INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Tarea 10:

Panorama histórico de la familia k-NN, empates, algoritmo k-NN y método de validación Leave-one-out.

Edgar Fernando Espinosa Torres

Clasificación inteligente de patrones

Dr. Cornelio Yáñez Márquez

Ciudad de México, 12 de noviembre de 2023.

Índice general

Índice de tablas

Introducción

La clasificación de patrones es una tarea fundamental del aprendizaje supervisado. Entre los algoritmos más sencillos y accesibles se encuentran el del Clasificador Euclidiano que fue tratado al inicio del curso y los de la familia de los k-Nearest Neighbors (k-NN). Aunque ambos se basan en medidas de distancia para clasificar, sus métodos difieren significativamente: el Clasificador Euclidiano utiliza un centroide para representar a cada clase basándose en la partición de entrenamiento, y clasifica los patrones de prueba según la cercanía a estos centroides. Por otro lado, el k-NN considera la proximidad de un patrón de prueba a los k patrones de entrenamiento más próximos (sus "k vecinos más cercanos").

Asimismo, hemos aprendido los métodos de validación de partición fija, hold-out estratificado y recientemente leave-one-out, que en esta tarea se emplea junto con k-NN de manera exhaustiva. Leave-one-out consiste en utilizar cada patrón del dataset como un conjunto de prueba individual, mientras que el resto de los datos constituyen el conjunto de entrenamiento. Este proceso se repite tantas veces como patrones hay en el dataset, garantizando que cada uno de ellos forme parte del conjunto de prueba en algún momento.

En esta tarea, nos centramos en aplicar el método de validación leave-one-out y los algoritmos del 7-NN y 3-NN a los datasets Nutt y Electricity respectivamente.

También, se calcularon sus matrices de confusión y a partir de ellas, se obtuvieron las medidas de desempeño $Accuracy$ y $Error\ Rate$ si los valores de $IR \leq 1.5$ y $Sensitivity$, $Specificity$ y $Balanced\ Accuracy$ válidas para cualquier valor de IR.

Desarrollo y discusión

Propósitos de la tarea

Ejemplificar algunos algoritmos de la familia k-NN con el método de validación Leave-one-out.

Parte 1

- **Aplicar el algoritmo 7-NN en el Nutt dataset con el método de validación Leave-one-out y generar la matriz de confusión.**
- **Reportar los valores de *accuracy*, *error rate*, *sensitivity*, *specificity* y *balanced accuracy*.**

El dataset Nutt se generó a partir del trabajo de Nutt en colaboración con otros investigadores que estudiaban el cáncer de cerebro. Como resultado de sus investigaciones, generaron un conjunto de 28 patrones de tejido cerebral, de los cuales 14 corresponden a glioblastomas clásicos (clase 0 - Positive), mientras que los 14 patrones restantes corresponden a glioblastomas no clásicos (clase 1 - Negative).

Cada patrón contiene 1070 atributos numéricos, los cuales representan los niveles de intensidad de expresión de los genes [1].

A continuación, se presentan los resultados del 7-NN con método de validación Leave-one-out.

Patrón de prueba (P)	Clase del patrón de prueba	Clase asignada por votación	Resultado
P1	0	1	FN
P2	0	1	FN
P3	0	1	FN
P4	0	1	FN
P5	0	1	FN
P6	0	0	TP
P7	0	0	TP
P8	0	0	TP
P9	0	1	FN
P10	0	1	FN
P11	0	1	FN
P12	0	1	FN
P13	0	1	FN

P14	0	1	FN
P15	1	1	TN
P16	1	1	TN
P17	1	1	TN
P18	1	1	TN
P19	1	1	TN
P20	1	1	TN
P21	1	1	TN
P22	1	1	TN
P23	1	1	TN
P24	1	1	TN
P25	1	1	TN
P26	1	1	TN
P27	1	1	TN
P28	1	1	TN

Tabla 1: Resultados del clasificador 7-NN con Leave-one-out aplicado al dataset Nutt.

A continuación, se presenta la matriz de confusión correspondiente al ejemplo:

3	11
0	14

Tabla 2: Matriz de confusión para Nutt.

El $imbalance\ ratio$ es $IR = \frac{14}{14} = 1.0 \leq 1.5$, es decir el dataset está balanceado. Por lo tanto, podemos utilizar las medidas de desempeño $accuracy$ y $error\ rate$.

$$accuracy = \frac{TP+TN}{TP+FN+FP+TN} = \frac{3+14}{3+11+0+14} = \frac{17}{28} = 0.6071 = 60.71\ \%$$

$$error\ rate = 1 - accuracy = 1 - 0.6071 = 0.3929 = 39.29\ \%$$

Para cualquier valor de IR es válido calcular las siguientes medidas de desempeño:

$$Sensitivity = \frac{TP}{TP + FN} = \frac{3}{3 + 11} = \frac{3}{14} = 0.2143 = 21.43\ \%$$

$$Specificity = \frac{TN}{FP + TN} = \frac{14}{14 + 0} = \frac{14}{14} = 1 = 100\%$$

En consecuencia, se puede obtener:

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} = \frac{0.2143 + 1}{2} = 0.6071 = 60.71\ \%$$

Parte 2

- **Aplicar el algoritmo 3-NN en el Electricity dataset con el método de validación Leave-one-out y generar la matriz de confusión.**
- **Reportar los valores de *accuracy*, *error rate*, *sensitivity*, *specificity* y *balanced accuracy*.**

El dataset Electricity consta de 2400 patrones, 8 atributos numéricos y es de tipo biclase. se recopiló del mercado eléctrico de Nueva Gales del Sur en Australia y contiene información relacionada con los precios de la electricidad y la demanda de electricidad en un periodo de tiempo de 30 minutos [2].

Atributos:

1. Fecha: La fecha de las observaciones, normalizada entre 0 y 1.
2. Día de la semana: Representa el día de la semana, con valores del 1 al 7.
3. Periodo: El momento de la medición en intervalos de media hora, normalizado entre 0 y 1 (a lo largo de un día).
4. Precio de electricidad en Nueva Gales del Sur (NSWprice): Normalizado entre 0 y 1.
5. Demanda de electricidad en Nueva Gales del Sur (NSWdemand): Normalizado entre 0 y 1.
6. Precio de electricidad en Victoria (VICprice): Normalizado entre 0 y 1.
7. Demanda de electricidad en Victoria (VICdemand): Normalizado entre 0 y 1.
8. Transferencia programada de electricidad entre ambos estados (transfer): Normalizado entre 0 y 1.

Vamos a considerar a la clase Positive cuando se tenga el valor de 1. Esto indica un cambio al alza en el precio de la electricidad en Nueva Gales del Sur en relación con un promedio

móvil de las últimas 24 horas. Por el contrario, la clase Negative será cuando se tenga el valor de 2, esto indica un cambio a la baja.

1247	183
214	756

Tabla 3: Matriz de confusión para Electricity.

El $imbalance\ ratio$ es $IR = \frac{1430}{970} = 1.4742 \leq 1.5$, es decir el dataset está balanceado. Por lo tanto, es apropiado utilizar las medidas de desempeño $accuracy$ y $error\ rate$.

$$accuracy = \frac{TP+TN}{TP+FN+FP+TN} = \frac{1247+756}{1247+183+214+756} = \frac{2003}{2400} = 0.8345 = 83.45\ \%$$

$$error\ rate = 1 - accuracy = 1 - \ 0.8345 = 0.1655 = 16.55\ \%$$

Para cualquier IR es válido calcular:

$$Sensitivity = \frac{TP}{TP + FN} = \frac{1247}{1247 + 183} = \frac{1247}{1430} = 0.8720 = 87.20\ \%$$

$$Specificity = \frac{TN}{FP + TN} = \frac{756}{756 + 214} = \frac{756}{970} = 0.7793 = 77.93\ \%$$

En consecuencia, se puede obtener:

$$Balanced\ Accuracy$$

$$= \frac{Sensitivity + Specificity}{2} = \frac{0.8720 + 0.7793}{2} = 0.8257\ \ = 82.57\%$$

Conclusiones

Con la tarea 10, expandimos nuestro conocimiento de métodos de validación al introducir leave-one-out y de algoritmos de clasificación al transitar por el del Clasificador Euclidiano, 1-NN y ahora una versión "generalizada" llamada k-NN que sigue formando parte del enfoque perezoso (lazy).

Las medidas de desempeño calculadas aplicando 7-NN y 3-NN para los datasets Nutt y Electricity, respectivamente y cuyas particiones P y E fueron generadas mediante el método de validación leave-one-out revelan resultados variados. Para el dataset Nutt se obtuvieron valores que según el contexto podrían considerarse malos mientras que para el dataset Electricity los valores podrían considerarse buenos.

Una limitación notable de k-NN para $k \geq 2$ al igual que 1-NN sigue siendo su alta demanda computacional. El algoritmo k-NN compara la distancia, en este caso euclidiana, de cada patrón de prueba contra todos los del conjunto de entrenamiento para identificar los k vecinos más próximos. Este proceso puede prolongar significativamente el tiempo de ejecución y aún más si se combina con leave-one-out, especialmente con datasets de gran volumen o alta dimensionalidad.

Referencias bibliográficas

[1] Nutt, C. L. et al. (2003). Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. Cancer Research, 63(7), 1602–1607.

[2] Datopian. (s. f.). *Electricity*. DataHub.

https://datahub.io/machinelearning/electricity#data

Anexos

```
import pandas as pd

# Carga del dataset
file_path = 'Electricity.csv'
data = pd.read_csv(file_path)

X = data.iloc[:, :-1].values  # Convertir a array de numpy para
facilitar el cálculo de distancias
y = data.iloc[:, -1].values

# Calculo de la distancia euclidiana
def euclidean_distance(row1, row2):
    return sum((r1 - r2) ** 2 for r1, r2 in zip(row1, row2)) ** 0.5

# Implementación del algoritmo k-NN
def knn(k, training_data, test_data_point):
    # Calcular la distancia entre el patrón de prueba y todos los
patrones de entrenamiento
    distances = [(euclidean_distance(training_data[i],
test_data_point), i) for i in range(len(training_data))]
    distances.sort(key=lambda x: x[0])

    # Obtener las etiquetas de los k vecinos más cercanos
    neighbors_labels = [y[distances[i][1]] for i in range(k)]

    # Voto por mayoría
    from collections import Counter
    label_count = Counter(neighbors_labels)
    return label_count.most_common(1)[0][0]

# Leave-One-Out
predictions = []
for i in range(len(X)):
    test_data_point = X[i]

    leave_one_out_training_data = [X[j] for j in range(len(X)) if j !=
i]

    predicted_label = knn(3, leave_one_out_training_data,
test_data_point)
    predictions.append(predicted_label)

results_df = pd.DataFrame({
```

```
    'Clase real': y,
    'Clase predicha': predictions
})

# Imprimir los resultados
print(results_df)
```