Libro sobre Ciencia de Datos y Deep Learning

Tú Nombre

April 24, 2025

Contents

Chapter 1

Ciencia de Datos y Machine Learning

1.1 Algoritmos de Aprendizaje Supervisado

1.1.1 Regresión Lineal y Logística

Regresión Lineal

La regresión lineal es un modelo que busca modelar la relación lineal entre una variable dependiente Y y una o más variables independientes X. El modelo se define como:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

donde:

- Y es la variable dependiente que queremos predecir.
- X es la variable independiente que utilizamos para la predicción.
- β_0 es la ordenada al origen, que representa el valor de Y cuando X es cero.
- β_1 es la pendiente de la recta, que indica cuánto cambia Y por un cambio unitario en X.
- ε es el término de error que captura la variabilidad no explicada por el modelo.

El objetivo es encontrar los valores de β_0 y β_1 que minimizan la suma de los cuadrados de los errores ε:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} (Y_i - (\beta_0 + \beta_1 X_i))^2$$

Esto se puede hacer utilizando técnicas como el método de mínimos cuadrados.

Regresión Logística

La regresión logística es un modelo utilizado para problemas de clasificación binaria. Se emplea la función logística, también conocida como la función sigmoide, para transformar la salida de la regresión lineal en un valor entre 0 y 1, que se interpreta como la probabilidad de pertenecer a la clase positiva. La función sigmoide está definida como:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

donde:

- $P(Y = 1)$ es la probabilidad de pertenecer a la clase positiva.
- e es la base del logaritmo natural.

El modelo de regresión logística busca encontrar los valores de β_0 y β_1 que maximizan la función de verosimilitud. La función de verosimilitud es el producto de las probabilidades condicionales de observar las etiquetas de clase dados los valores de X:

$$\max_{\beta_0,\beta_1} \mathcal{L}(\beta_0, \beta_1) = \prod_{i=1}^{n} P(Y_i)^{y_i} \cdot (1 - P(Y_i))^{1-y_i}$$

donde:

- $\mathcal{L}(\beta_0, \beta_1)$ es la función de verosimilitud.
- y_i es la etiqueta de la clase para la observación i.

La regresión logística se ajusta típicamente maximizando la log-verosimilitud para obtener estimaciones de β_0 y β_1.

Implementacion

```
# Ejemplo de Regresión Lineal
set.seed(123)
# Crear datos de ejemplo
X <- rnorm(100)
Y <- 2 * X + rnorm(100)

# Ajustar el modelo de regresión lineal
modelo_lineal <- lm(Y ~ X)

# Imprimir resultados
summary(modelo_lineal)

# Graficar el modelo
plot(X, Y, main = "Regresión Lineal", xlab = "X", ylab = "Y")
abline(modelo_lineal, col = "red")

# Ejemplo de Regresión Logística
set.seed(123)
# Crear datos de ejemplo para clasificación binaria
X <- rnorm(100)
probabilidades <- exp(2 * X) / (1 + exp(2 * X))
Y_binario <- rbinom(100, 1, probabilidades)

# Ajustar el modelo de regresión logística
modelo_logistico <- glm(Y_binario ~ X, family = binomial)

# Imprimir resultados
summary(modelo_logistico)

# Graficar el modelo
plot(X, Y_binario, main = "Regresión Logística",
         xlab = "X", ylab = "Y", col = Y_binario + 1)
curve(predict(modelo_logistico,
           data.frame(X = x), type = "response"),
           add = TRUE, col = "red")
```

1.1.2 Máquinas de Soporte Vectorial (SVM)

1. **Hiperplano:** En un espacio de características n-dimensional, un hiperplano es un subespacio de dimensión $n - 1$. Para un problema de clasificación binaria, un hiperplano divide el espacio en dos regiones, asignando puntos a una clase u otra.

2. **Margen:** El margen es la distancia perpendicular desde el hiperplano a los puntos más cercanos de cada clase. SVM busca el hiperplano que maximiza este margen, lo que se traduce en una mayor robustez y generalización del modelo.

3. **Vectores de Soporte:** Estos son los puntos de datos más cercanos al hiperplano y tienen un papel crucial en la definición del margen. Cambiar estos vectores de soporte afecta directamente al modelo, y son los únicos puntos que importan para la determinación del hiperplano.

4. **Función de Decisión:** La función de decisión de SVM es el hiperplano que se utiliza para clasificar nuevos puntos de datos. Dada una entrada, la función de decisión evalúa de qué lado del hiperplano cae el punto y asigna la etiqueta correspondiente.

5. **Kernel Trick:** SVM puede manejar eficientemente datos no lineales mediante el uso de funciones de kernel. Estas funciones transforman el espacio de características original en uno de mayor dimensión, permitiendo así que los datos sean separados de manera no lineal en el espacio transformado.

6. **Parámetros de SVM:**

   - **C (Parámetro de Regularización):** Controla el equilibrio entre tener un margen más amplio y clasificar correctamente los puntos de entrenamiento.
   - **Kernel:** Define la función de kernel utilizada (lineal, polinómica, radial, etc.).
   - **Gamma (para kernels no lineales):** Controla el alcance de influencia de un solo punto de datos en la decisión.

7. **Proceso de Entrenamiento:** Dado un conjunto de datos de entrenamiento etiquetado, SVM busca el hiperplano óptimo que maximiza el margen entre las clases. Esto se realiza a través de técnicas de optimización cuadrática.

8. **SVM para Regresión:** Además de la clasificación, SVM se puede utilizar para problemas de regresión. En este caso, el objetivo es ajustar un hiperplano de modo que contenga la mayor cantidad posible de puntos dentro de un margen predefinido.

9. **Ventajas y Desventajas:**

   - **Ventajas:** Efectivo en espacios de alta dimensión, eficaz en conjuntos de datos pequeños y versátil gracias al kernel trick.
   - **Desventajas:** Sensible a la escala de las características, puede ser computacionalmente costoso para grandes conjuntos de datos y requiere la elección cuidadosa de parámetros.

10. **Aplicaciones:** SVM se utiliza en una variedad de campos, como reconocimiento de escritura, clasificación de imágenes, diagnóstico médico, entre otros.

Elementos Matemáticos de las SVM

1. **Hiperplano:** Un hiperplano se define como $w \cdot x - b = 0$, donde w es el vector de pesos, x es el vector de entrada, y b es el sesgo.

2. **Margen:** El margen M entre un hiperplano y un punto x_i se define como $M = \frac{1}{\|w\|}|w \cdot x_i - b|$.

3. **Vectores de Soporte:** Los vectores de soporte son los puntos x_i que cumplen la condición $|w \cdot x_i - b| = 1/\|w\|$.

4. **Función de Decisión:** La función de decisión es $f(x) = w \cdot x - b$. Si $f(x) > 0$, el punto x se clasifica como clase 1; si $f(x) < 0$, se clasifica como clase -1.

5. **Kernel Trick:** La función de kernel $K(x, x')$ representa el producto escalar en un espacio de características de mayor dimensión. Ejemplos comunes incluyen el kernel lineal ($K(x, x') = x \cdot x'$), kernel polinómico ($K(x, x') = (x \cdot x' + 1)^d$), y kernel radial ($K(x, x') = \exp(-\gamma\|x - x'\|^2)$).

6. **Parámetros de SVM:**

   - C (Parámetro de Regularización): Se introduce en la función de pérdida para controlar el equilibrio entre tener un margen más amplio y clasificar correctamente los puntos de entrenamiento.

- w (Vector de Pesos): Aprende durante el entrenamiento y define la orientación del hiperplano.
- b (Sesgo): Parámetro de ajuste del hiperplano.
- γ (para kernels no lineales): Controla el alcance de influencia de un solo punto de datos en la decisión.

7. **Proceso de Entrenamiento:** El proceso de entrenamiento implica la minimización de la función de pérdida, que incluye el término de regularización $C\|w\|^2$ y la función de pérdida hinge.

8. **SVM para Regresión:** Para regresión, el objetivo es ajustar un hiperplano de modo que $|w \cdot x_i - b| \leq \epsilon$ para puntos de entrenamiento x_i.

9. **Ventajas y Desventajas:**

   - Ventajas: Efectivo en espacios de alta dimensión, eficaz en conjuntos de datos pequeños y versátil gracias al kernel trick.
   - Desventajas: Sensible a la escala de las características, puede ser computacionalmente costoso para grandes conjuntos de datos y requiere la elección cuidadosa de parámetros.

10. **Aplicaciones:** SVM se aplica en una variedad de problemas, como reconocimiento de escritura, clasificación de imágenes y diagnóstico médico.

1.1.3 Árboles de Decisión y Bosques Aleatorios

1. **Definición:** Un árbol de decisión es una estructura jerárquica en forma de árbol que se utiliza para representar decisiones y sus posibles consecuencias. Cada nodo interno del árbol representa una prueba en una característica, cada rama representa un resultado posible de la prueba, y cada hoja representa un resultado final o una decisión.

2. **Proceso de Construcción:** El árbol se construye de manera recursiva. En cada paso, se elige la mejor característica para dividir el conjunto de datos en función de algún criterio, como la ganancia de información o la impureza de Gini. Este proceso se repite hasta que se alcanza algún criterio de parada, como la profundidad máxima del árbol o un número mínimo de puntos en una hoja.

3. **Criterios de División:** Los criterios comunes para la división incluyen:

   - **Ganancia de Información:** Mide cuánta información nueva proporciona una característica.
   - **Impureza de Gini:** Mide la probabilidad de clasificar incorrectamente un elemento si es etiquetado aleatoriamente.

4. **Ventajas y Desventajas:**

   - **Ventajas:** Fácil interpretación, no requiere normalización de datos, manejo natural de características categóricas.
   - **Desventajas:** Propenso al sobreajuste, especialmente en conjuntos de datos pequeños y ruidosos.

Bosques Aleatorios

1. **Definición:** Un bosque aleatorio es una colección de árboles de decisión entrenados en subconjuntos aleatorios del conjunto de datos y utilizando técnicas de agregación para mejorar la precisión y controlar el sobreajuste.

2. **Proceso de Construcción:** Se crean múltiples árboles de decisión utilizando diferentes subconjuntos del conjunto de datos de entrenamiento y características aleatorias en cada división. Luego, las predicciones de cada árbol se promedian (en regresión) o se votan (en clasificación) para obtener la predicción final del bosque.

3. **Técnica de Bagging:** La construcción de árboles en subconjuntos aleatorios del conjunto de datos se conoce como bagging (bootstrap aggregating). Esto ayuda a reducir la varianza y evitar el sobreajuste al promediar los errores.

4. **Importancia de Características:** Los bosques aleatorios proporcionan una medida de la importancia de las características, que indica cuánto contribuye cada característica a la precisión del modelo. Esto es útil para la selección de características.

5. **Ventajas y Desventajas:**
   - **Ventajas:** Reducción del sobreajuste en comparación con un solo árbol, manejo automático del sobreajuste, buen rendimiento en conjuntos de datos grandes y complejos.
   - **Desventajas:** Menos interpretables que los árboles de decisión individuales.

Elementos Matemáticos de Árboles de Decisión

1. **Definición:** Un árbol de decisión se representa como una función $T(x)$ que asigna una instancia x a una hoja del árbol. Cada nodo interno j realiza una prueba en una característica $f_j(x)$, y cada rama representa una condición de prueba. La estructura del árbol se define por las funciones indicadoras $I(x, j)$ que indican si la instancia x llega al nodo j.

$$T(x) = \sum_{j=1}^{J} I(x, j) \cdot C_j$$

   Donde J es el número de nodos, C_j es el valor en la hoja correspondiente al nodo j, y $I(x, j)$ es 1 si la instancia x llega al nodo j y 0 de lo contrario.

2. **Proceso de Construcción:** La construcción del árbol implica seleccionar la mejor característica f_j y el umbral t_j en cada nodo j para maximizar la ganancia de información o reducir la impureza de Gini.

$$\text{Ganancia de Información: } Gain(D, j, t) = H(D) - \frac{N_L}{N} H(D_L) - \frac{N_R}{N} H(D_R)$$

$$\text{Impureza de Gini: } Gini(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2$$

   Donde D es el conjunto de datos en el nodo, D_L y D_R son los conjuntos de datos en los nodos izquierdo y derecho después de la división, N es el número total de instancias en D, y N_L y N_R son los números de instancias en los nodos izquierdo y derecho.

3. **Ventajas y Desventajas:**
   - **Ventajas:** Fácil interpretación, no requiere normalización de datos, manejo natural de características categóricas.
   - **Desventajas:** Propenso al sobreajuste, especialmente en conjuntos de datos pequeños y ruidosos.

Bosques Aleatorios

1. **Definición:** Un bosque aleatorio es una colección de B árboles de decisión $T_b(x)$, donde cada árbol se entrena en un subconjunto aleatorio de los datos de entrenamiento. La predicción se obtiene promediando (en regresión) o votando (en clasificación) las predicciones individuales de los árboles.

$$\text{Predicción del Bosque: } \hat{Y}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$

2. **Proceso de Construcción:** Cada árbol en el bosque se construye utilizando bagging, que consiste en seleccionar aleatoriamente un subconjunto de las instancias de entrenamiento con reemplazo. Además, en cada división de nodo, se selecciona un subconjunto aleatorio de características.

$$\text{Bagging: } D_b = \{(x_i, y_i)\} \text{ con } i \sim \text{Uniforme}(1, N)$$

$$\text{Características Aleatorias: } f_j \text{ con } j \sim \text{Uniforme}(1, P)$$

3. **Importancia de Características:** La importancia de la característica f_j se mide mediante la disminución promedio en la ganancia de impureza o la reducción en el error cuadrático medio cuando se utiliza f_j para dividir los nodos a lo largo de todos los árboles.

$$\text{Importancia de } f_j = \frac{1}{B}\sum_{b=1}^{B}\sum_{j \text{ en } T_b} \text{Importancia de } f_j \text{ en } T_b$$

1.1.4 Redes Neuronales

1.2 Aprendizaje No Supervisado

1.2.1 K-Means y Clustering Jerárquico

1.2.2 Análisis de Componentes Principales (PCA)

1.2.3 Algoritmos de Asociación

1.2.4 Mapas Autoorganizados (SOM)

1.3 Evaluación de Modelos y Métricas

1.3.1 Precisión, Sensibilidad, Especificidad

1.3.2 Curvas ROC y Área Bajo la Curva (AUC-ROC)

1.3.3 Matriz de Confusión

1.3.4 Validación Cruzada

1.4 Preprocesamiento de Datos

1.4.1 Normalización y Estandarización

1.4.2 Manejo de Datos Faltantes

1.4.3 Ingeniería de Características

1.4.4 Selección de Características

1.5 Optimización de Modelos

1.5.1 Hiperparámetros y Búsqueda en Cuadrícula

1.5.2 Optimización Bayesiana

1.5.3 Regularización

1.5.4 Redes Neuronales Convolucionales (CNN) y Recurrentes (RNN)

1.6 Aprendizaje por Refuerzo

1.6.1 Q-Learning

1.6.2 Algoritmos de Políticas

1.6.3 Exploración y Explotación

1.6.4 Funciones de Valor

1.7 Ética en el Machine Learning

1.7.1 Sesgo y Equidad

1.7.2 Transparencia y Explicabilidad

1.7.3 Privacidad y Seguridad

1.7.4 Responsabilidad en el Despliegue de Modelos

Chapter 2

Guía de Estudio de Deep Learning

2.1 Redes Neuronales Profundas

2.1.1 Perceptrones Multicapa (MLP)

2.1.2 Funciones de Activación: ReLU, Sigmoid, Tanh

2.1.3 Backpropagation

2.1.4 Regularización en Redes Neuronales

2.2 Redes Neuronales Convolucionales (CNN)

2.2.1 Convolutional Layers

2.2.2 Pooling Layers

2.2.3 Transfer Learning con CNN

2.2.4 Aplicaciones en Visión por Computadora

2.3 Redes Neuronales Recurrentes (RNN)

2.3.1 Arquitecturas de RNN

2.3.2 Long Short-Term Memory (LSTM)

2.3.3 Gated Recurrent Unit (GRU)

2.3.4 Aplicaciones en Procesamiento de Lenguaje Natural

2.4 Redes Generativas

2.4.1 Generative Adversarial Networks (GAN)

2.4.2 Variational Autoencoders (VAE)

2.4.3 Aplicaciones en Generación de Imágenes y Texto

2.5 Transferencia de Aprendizaje en Deep Learning

2.5.1 Fine-Tuning de Modelos Preentrenados

2.5.2 Domain Adaptation

2.5.3 Modelos Preentrenados como BERT, GPT

2.6 Técnicas Avanzadas

2.6.1 Normalización por Lotes (Batch Normalization)

2.6.2 Dropout

2.6.3 Redes Siamesas

2.6.4 Redes Neuronales Adversarias Condicionales (cGAN)

2.7 Herramientas y Frameworks

2.7.1 TensorFlow

2.7.2 PyTorch

2.7.3 Keras

2.7.4 TensorBoard para Visualización