**Instituto Politécnico Nacional**

**Centro de Investigación en Computación**

**Maestría en Ciencias de la Computación**

****

D

C

B

A

**Tarea 04: Elementos básicos de un clasificador inteligente de patrones (cip)**

**Propósito:** Proponer una partición arbitraria y desarrollar los cuatro elementos básicos de un cip sencillo: el Clasificador Euclidiano. Verificar que en ciertos datasets relevantes del estado del arte se utiliza la partición fija como un método de validación auténtico. Realizar un intento por aplicar el Clasificador Euclidiano en el MNIST dataset (conocido dataset con 10 clases).

**Materia:** Clasificación Inteligente de Patrones

**Alumno:** Andrés Burjand Torres Reyes

**Profesor:** Dr. Cornelio Yáñez Márquez

**Fecha de entrega:** 28 de septiembre de 2024

**Índice**

**Índice General**

Introducción……………………………………………………………………………………2

Desarrollo y Discusión………………………………………………………………………..3

Conclusiones…………………………………………………………………………………..10

Referencias…………………………………………………………………………………….9

**Introducción**

Una vez comprendida la posición del Reconocimiento de Patrones como disciplina integrada en la Inteligencia Artificial (“Vista como computación inteligente”), y después de analizar la Clasificación Inteligente de Patrones desde un punto de vista general, es momento de entender de manera más particular qué es un clasificador inteligente de patrones (cip).

Un cip es un algoritmo que consiste en utilizar programas informáticos para clasificar datos de entrada en clases o categorías, con base en sus características principales o elementos constantes [1]. Todo clasificador inteligente de patrones consta de cuatro partes fundamentales: Un dataset, que es el conjunto de datos que se utilizarán para el entrenamiento y la comprobación de eficacia del clasificador; un método de validación, que es la elección (mediante distintas técnicas) de los patrones que se utilizarán para entrenamiento y los que se usarán para prueba; un algoritmo de clasificación, que es la esencia del funcionamiento del cip; y, finalmente, una medida de evaluación de desempeño, que permite determinar qué tan efectivo es el algoritmo.

Para este primer acercamiento utilizaremos el algoritmo euclidiano, que, al ser simple, permite centrarse en empezar a entender el funcionamiento del clasificador. El algoritmo euclidiano es una de las formas de clasificadores por funciones de distancia, toma su nombre de uno de los matemáticos más importantes de todos los tiempos. Consiste en que los patrones sean analizados como puntos en un plano de n dimensiones (Siendo n la cantidad de atributos del patrón), luego se calcula el centroide de cada clase en los patrones de entrenamiento para, finalmente, clasificar cada patrón de prueba teniendo en cuenta con cuál centroide su distancia euclidiana es menor.

**Desarrollo y Discusión**

**PARTE 1.**

1. En el dataset D del año 65 ya estudiado, proponer una partición fija, de modo que el conjunto de prueba se forme de 4 patrones de la clase 1 y 4 patrones de la clase 2 (el conjunto P tendrá 8 patrones en total).

**Conjunto E:**

{

[30 0 1],

[31 4 1],

[40 0 1],

[41 0 1],

[42 0 1],

[43 0 1],

[50 4 1],

[51 0 1],

[52 0 1],

[61 8 1]

[45 6 2],

[46 20 2],

[47 0 2],

[53 1 2],

[53 12 2],

[54 23 2],

[54 5 2],

[56 9 2],

[60 0 2]

}

**Conjunto P:**

{

[64 22 1],

[67 0 1],

[69 0 1],

[77 3 1]

[61 0 2],

[62 19 2],

[74 3 2],

[78 1 2]

}

1. Reportar el desarrollo completo del Clasificador Euclidiano.
2. Diseño del clasificador
   1. Dataset

UCI Haberman’s Survival Data Set modificado para operar solo con los valores del año 1965 como single-label dataset.

* 1. Método de validación

Partición fija. 19 patrones para entrenamiento y 8 para prueba.

* 1. Algoritmo

**Euclidiano**. Se calcula el centroide por cada clase de los patrones de entrenamiento y se clasifican los patrones de prueba según su cercanía a los centroides.

* 1. Medida para evaluar el desempeño

Accuracy.

1. Código

Clase principal

A screenshot of a computer

Description automatically generated

Clase del método de validación

A screenshot of a computer program

Description automatically generated

Clase del clasificador Euclidiano

A screenshot of a computer screen

Description automatically generated

A black screen with many colorful lines

Description automatically generated with medium confidence

Clase de la medida de desempeño

A screenshot of a computer

Description automatically generated

Clase con herramientas matemáticas

A screenshot of a computer

Description automatically generated

1. Reportar el valor de accuracy.

Accuracy = 0.5

**PARTE 2.**

1. Verificar que el famoso MNIST dataset tiene una partición fija.

Tiene partición fija.

1. ¿Cuántos patrones contiene el MNIST dataset?

Contiene en total 70,000 patrones.

1. ¿Cuántos atributos contienen los patrones del MNIST dataset?

Contiene 784 atributos.

1. ¿Cuántas clases hay en el MNIST dataset?

10 clases, del 0 al 9.

1. ¿De qué tipo son los atributos del MNIST dataset?

Son de tipo numérico – enteros.

1. ¿Hay valores perdidos en el MNIST dataset?

No tiene valores perdidos.

1. ¿Cuántos patrones contiene el conjunto E?

El conjunto de entrenamiento consta de 60,000 patrones.

1. ¿Cuántos patrones contiene el conjunto P?

El conjunto de prueba consta de 10,000 patrones.

1. ¿Cuál es la cardinalidad de la clase mayoritaria del conjunto P?

La clase mayoritaria del conjunto P es la etiqueta 1 y contiene 1135 patrones.

1. ¿Cuál es la cardinalidad de la clase minoritaria del conjunto P?

La clase minoritaria del conjunto P es la etiqueta 4 y contiene 892 patrones.

1. ¿Cuál es el valor de IR del conjunto P?
2. ¿Por qué razón es posible aplicar accuracy en el conjunto P?

Porque está balanceado, pues su .

**PARTE 3**

1. Realizar la fase de aprendizaje del Clasificador Euclidiano al aplicarlo en el MNIST dataset.
2. Realizar la fase de prueba del Clasificador Euclidiano en el MNIST dataset.
3. Reportar el valor de accuracy.

Accuracy = 0.0962

**Conclusiones**

La comparación entre los índices de los libros sobre Pattern Recognition, Machine Learning y Data Mining revela un panorama enriquecedor sobre las diferentes perspectivas y enfoques en el campo del aprendizaje automático y el análisis de datos. Los libros seleccionados muestran una diversidad de temas y técnicas que abarcan desde fundamentos teóricos hasta aplicaciones prácticas, cada uno con su propio enfoque único sobre el reconocimiento de patrones, el aprendizaje automático y la minería de datos.

El libro sobre Pattern Recognition tiende a profundizar en la teoría y los métodos estadísticos, proporcionando una base sólida para entender cómo los algoritmos pueden aprender de los datos. Por otro lado, el texto sobre Machine Learning se centra más en la variedad de modelos y algoritmos aplicables a diferentes tipos de problemas. Finalmente, el libro sobre Data Mining se enfoca en la extracción de conocimiento a partir de grandes volúmenes de datos, lo cual es crucial para el manejo y análisis de datasets complejos.

Es válido concluir que estos textos comparten muchas similitudes, claramente las bases matemáticas son muy parecidas y tienden a diferir más en el campo de aplicación. Sin embargo, tener la perspectiva de los 3 podría ser crucial en la formación de un profesional como un Científico de Datos, pues aportan enfoques complementarios para sacar el máximo provecho a los conjuntos de datos y asistir en la resolución de problemas que los involucren.

**Referencias**

[1] Reconocimiento de patrones (Pattern Recognition). (s. f.). MATLAB & Simulink. https://la.mathworks.com/discovery/pattern-recognition.html