2022). En *Wikipedia, la enciclopedia libre*. https://en.wikipedia.org/wiki/Dorothy_E._Denning.

[6] PRADEEP, K. Y SURANA, H. (2021, 16 DE AGOSTO). *A Brief History of Anomaly Detection*. https://www.chaosgenius.io/blog/a-brief-history-of-anomaly-detection/

[7] AGGARWAL, C. C. (2017). *Outlier analysis. (2ª ed.)* Springer.

[8] PEÑA, D. (2002), *Análisis de Datos Multivariantes.* McGraw-Hill. Universidad Carlos III de Madrid.

[9] DHIRAJ, K. (2019), *Anomaly Detection Using Isolation Forest in Python*. https://blog.paperspace.com/anomaly-detection-isolation-forest/

[10] CUTLER, A., CUTLER, D. R. Y STEVENS, J. R. (2012). Random Forest. En *Ensemble Machine Learning: Methods and Applications. (1ª ed.)* (pp. 157-176). Springer, Boston, MA.

[11] Moosmann, F., Triggs, B. y Jurie, F. (2006). Fast Discriminative Visual Codebooks using Randomized Clustering Forests. En *Neural Information Processing Systems*. (pp. 985–992). NeurIPS. https://proceedings.neurips.cc/paper/2006

[12] LIU, F. T., TING, K. M. Y ZHI-HUA Z. (2008). *Isolation forest*, IEEE International Conference on Data Mining (ICDM). https://ieeexplore.ieee.org/document/4781136

[13] KNUTH, D. E.(1968). Volume 3: Sorting and Searching, Chapter 6: Searching. En *The Art of Computer Programming. (2ª ed.)* Addison-Wesley. Standford University.

[14] PREISS, B. R. (1999). *Unsuccessful Search.* Consultado el 10 de septiembre de 2022 desde https://book.huihoo.com/data-structures-and-algorithms-with-object-oriented-design-patterns-in-java/html/page308.html

[15] RAYANA, S. (2016). *ODDS Library*. Stony Brook, NY: Stony Brook University, Department of Computer Science. http://odds.cs.stonybrook.edu/

[16] AMAT RODRIGO, J. (FEBRERO 2020). *Detección de anomalías: Autoencoders y PCA*. https://www.cienciadedatos.net/documentos/52_deteccion_anomalias_autoencoder_pca.html

[17] Youden's J statistic. (6 de agosto de 2022). En *Wikipedia, la enciclopedia libre*. https://en.wikipedia.org/wiki/Youden%27s_J_statistic

[18] Receiver operating characteristic. (22 de agosto de 2022). En *Wikipedia, la enciclopedia libre*. https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Anexo

Escribimos en el anexo el código de R que ha sido utilizado para la aplicación de ambos modelos. Solo se mostrará el código del conjunto de datos Pendigits escrito sin su output, ya que el código del conjunto Cardio es muy similar. En primer lugar se cargan los paquetes y los datos.

```
library(R.matlab) # Leer y escribir archivos .mat desde R

## R.matlab v3.7.0 (2022-08-25 21:52:34 UTC) successfully loaded.
See ?R.matlab for help.
##
## Attaching package: 'R.matlab'
## The following objects are masked from 'package:base':
##
##     getOption, isOpen

library(pROC) # Curvas ROC empíricas y paramétricas sin
covariables

## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

library(isotree) # Aplicación rápida y personalizada de Isolation
Forest

pendigits <- readMat("pendigits.mat")
```

Se definen las funciones de reconstrucción y de cálculo del error de reconstrucción, las cuales serán posteriormente utilizadas.

```
reconstruct_prcomp <- function(pca, comp = NULL){

# Esta función reconstruye las mismas observaciones con las que
# se ha creado el PCA empleando únicamente determinadas
# componentes principales.

# Parameters
# ----------
# pca: "prcomp"
#   objeto prcomp con los resultados del PCA.
#
```

```
# comp: "numeric"
#   componentes principales empleadas en la reconstrucción.
#
# Return
# ----------
# "matrix" con la reconstrucción de cada observación. Las
# dimensiones de la matriz son las mismas que las de la matriz
# o dataframe con el que se estrenó el objeto pca.

# Se comprueba que el objeto PCA es de clase "prcomp"
if (class(pca) != "prcomp") {
stop("El objeto PCA debe haber sido creado mediante prcomp()")
}

# Si no se especifica comp, se emplean todas las componentes.
if (is.null(comp)) {
comp <- seq_along(pca$sdev)
}

# Reconstrucción
recon <- as.matrix(pca$x[, comp]) %*%
t(as.matrix(pca$rotation[, comp]))

# Si se ha aplicado centrado o escalado se revierte la
# transformación.
if (pca$scale[1] != FALSE) {
recon <- scale(recon , center = FALSE, scale = 1/pca$scale)
}
if (pca$center[1] != FALSE) {
recon <- scale(recon , center = -1*pca$center, scale = FALSE)
}

return(recon)
}

error_reconstruccion_prcomp <- function(pca, new_data, comp=NULL){

# Esta función calcula el error de reconstrucción de un PCA al
# proyectar y reconstruir las observaciones empleando únicamente
# determinadas componentes principales.
#
# Parameters
# ----------
# pca: "prcomp"
#   objeto prcomp con los resultados del PCA
#
# new_data: "matriz" o "data.frame"
#  nuevas observaciones
#
# comp: "numeric"
#   componentes principales empleadas en la reconstrucción.
```

```
#
# Return
# ----------
# "numeric" vector con el error de reconstrucción de cada
# observación.

# Se comprueba que el objeto PCA es de clase "prcomp"
if (class(pca) != "prcomp") {
stop("El objeto PCA debe haber sido creado mediante prcomp()")
}

# Si no se especifica comp, se emplean todas las componentes
if (is.null(comp)) {
comp <- seq_along(pca$sdev)
}

# Se seleccionan únicamente las componentes en comp
pca$rotation <- pca$rotation[, comp]

proyeciones <- predict(object = pca,
newdata = new_data
)

# Recostrucción
reconstruccion <- as.matrix(proyeciones) %*%
t(as.matrix(pca$rotation))

# Si se ha aplicado centrado o escalado se revierte la
# trasformación
if (pca$scale[1] != FALSE) {
reconstruccion <- scale(reconstruccion , center = FALSE,
scale = 1/pca$scale)
}
if (pca$center[1] != FALSE) {
reconstruccion <- scale(reconstruccion , center =
-1*pca$center, scale = FALSE)
}

# Cálculo del error de reconstrucción
error_reconstruccion <- reconstruccion - new_data
error_reconstruccion <- error_reconstruccion^2
error_reconstruccion <- apply(X = error_reconstruccion,
MARGIN = 1, FUN = mean)

return(error_reconstruccion)
}
```

Se aplica el modelo PCA al conjunto de datos y se evalúa cada una de las principales componentes calculando sus deviaciones típicas, sus varianzas y sus varianzas acumuladas.

```
pendigits.pca<-prcomp(pendigits$X, scale.=TRUE, center=TRUE)
summary(pendigits.pca)
ncp<-dim(pendigits$X)[2]
```

Se construye una representación grafica de la relación entre las cinco primeras componentes principales y las 5 últimas, con una muestra de 1000 datos.

```
pairs(pendigits.pca$x[1:1000,1:5], col=(pendigits$y)[1:1000]+1)
```

```
pairs(pendigits.pca$x[1:1000,12:16], col=(pendigits$y)[1:1000]+1)
```

A continuación, se realiza un análisis de las posibles reconstrucciones utilizando las primeras componentes principales (utilizando desde únicamente la primera hasta utilizar las 15 primeras).

```
for (i in 1:ncp-1)
{
# calculo de la reconstruccion con i componentes
print(c("numero de componentes", i))
pendigits.reconst<-reconstruct_prcomp(pendigits.pca, comp=i:ncp)
# calculo del error de reconstruccion asociado
error_reconstruccion <- pendigits.reconst - pendigits$X
error_reconstruccion <- error_reconstruccion^2
error_reconstruccion <- apply(X = error_reconstruccion,
MARGIN = 1, FUN = mean)
# evaluacion del error_reconstruccion como 'score' de anomalia
pca.roc<-roc(as.factor(pendigits$y),
as.vector(error_reconstruccion), direction='<')
print(auc(pca.roc))
print(coords(pca.roc, x="best"))
}
```

De la misma forma que anteriormente se procede con las últimas componentes principales (utilizando desde la última componente hasta llegar a utilizar todas las componentes en orden regresivo). Obteniendo así las AUC, sensitivity y specificity de cada combinación.

```
for (i in ncp:1)
{
  # calculo de la reconstruccion con i componentes
print(c("numero de componentes", i))
pendigits.reconst<-reconstruct_prcomp(pendigits.pca, comp=i:ncp)
  # calculo del error de reconstruccion asociado
error_reconstruccion <- pendigits.reconst - pendigits$X
error_reconstruccion <- error_reconstruccion^2
error_reconstruccion <- apply(X = error_reconstruccion,
MARGIN = 1, FUN = mean)
  # evaluacion del error_reconstruccion como 'score' de anomalia
pca.roc<-roc(as.factor(pendigits$y),
as.vector(error_reconstruccion), direction='<')
print(auc(pca.roc))
print(coords(pca.roc, x="best"))
```

Se aplica ahora la mejor reconstrucción obtenida, con las últimas 5 componentes principales.

```
pendigits.reconst<-reconstruct_prcomp(pendigits.pca, comp=12:16)
```

Se calcula el error de reconstrucción obtenido y se enseña gráficamente la diferencia entre las densidades de los datos clasificados como atípicos y como normales.

```
error_reconstruccion <- pendigits.reconst - pendigits$X
error_reconstruccion <- error_reconstruccion^2
error_reconstruccion <- apply(X = error_reconstruccion,
MARGIN = 1, FUN = mean)

plot(density(error_reconstruccion[pendigits$y==0]),col = 'blue',
main = ' ', xlab = ' ', ylab = "Density")
lines(density(error_reconstruccion[pendigits$y==1]), col= 'red')
```

Se obtiene la matriz de confusión respectiva al modelo.

```
xtabs(~(error_reconstruccion>=0.1117753)+pendigits$y)
```

Se construye el Isolation Forest utilizando la libreria isotree y aplicando su función isolation.tree con los parámetros ya optimizados: $sample_size = 500$, $ntrees = 250$. Una vez construido se aplica al conjunto Pendigits.

```
model.if<-isolation.forest(pendigits$X,sample_size=500, ndim=1,
ntrees=250, nthreads=1)
model.if.avgd<-predict(model.if, pendigits$X, type="avg_depth")
```

Se realiza un gráfico de comparación de las funciones de densidad de las longitudes de ruta respectivas a los datos clasificados como atípicos y no atípicos.

```
plot(density(model.if.avgd[pendigits$y==0]),ylim = c(0, 0.7),
col = 'blue', main = ' ', xlab = ' ', ylab = "Density")
lines(density(model.if.avgd[pendigits$y==1]), col = 'red')
```

Se comprueba la calidad de la curva *ROC* tanto gráficamente como numéricamente.

```
if.roc<-roc(as.factor(pendigits$y),as.vector(model.if.avgd),
direction='>')

plot(roc(as.factor(pendigits$y),as.vector(model.if.avgd),
direction='>'))
```

```
auc(roc(as.factor(pendigits$y),as.vector(model.if.avgd),
direction='>'))

coords(if.roc, x="best")
```

Por último, se realiza la matriz de confusión del modelo, comparando los datos clasificados con los previamente etiquetados.

```
xtabs(~(model.if.avgd<=10.41695)+pendigits$y)
```