Ingeniería en Inteligencia Artificial,
Machine Learning
Sem: 2025-1, 5BM1, Práctica 6, Fecha: 28 de Octubre 2024

Laboratorio 6: Clasificadores de la Distancia Mínima y 1NN

Machine Learning

Grupo: 5BM1

Profesor: Andrés Floriano García

Integrantes:
Juan Manuel Alvarado Sandoval
Alexander Iain Crombie Esquinca
Herrera Saavedra Jorge Luis
Quiñones Mayorga Rodrigo

Contents

1 Introducción

En el presente laboratorio se abordarán dos clasificadores: el clasificador de la distancia mínima y el clasificador 1NN (1-Nearest Neighbor). Ambos métodos son fundamentales en el campo del aprendizaje automático para la clasificación de datos.

El clasificador de la distancia mínima se basa en el principio de que la clase de un punto de datos nuevo se determina asignando el punto a la clase más cercana en función de una métrica de distancia, como la distancia euclidiana. Este enfoque es intuitivo y puede ser efectivo en diversas situaciones, aunque su desempeño puede verse afectado por la presencia de ruido en los datos o por la elección inapropiada de la métrica de distancia.

Por otro lado, el clasificador 1NN es una variante del método de vecinos más cercanos (k-NN), donde se asigna un punto de datos a la clase del único vecino más cercano. Este enfoque es sencillo y directo, pero puede ser sensible a la variabilidad en los datos, especialmente si hay clases desbalanceadas.

En este laboratorio, se implementarán y validarán ambos clasificadores utilizando tres conjuntos de datos: Iris, Wine y Breast Cancer. Se aplicarán métodos de validación como Hold-Out 70/30 estratificado, 10-Fold Cross-Validation estratificado y Leave-One-Out para evaluar su rendimiento.

2 Clasificador de la Distancia Mínima

El clasificador de la distancia mínima asigna una instancia a la clase cuyo centroide (promedio de las características) está más cercano a ella. Este método puede expresarse matemáticamente de la siguiente manera:

$$\hat{y} = \arg\min_{c \in C} d(x, \mu_c)$$

donde $\hat{y}$ es la clase predicha, C es el conjunto de clases, x es la instancia a clasificar y μ_c es el centroide de la clase c. La función de distancia d comúnmente utilizada es la distancia euclidiana:

$$d(x, \mu_c) = \sqrt{\sum_{i=1}^{n} (x_i - \mu_{c,i})^2}$$

Este clasificador es efectivo en contextos donde las clases están bien separadas, pero puede fallar en situaciones con superposición significativa entre clases.

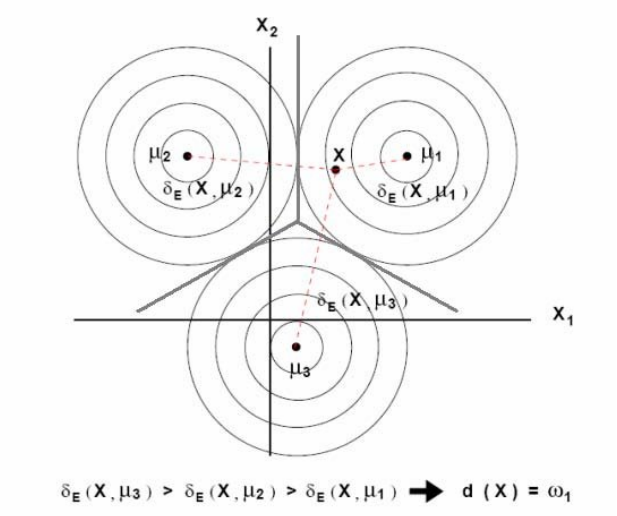

Figure 1: Clasificador de distancia mínima euclidiana.

3 Clasificador 1NN

El clasificador 1NN (uno de los vecinos más cercanos) se basa en la búsqueda del vecino más cercano de una instancia en el conjunto de entrenamiento. La clase de este vecino se asigna a la instancia a clasificar. Este método se puede formalizar de la siguiente manera:

$$\hat{y} = \arg\min_{x_j \in X} d(x, x_j)$$

donde $\hat{y}$ es la clase predicha, X es el conjunto de entrenamiento y d es la distancia euclidiana.

1. **Entrenamiento**: No se realiza un proceso de entrenamiento en el sentido tradicional, ya que el clasificador 1NN almacena todas las instancias del conjunto de entrenamiento.

2. **Clasificación**: Para clasificar una nueva instancia x:

   - Se calcula la distancia $d(x, x_j)$ entre x y cada instancia x_j en el conjunto de entrenamiento.
   - Se selecciona la instancia x_j que tiene la menor distancia a x.
   - La clase de x_j se asigna como la clase predicha para x.

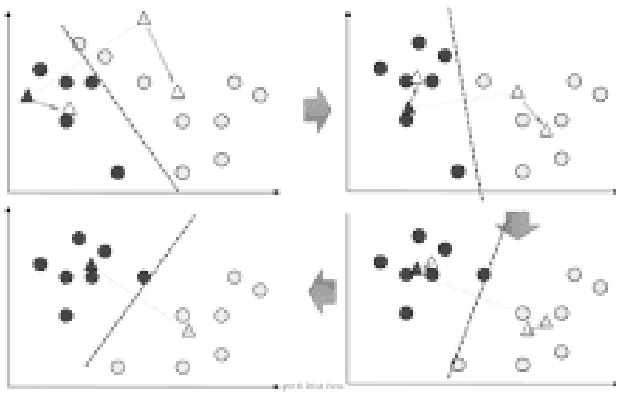

Figure 2: Clasificador K Nearest Neighbours.

4 Evidencia de la Práctica

Se presentarán los resultados obtenidos al aplicar ambos clasificadores en los conjuntos de datos mencionados, así como las métricas de desempeño como accuracy y matriz de confusión.

4.1 Resultados del Clasificador de la Distancia Mínima

```
--- Parte I: Clasificador de Distancia Mínima - Iris Dataset ---

Distancia Mínima - Hold-Out
Matriz de Confusión:
 [[15  0  0]
 [ 0 14  1]
 [ 0  3 12]]
Precisión: 0.9111111111111111

Distancia Mínima - K-Fold
Matriz de Confusión (Acumulada):
 [[50.  0.  0.]
 [ 0. 45.  5.]
 [ 0.  7. 43.]]
Precisión Promedio: 0.9200000000000002

Distancia Mínima - Leave-One-Out
Matriz de Confusión (Acumulada):
 [[50.  0.  0.]
 [ 0. 45.  5.]
 [ 0.  7. 43.]]
Precisión Promedio: 0.92
```

Figure 3: Resultados y métricas para Iris Dataset.

4.2 Resultados del Clasificador 1NN

```
--- Parte I: Clasificador de Distancia Mínima - Wine Dataset ---

Distancia Mínima - Hold-Out
Matriz de Confusión:
 [[15  0  3]
 [ 0 14  7]
 [ 0  5 10]]
Precisión: 0.7222222222222222

Distancia Mínima - K-Fold
Matriz de Confusión (Acumulada):
 [[50.  0.  9.]
 [ 3. 49. 19.]
 [ 1. 17. 30.]]
Precisión Promedio: 0.7245098039215687

Distancia Mínima - Leave-One-Out
Matriz de Confusión (Acumulada):
 [[50.  0.  9.]
 [ 3. 49. 19.]
 [ 1. 17. 30.]]
Precisión Promedio: 0.7247191011235955
```

Figure 4: Resultados y métricas para Wine Dataset.

```
--- Parte I: Clasificador de Distancia Mínima - Wine Dataset ---

Distancia Mínima - Hold-Out
Matriz de Confusión:
 [[15  0  3]
 [ 0 14  7]
 [ 0  5 10]]
Precisión: 0.7222222222222222

Distancia Mínima - K-Fold
Matriz de Confusión (Acumulada):
 [[50.  0.  9.]
 [ 3. 49. 19.]
 [ 1. 17. 30.]]
Precisión Promedio: 0.7245098039215687

Distancia Mínima - Leave-One-Out
Matriz de Confusión (Acumulada):
 [[50.  0.  9.]
 [ 3. 49. 19.]
 [ 1. 17. 30.]]
Precisión Promedio: 0.7247191011235955
```

Figure 5: Resultados y métricas para Breast Cancer Dataset.

```
--- Parte II: Clasificador 1NN - Iris Dataset ---

1NN - Hold-Out
Matriz de Confusión:
 [[15  0  0]
 [ 0 15  0]
 [ 0  3 12]]
Precisión: 0.9333333333333333

1NN - K-Fold
Matriz de Confusión (Acumulada):
 [[50.  0.  0.]
 [ 0. 47.  3.]
 [ 0.  3. 47.]]
Precisión Promedio: 0.9600000000000002

1NN - Leave-One-Out
Matriz de Confusión (Acumulada):
 [[50.  0.  0.]
 [ 0. 47.  3.]
 [ 0.  3. 47.]]
Precisión Promedio: 0.96
```

Figure 6: Resultados y métricas para Iris Dataset.

```
--- Parte II: Clasificador 1NN - Wine Dataset ---

1NN - Hold-Out
Matriz de Confusión:
 [[14  3  1]
 [ 1 15  5]
 [ 1  5  9]]
Precisión: 0.7037037037037037

1NN - K-Fold
Matriz de Confusión (Acumulada):
 [[50.  5.  4.]
 [ 5. 53. 13.]
 [ 3. 18. 27.]]
Precisión Promedio: 0.7300653594771241

1NN - Leave-One-Out
Matriz de Confusión (Acumulada):
 [[52.  3.  4.]
 [ 5. 54. 12.]
 [ 3. 14. 31.]]
Precisión Promedio: 0.7696629213483146
```

Figure 7: Resultados y métricas para Wine Dataset.

```
--- Parte II: Clasificador 1NN - Breast Cancer Dataset ---

1NN - Hold-Out
Matriz de Confusión:
 [[ 58   6]
 [  7 100]]
Precisión: 0.9239766081871345

1NN - K-Fold
Matriz de Confusión (Acumulada):
 [[185.  27.]
 [ 21. 336.]]
Precisión Promedio: 0.9156954887218044

1NN - Leave-One-Out
Matriz de Confusión (Acumulada):
 [[182.  30.]
 [ 18. 339.]]
Precisión Promedio: 0.9156414762741653
```

Figure 8: Resultados y métricas para Breast Cancer Dataset.

5 Conclusión

En este ejercicio, se exploraron dos métodos fundamentales de clasificación en el ámbito del aprendizaje automático: el clasificador de la distancia mínima y el clasificador 1NN. A través de la implementación de estos algoritmos en los conjuntos de datos Iris, Wine y Breast Cancer, se pudo observar su funcionamiento y rendimiento en la clasificación de datos.

El clasificador de la distancia mínima demostró ser efectivo en contextos donde las clases estaban bien separadas, permitiendo asignar instancias a la clase más cercana en función de la distancia euclidiana. Sin embargo, su sensibilidad a la variabilidad en los datos y la elección de la métrica de distancia subraya la importancia de una correcta preprocesamiento y selección de características.

Por otro lado, el clasificador 1NN, aunque simple y directo, mostró ser susceptible a la influencia de ruido y a la presencia de clases desbalanceadas. A pesar de su sencillez, este método resaltó la relevancia de la selección de vecinos y su impacto en la clasificación.

Los resultados obtenidos en ambas implementaciones resaltan la necesidad de un enfoque cuidadoso al elegir y aplicar algoritmos de clasificación, considerando las características específicas de los datos y los objetivos del análisis.

6 Enlace al repositorio

Haz click para seguir el enlace

7 Referencias

- Scikit-learn developers. (2024). *Scikit-learn: Machine Learning in Python*. Recuperado de: https://scikit-learn.org/stable/