

Detección de anomalías basada en técnicas de ensembles

Ignacio Aguilera Martos

7 de septiembre de 2019

Trabajo Fin de Grado

Código disponible en [GitHub](#)

1. Concepto de anomalía basado en distancias
2. Aprendizaje Automático
3. Probabilidad Multivariante
4. Concepto probabilístico de anomalía
5. Modelos implementados

Concepto de anomalía basado en distancias

Tukey's Fences

Tukey's Fences

Pensado para el caso uno-dimensional.

Tukey's Fences

Pensado para el caso uno-dimensional.

Tukey's Fences

Valores fuera del rango $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$ con $k = 1,5$

Tukey's Fences

Pensado para el caso uno-dimensional.

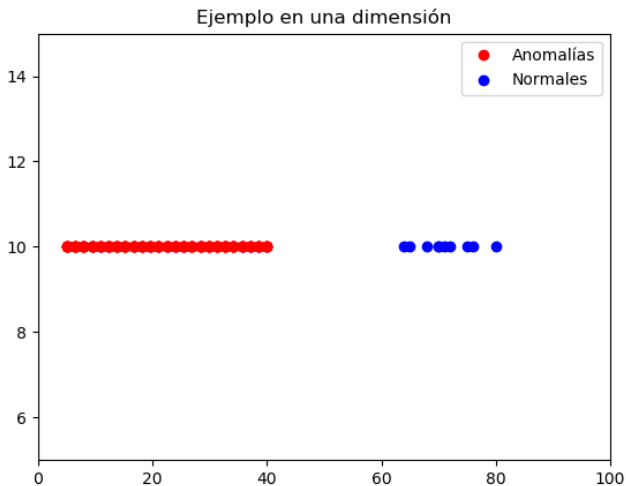
Tukey's Fences

Valores fuera del rango $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$ con $k = 1,5$

La propuesta de $k = 1,5$ es arbitraria.

Tukey's Fences

Tukey's Fences



Extensión al caso de mayor dimensionalidad

Extensión al caso de mayor dimensionalidad

Criterio

Aplicar el criterio de Tukey a cada una de las características.

Extensión al caso de mayor dimensionalidad

Criterio

Aplicar el criterio de Tukey a cada una de las características. **Trivial**.

Extensión al caso de mayor dimensionalidad

Criterio

Aplicar el criterio de Tukey a cada una de las características. **Trivial**.

Criterio de clusters

1. Agrupamos los datos por clusters.

Extensión al caso de mayor dimensionalidad

Criterio

Aplicar el criterio de Tukey a cada una de las características. **Trivial**.

Criterio de clusters

1. Agrupamos los datos por clusters.
2. Encontramos el cluster más cercano para cada instancia.

Extensión al caso de mayor dimensionalidad

Criterio

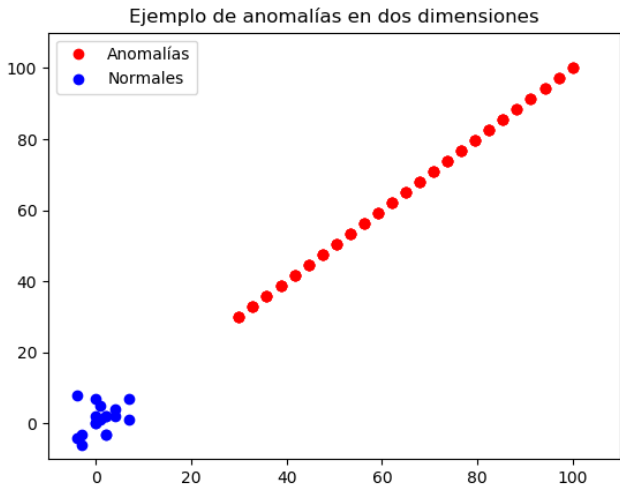
Aplicar el criterio de Tukey a cada una de las características. **Trivial**.

Criterio de clusters

1. Agrupamos los datos por clusters.
2. Encontramos el cluster más cercano para cada instancia.
3. Si la distancia del objeto al centroide del cluster es mayor que 1,5 veces la mayor distancia intercluster entonces es una anomalía.

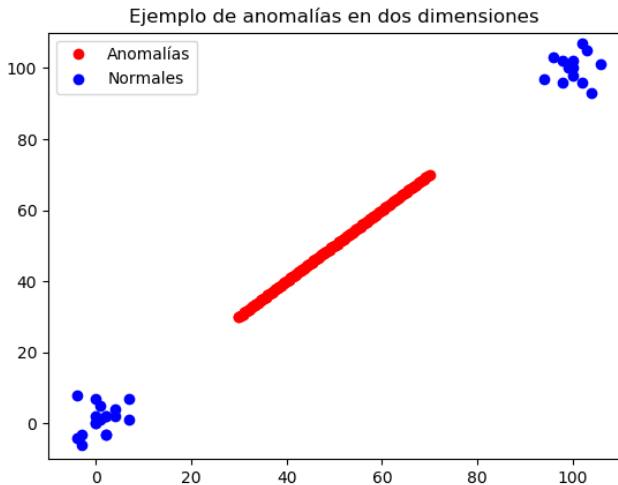
Ejemplo 1

Ejemplo 1



Ejemplo 2

Ejemplo 2



Aprendizaje Automático

Problema que abordamos

Problema que abordamos

Problema de detección de anomalías

Es un problema de aprendizaje no supervisado pues no disponemos de las etiquetas.

Problema que abordamos

Problema de detección de anomalías

Es un problema de aprendizaje no supervisado pues no disponemos de las etiquetas.

Partes teóricas del problema

1. Generador.
2. Sistema.
3. Máquina de aprendizaje.

Propiedades de conjuntos de alta dimensionalidad

1. La densidad disminuye exponencialmente al aumentar la dimensionalidad.

Propiedades de conjuntos de alta dimensionalidad

1. La densidad disminuye exponencialmente al aumentar la dimensionalidad.
2. Cuanto mayor es la dimensionalidad mayor debe ser el radio de una bola para englobar el mismo porcentaje de datos.

Propiedades de conjuntos de alta dimensionalidad

1. La densidad disminuye exponencialmente al aumentar la dimensionalidad.
2. Cuanto mayor es la dimensionalidad mayor debe ser el radio de una bola para englobar el mismo porcentaje de datos.
3. Casi todo punto está más cerca del borde del conjunto que de otro punto.

$$D(d, n) = \left(1 - \frac{1}{2^{\frac{1}{n}}}\right)^{\frac{1}{d}}$$

Propiedades de conjuntos de alta dimensionalidad

1. La densidad disminuye exponencialmente al aumentar la dimensionalidad.
2. Cuanto mayor es la dimensionalidad mayor debe ser el radio de una bola para englobar el mismo porcentaje de datos.
3. Casi todo punto está más cerca del borde del conjunto que de otro punto.

$$D(d, n) = \left(1 - \frac{1}{2^{\frac{1}{n}}}\right)^{\frac{1}{d}}$$

4. Casi todo punto es una anomalía sobre su propia proyección.

Propiedades de conjuntos de alta dimensionalidad

1. La densidad disminuye exponencialmente al aumentar la dimensionalidad.
2. Cuanto mayor es la dimensionalidad mayor debe ser el radio de una bola para englobar el mismo porcentaje de datos.
3. Casi todo punto está más cerca del borde del conjunto que de otro punto.

$$D(d, n) = \left(1 - \frac{1}{2^{\frac{1}{n}}}\right)^{\frac{1}{d}}$$

4. Casi todo punto es una anomalía sobre su propia proyección.

Maldición de la alta dimensionalidad

A mayor dimensionalidad mayor número de puntos necesitamos para obtener una aproximación con funciones de igual regularidad.

Aproximación de funciones

Nuestro objetivo es aproximar la función de salida del sistema con los datos que tenemos.

Aproximación de funciones

Aproximación de funciones

Nuestro objetivo es aproximar la función de salida del sistema con los datos que tenemos.

Teoremas útiles

- Teorema de Aproximación de Weierstrass

Aproximación de funciones

Aproximación de funciones

Nuestro objetivo es aproximar la función de salida del sistema con los datos que tenemos.

Teoremas útiles

- Teorema de Aproximación de Weierstrass
- Serie de Fourier

Aproximación de funciones

Aproximación de funciones

Nuestro objetivo es aproximar la función de salida del sistema con los datos que tenemos.

Teoremas útiles

- Teorema de Aproximación de Weierstrass
- Serie de Fourier
- Teorema de Kolmogorov-Arnold

Teorema de Aproximación de Weierstrass

Teorema de Aproximación de Weierstrass

Teorema de Aproximación de Weierstrass

Supongamos que tenemos una función $f : [a, b] \rightarrow \mathbb{R}$ continua.

Entonces $\forall \epsilon > 0$ existe un polinomio p tal que $\forall x \in [a, b]$ tenemos que $|f(x) - p(x)| < \epsilon$.

Teorema de Aproximación de Weierstrass

Teorema de Aproximación de Weierstrass

Supongamos que tenemos una función $f : [a, b] \rightarrow \mathbb{R}$ continua.

Entonces $\forall \epsilon > 0$ existe un polinomio p tal que $\forall x \in [a, b]$ tenemos que $|f(x) - p(x)| < \epsilon$.

Aproximación por polinomios

El Teorema de Aproximación de Weierstrass nos da una forma de aproximar funciones por polinomios.

Serie de Fourier

Si tenemos una función $f : \mathbb{R} \rightarrow \mathbb{R}$ integrable en el intervalo $[t_0 - \frac{T}{2}, t_0 + \frac{T}{2}]$ entonces se puede obtener el desarrollo de Fourier de f en dicho intervalo. Si f es periódica en toda la recta real la aproximación es válida en todos los valores en los que esté definida.

$$f(t) \approx \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{2n\pi}{T} t\right) + b_n \sin\left(\frac{2n\pi}{T} t\right) \right]$$

$$a_0 = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) dt, \quad a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cos\left(\frac{2n\pi}{T} t\right) dt,$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \sin\left(\frac{2n\pi}{T} t\right) dt$$

Teorema de Kolmogorov-Arnold

Teorema de Kolmogorov-Arnold

Teorema de Superposición Kolmogorov-Arnold

Sea f una función continua de varias variables $f : X_1 \times \dots \times X_n \rightarrow \mathbb{R}$, entonces existen funciones $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$ y $\phi_{q,p} : X_p \rightarrow [0, 1]$ tales que f se puede expresar como:

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$$

Teorema de Kolmogorov-Arnold

Teorema de Superposición Kolmogorov-Arnold

Sea f una función continua de varias variables $f : X_1 \times \dots \times X_n \rightarrow \mathbb{R}$, entonces existen funciones $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$ y $\phi_{q,p} : X_p \rightarrow [0, 1]$ tales que f se puede expresar como:

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$$

Dimensionalidad en datos y funciones

La maldición de la dimensionalidad es intrínseca a los datos. Este Teorema nos dice que la capacidad expresiva o complejidad de las funciones de una sola variable es la misma que las de varias variables.

Notación

$$MSE = \frac{1}{n} \sum_{i=1}^n y_i - g(X_i, \mathcal{D})^2$$

$$E[MSE] = \frac{1}{n} \sum_{i=1}^n E[y_i - g(X_i, \mathcal{D})^2] = \dots$$

$$= \frac{1}{n} \sum_{i=1}^n \{f(X_i) - E[g(X_i, \mathcal{D})]\}^2 + \frac{1}{n} \sum_{i=1}^n E[\{E[g(X_i, \mathcal{D})] - g(X_i, \mathcal{D})\}^2]$$

$$= \text{sesgo}^2 + \text{varianza}$$

Teorema clave de la Teoría del Aprendizaje

Para funciones de pérdida acotadas el principio inductivo de minimización del error empírico es consistente sí y sólo si el error empírico converge uniformemente al valor real del error en el siguiente sentido:

$$\lim_{n \rightarrow \infty} P[\sup_{\omega} |R(\omega) - R_{emp}(\omega)| > \epsilon] = 0, \quad \forall \epsilon > 0$$

Dimensión VC

Decimos que un conjunto de funciones tiene dimensión VC h si puede resolver de forma óptima todos los casos de tamaño h pero existe al menos uno de tamaño $h + 1$ que no puede resolver.

Dimensión VC

Decimos que un conjunto de funciones tiene dimensión VC h si puede resolver de forma óptima todos los casos de tamaño h pero existe al menos uno de tamaño $h + 1$ que no puede resolver.

Cota ERM

Con probabilidad $1 - \eta$

$$R(\omega) \leq R_{emp}(\omega) + \frac{\epsilon}{2} \left(1 + \sqrt{1 + \frac{4 \cdot R_{emp}(\omega)}{\epsilon}} \right)$$

Dimensión VC

Decimos que un conjunto de funciones tiene dimensión VC h si puede resolver de forma óptima todos los casos de tamaño h pero existe al menos uno de tamaño $h + 1$ que no puede resolver.

Cota ERM

Con probabilidad $1 - \eta$

$$R(\omega) \leq R_{emp}(\omega) + \frac{\epsilon}{2} \left(1 + \sqrt{1 + \frac{4 \cdot R_{emp}(\omega)}{\epsilon}} \right)$$

$$\epsilon = a_1 \cdot \frac{h(\ln(\frac{a_2 n}{h}) + 1) - \ln(\frac{\eta}{4})}{n}$$

Probabilidad Multivariante

Contenidos de la sección

Contenidos útiles

- Vectores aleatorios.

Contenidos útiles

- Vectores aleatorios.
- Independencia.

Contenidos útiles

- Vectores aleatorios.
- Independencia.
- Probabilidad condicionada (sucesos, variables y σ -álgebras) y sus propiedades.

Contenidos útiles

- Vectores aleatorios.
- Independencia.
- Probabilidad condicionada (sucesos, variables y σ -álgebras) y sus propiedades.
- Esperanza condicionada (sucesos, variables y σ -álgebras) y sus propiedades.

Contenidos útiles

- Vectores aleatorios.
- Independencia.
- Probabilidad condicionada (sucesos, variables y σ -álgebras) y sus propiedades.
- Esperanza condicionada (sucesos, variables y σ -álgebras) y sus propiedades.
- Desigualdades famosas.

Desigualdad de Markov y Chebychev

Desigualdad de Markov y Chebychev

Desigualdad de Markov

Sea X una variable aleatoria que toma valores no negativos. Entonces para cualquier constante α satisfaciendo $E[X] < \alpha$ se cumple que:

$$P(X > \alpha) \leq \frac{E[X]}{\alpha}$$

Desigualdad de Chebychev

Sea X una variable aleatoria arbitraria. Entonces para cualquier constante α se tiene que:

$$P(|X - E[X]| > \alpha) \leq \frac{\text{Var}[X]}{\alpha^2 d}$$

Desigualdades estudiadas

- Desigualdad de Markov
- Desigualdad de Chebychev
- Cotas de Chernoff
- Desigualdad de Hoeffding

Concepto probabilístico de anomalía

Notación usada

$$X = \{x_1, \dots, x_n\}, \quad x_i = (x_{s_1}, \dots, x_{s_d})$$

Notación usada

$$X = \{x_1, \dots, x_n\}, \quad x_i = (x_{s_1}, \dots, x_{s_d})$$

$$S = \{s_i | s_i \in \{s_1, \dots, s_d\} \text{ con } i \in \Delta\}$$

Notación usada

$$X = \{x_1, \dots, x_n\}, \quad x_i = (x_{s_1}, \dots, x_{s_d})$$

$$S = \{s_i | s_i \in \{s_1, \dots, s_d\} \text{ con } i \in \Delta\}$$

X_S *proyección de los datos en el subespacio S*

Notación usada

$$X = \{x_1, \dots, x_n\}, \quad x_i = (x_{s_1}, \dots, x_{s_d})$$

$$S = \{s_i | s_i \in \{s_1, \dots, s_d\} \text{ con } i \in \Delta\}$$

X_S *proyección de los datos en el subespacio S*

$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p})$$

Notación usada

$$X = \{x_1, \dots, x_n\}, \quad x_i = (x_{s_1}, \dots, x_{s_d})$$

$$S = \{s_i | s_i \in \{s_1, \dots, s_d\} \text{ con } i \in \Delta\}$$

X_S *proyección de los datos en el subespacio S*

$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p})$$

$$p_{s_i}(x_{s_i})$$

Definición subespacio incorrelado

Decimos que un subespacio S es un subespacio incorrelado si y sólo si:

$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p}) = \prod_{i=1}^p p_{s_i}(x_{s_i})$$

Definición subespacio incorrelado

Decimos que un subespacio S es un subespacio incorrelado si y sólo si:

$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p}) = \prod_{i=1}^p p_{s_i}(x_{s_i})$$

Definición anomalía no trivial

Decimos que un objeto x_S es una anomalía no trivial respecto del subespacio S si:

$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p}) \ll p_{esp}(x_{s_1}, \dots, x_{s_p})$$

Definición subespacio incorrelado

Decimos que un subespacio S es un subespacio incorrelado si y sólo si:

$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p}) = \prod_{i=1}^p p_{s_i}(x_{s_i})$$

Definición anomalía no trivial

Decimos que un objeto x_S es una anomalía no trivial respecto del subespacio S si:

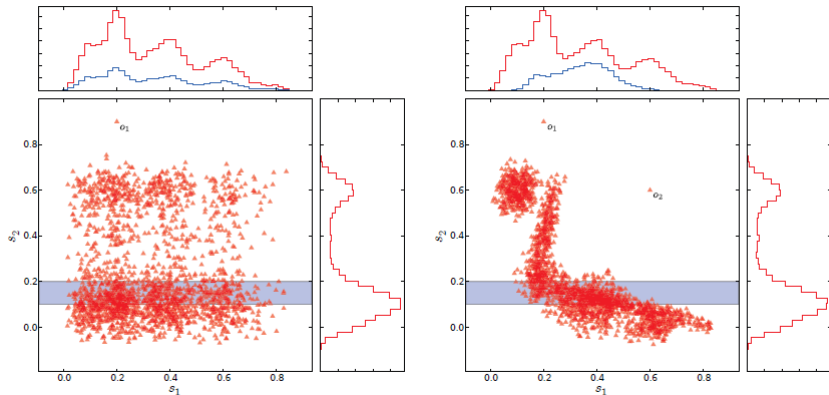
$$p_{s_1, \dots, s_p}(x_{s_1}, \dots, x_{s_p}) \ll p_{esp}(x_{s_1}, \dots, x_{s_p})$$

Relación entre conceptos de anomalía

Este concepto de anomalía es complementario.

Ejemplo de anomalía

Ejemplo de anomalía



Modelos implementados

Algoritmos de ensamblaje

- Algoritmos secuenciales

Algoritmos de ensamblaje

- Algoritmos secuenciales
- Algoritmos independientes

Algoritmos de ensamblaje

- Algoritmos secuenciales
- Algoritmos independientes

Ensamblaje secuencial:

Entrada: Conjunto de datos \mathcal{D} , Algoritmos base $\mathcal{A}_1, \dots, \mathcal{A}_r$

j=1

repetir

Tomamos el algoritmo \mathcal{A}_j según los resultados anteriores

Tomamos el conjunto de datos modificado $f_j(\mathcal{D})$ de anteriores ejecuciones

Ejecutamos el algoritmo \mathcal{A}_j sobre $f_j(\mathcal{D})$

j=j+1

hasta que *fin*;

Resultado: Combinación de los resultados

Algoritmos de ensamblaje

- Algoritmos secuenciales
- Algoritmos independientes

Ensamblaje independiente:

Entrada: Conjunto de datos \mathcal{D} , Algoritmos base $\mathcal{A}_1, \dots, \mathcal{A}_r$

j=1

repetir

 Tomamos el algoritmo \mathcal{A}_j

 Creamos el conjunto de datos modificado $f_j(\mathcal{D})$

 Ejecutamos el algoritmo \mathcal{A}_j sobre $f_j(\mathcal{D})$

 j=j+1

hasta que *fin*;

Resultado: Combinación de los resultados

HICS

Entrada: D: dataset

scores = []

sub = subespacios de alto contraste

para cada $S \in sub$ **hacer**

 Ajustamos un modelo con el algoritmo LOF con la proyección
 sobre S

scores = *scores* + *puntaje LOF*

fin

scores = $\frac{scores}{|sub|}$

Salida: *scores*: puntajes

CalcularContraste

Entrada: subespacio: subespacio, M : número de iteraciones del subsampling, α : valor para obtener el tamaño de la muestra, D : conjunto de datos

$$size = n \cdot \sqrt[|subespacio|]{\alpha}$$

$$dev = 0$$

para cada $i \in [1, M]$ **hacer**

$comp_atr = \text{aleatorio de subespacio}$

$sel_obj = \text{muestra aleatoria de } D \text{ de tamaño } size$

$dev = dev + \text{CalcularDev}(comp_atr, sel_obj, subespacio, D)$

fin

$$dev = \frac{dev}{M}$$

Salida: dev : contraste

CalcularDev

Entrada: *comp_atr*: atributo con el que comparar, *sel_obj*: muestra seleccionada aleatoriamente, *subespacio*: subespacio sobre el que calcular la desviación, *D*: conjunto de datos

max = 0

para cada $d \in D$ **hacer**

$cum_1 = \sum_{o \in D} o[comp_atr]$ si $o[comp_atr] < d[comp_atr]$

$cum_2 = \sum_{o \in sel_obj} o[comp_atr]$ si $o[comp_atr] < d[comp_atr]$

$f_a = \frac{cum_1}{|D|}$

$f_b = \frac{cum_2}{|D|}$

$subs = |f_a - f_b|$

si $subs > max$ **entonces**

$max = subs$

fin

fin

Salida: *max*: máxima desviación

OUTRES

Entrada: o : instancia, S : subespacio

para cada $i \in (D \setminus S)$ **hacer**

$S' = S \cup \{i\}$

si S' *es relevante* **entonces**

$$\text{den}(o, S') = \frac{1}{n} \sum_{p \in AN(o, S')} K_e\left(\frac{\text{dist}_{S'}(o, p)}{\epsilon(|S'|)}\right)$$

$$\text{dev}(o, S') = \frac{\mu - \text{den}(o, S')}{2\sigma}$$

si $\text{dev}(o, S') \geq 1$ **entonces**

$$\quad \quad r(o) = r(o) \cdot \frac{\text{den}(o, S')}{\text{dev}(o, S')}$$

fin

$OUTRES(o, S')$

fin

en otro caso

 Para recursividad

fin

fin

Salida: r : puntajes

Subespacio relevante

Decimos que un subespacio es relevante si no está distribuido uniformemente.

Subespacio relevante

Decimos que un subespacio es relevante si no está distribuido uniformemente.

Procedimiento

Si la proyección de los datos sobre un subespacio está distribuida según una uniforme entonces sus proyecciones en una dimensión también lo están.

Elementos involucrados

$$AN(o, S) = \{p | dist_S(o, p) \leq \epsilon(|S|)\}$$

Elementos involucrados

$$AN(o, S) = \{p | dist_S(o, p) \leq \epsilon(|S|)\}$$

$$\epsilon(|S|) = 0,5 \cdot \frac{h_{optimal}(|S|)}{h_{optimal}(2)}$$

Elementos involucrados

$$AN(o, S) = \{p | \text{dist}_S(o, p) \leq \epsilon(|S|)\}$$

$$\epsilon(|S|) = 0,5 \cdot \frac{h_{\text{optimal}}(|S|)}{h_{\text{optimal}}(2)}$$

$$h_{\text{optimal}}(d) = \left(\frac{8\Gamma(\frac{d}{2} + 1)}{\pi^{\frac{d}{2}}} \right) (d + 4)(2\sqrt{\pi})^d n^{\frac{-1}{d+4}}$$

Elementos involucrados

$$AN(o, S) = \{p | dist_S(o, p) \leq \epsilon(|S|)\}$$

$$\epsilon(|S|) = 0,5 \cdot \frac{h_{optimal}(|S|)}{h_{optimal}(2)}$$

$$h_{optimal}(d) = \left(\frac{8\Gamma(\frac{d}{2} + 1)}{\pi^{\frac{d}{2}}} \right) (d + 4)(2\sqrt{\pi})^d n^{\frac{-1}{d+4}}$$

$$K_{\epsilon}(x) = (1 - x^2) \quad \forall x < 1$$

Proyecciones Aleatorias

Entrada: d : dimension, D : dataset, k : número de histogramas y proyecciones

$no_neg = \lceil \sqrt{d} \rceil$

$proyecciones = []$

para cada $i \in [1, k]$ **hacer**

$ind = no_neg$ índices aleatorios en $[0, d]$

$proy =$ vector con ceros en todas las posiciones menos en ind ,
 donde hay valores sacados de una normal $\mathcal{N}(0, 1)$

$proyecciones = [proyecciones, proy]$

fin

Salida: $proyecciones$: proyecciones

ObtenerHistogramas

Entrada: D : dataset, $\{w_i\}_{i=1}^k$: vectores de proyecciones, k : numero de histogramas y vectores de proyección

Inicializamos los histogramas $\{h_i\}_{i=1}^k$

para $j = 1 \rightarrow |D|$ **hacer**

para $i = 1 \rightarrow k$ **hacer**

$z_i = x_j^T w_i$

 Actualizar el histograma h_i con z_i

fin

fin

Salida: $\{h_i\}_{i=1}^k$: histogramas

LODA

Entrada: x : instancia, $\{h_i\}_{i=1}^k$: histogramas, $\{w_i\}_{i=1}^k$: vectores de proyección

para $i = 1 \rightarrow k$ **hacer**

$$z_i = x^T w_i$$

Obtenemos $p_i = p_i(z_i)$ del histograma h_i

fin

$$f = \frac{-1}{k} \sum_{i=1}^k \log(p_i(z_i))$$

Salida: f : puntaje de anomalía de x

Mahalanobis Kernel

Mahalanobis Kernel Entrada: D

$$S = DD^T.$$

$$S = Q\Delta^2Q^T.$$

Almacenamos los vectores propios columna no negativos de $Q\Delta$ en una matriz D'

Normalizamos D' para que tenga media 0 y varianza 1.

vector_media = *media*(D')

puntuaciones = []

para cada fila en D' hacer

score = *distancia*(*vector_media*, *fila*)

puntuaciones = [*puntuaciones*, *score*]

fin

Salida: *puntuaciones*

Componentes

1. Componente basado en distancias: KNN con $k = 5$

Componentes

1. Componente basado en distancias: KNN con $k = 5$
2. Componente basado en dependencia: Mahalanobis Kernel

Componentes

1. Componente basado en distancias: KNN con $k = 5$
2. Componente basado en dependencia: Mahalanobis Kernel
3. Componente basado en densidad en subespacios: IForest

A

plicamos en todos los componentes la técnica de subsampling.

Trinity Entrada: D

Ejecutamos KNN con $k = 5$ y guardamos el resultado en E_1

Ejecutamos Mahalanobis Kernel y guardamos el resultado en E_2

Ejecutamos IForest y guardamos el resultado en E_3

Estandarizamos E_1 , E_2 y E_3 a media 0 y varianza 1

Hacemos la media para obtener las puntuaciones

Salida: puntuaciones

Recursos usados

- Implementación hecha en Python3

Recursos usados

- Implementación hecha en Python3
- Disponible en GitHub

Recursos usados

- Implementación hecha en Python3
- Disponible en GitHub
- Documentada con Sphinx

¿Preguntas?