**Nombre del Algoritmo:** Decision boundarysifier chains

**Descripción del Problema Computacional para la cual el algoritmo es una solución.**

**Delimitación del Problema:**

El problema computacional resuelto por un decision boundary (frontera de decisión) implica la separación de un espacio de características en regiones distintas asociadas a diferentes clases en un problema de clasificación. Dada una entrada 𝑋 en el espacio de características, se determina en qué lado de la frontera de decisión se encuentra para asignar una clase.

**Comprensión del Problema:**

En un problema de clasificación, la frontera de decisión define los límites entre clases. Esto es fundamental para:

* Clasificación binaria: Decidir entre dos clases 𝐶1​ y 𝐶2.
* Clasificación multiclase: Extender los límites a múltiples clases mediante combinaciones de fronteras. La frontera puede ser lineal (modelos como SVM lineal) o no lineal (modelos como redes neuronales o SVM con kernel).

**Clasificación del Problema:**

* Clasificación supervisada.
* Problemas geométricos y de optimización.
* Multidimensional: Puede operar en espacios de varias dimensiones.

**Viabilidad Computacional:**

La computación de la frontera de decisión depende del modelo subyacente:

* Modelos lineales: Computacionalmente eficientes (𝑂(𝑛) en general).
* Modelos no lineales: Puede ser costosa dependiendo de la complejidad del kernel o estructura.

**Contexto del Problema:**

Fronteras de decisión son útiles en:

* Reconocimiento de patrones (visión por computadora, procesamiento de señales).
* Diagnósticos médicos (clasificación de enfermedades).
* Sistemas de recomendación.
* Clasificación de texto.

**Análisis de Datos y Estructura del Problema:**

Datos:

* 𝑋: Características de las instancias.
* 𝑦: Clases asociadas.

Estructura:

* Modelos que optimizan la separación (SVM, regresión logística, etc.).
* Métricas para validar la calidad de la frontera (precisión, AUC).

**Evaluación de Complejidad:**

Entrenamiento:

* Modelos lineales: 𝑂(𝑛𝑑), donde 𝑛 es el número de datos y 𝑑 la dimensionalidad.
* Modelos no lineales: 𝑂(𝑛3) en algunos casos, como kernels complejos.

Predicción:

* 𝑂(𝑑) para una instancia en modelos simples.

**Análisis del Contexto del Problema para la cual el algoritmo es una solución**.

**Contexto general del Problema:**

El uso de una frontera de decisión es central en problemas de clasificación. Ejemplos prácticos incluyen:

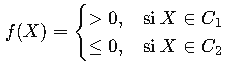
* Clasificación de imágenes (e.g., distinguir entre gatos y perros).
* Diagnóstico basado en datos de sensores.
* Clasificación de datos financieros para evaluar riesgo.

**Identificación de los conjuntos y estructuras de datos:**

* Conjuntos de datos:
  + Datos de entrada 𝑋∈𝑅𝑑.
  + Etiquetas 𝑦 ∈{𝐶1,𝐶2,…,𝐶𝑘}.
* Estructuras:
  + Hiperplanos (lineales) o superficies (no lineales) que dividen el espacio.

**Formalización del problema:**

Sea 𝑓(𝑋) una función de decisión tal que:



Para el caso multiclase, 𝑓(𝑋) puede ser extendido con múltiples funciones.

**Análisis de la Dinámica del Problema:**

* Entrenamiento: Definir 𝑓(𝑋) ajustando parámetros mediante optimización.
* Predicción: Evaluar 𝑓(𝑋) para una nueva instancia.

**Evaluación de la Complejidad del Contexto:**

* Complejidad: Relacionada con la dimensionalidad de los datos y el tipo de frontera.
* Límites: Modelos con datos muy complejos pueden requerir fronteras altamente no lineales.
* Generalización: Evitar el sobreajuste es crucial.

**Identificación de los casos límites y excepciones:**

* Datos completamente no lineales: Modelos lineales fallan.
* Desequilibrio en clases: La frontera puede estar sesgada.

**Interacción con otros sistemas:**

* Integración con sistemas de detección y clasificación en tiempo real.
* Uso como subcomponente en sistemas más grandes (e.g., análisis de flujo).

**Especificación Formal:**

**Definición de las entradas y salidas:**

Entradas:

* Conjunto de datos 𝑋∈𝑅𝑛×𝑑.
* Etiquetas 𝑦 ∈{𝐶1,𝐶2,…,𝐶𝑘}.

Salidas:

* Frontera de decisión 𝑓(𝑋).
* Predicciones para nuevos datos.

**Definición formal del problema mediante funciones:**

 Encontrar 𝑓(𝑋) tal que:

**Relaciones y Restricciones:**

* La frontera debe minimizar errores de clasificación.
* Restricciones del modelo subyacente (e.g., linealidad).

**Invariantes y propiedades:**

* 𝑓(𝑋)debe ser consistente para datos similares.

**Control de Flujo:**

* Entrenar modelo basado en 𝑋,𝑦.
* Generar frontera 𝑓(𝑋)
* Evaluar 𝑓(𝑋) para clasificar nuevas instancias.

**Pseudocodigo del Algoritmo**

Algoritmo DecisionBoundary

Definir datos\_X como Matriz[n][d] // Matriz de características

Definir etiquetas\_Y como Vector[n] // Clases asociadas

Definir modelo // Algoritmo para aprender la frontera

// Entrenamiento del modelo

Escribir "Entrenando modelo para encontrar la frontera de decisión"

modelo <- EntrenarModelo(datos\_X, etiquetas\_Y)

// Predicción de nuevos datos

Definir nuevo\_X como Vector[d] // Nueva instancia

Escribir "Ingrese los valores de las características del nuevo dato:"

Para i <- 1 Hasta d Hacer

Leer nuevo\_X[i]

FinPara

// Clasificación

clase\_predicha <- modelo.Prediccion(nuevo\_X)

Escribir "La clase predicha es: ", clase\_predicha

FinAlgoritmo

**Caso de Usos Posibles:**

* Clasificación de Texto:
  + Determinar si un correo es spam o no.
* Diagnóstico Médico:
  + Predecir si un paciente tiene una enfermedad basada en datos biomédicos.
* Sistemas de Recomendación:
  + Identificar si un producto es relevante para un usuario.
* Visión por Computadora:
  + Clasificar imágenes en categorías.