**Nombre del Algoritmo:** Classifier chains

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

**Delimitación del Problema:**

El algoritmo Classifier Chains está diseñado para resolver problemas de clasificación multietiqueta, donde cada instancia puede estar asociada con múltiples etiquetas simultáneamente. A diferencia de los modelos independientes, este enfoque modela explícitamente las dependencias entre etiquetas.

**Comprensión del Problema:**

El desafío radica en predecir un conjunto de etiquetas 𝑌 = {𝑦1, 𝑦2,…,𝑦𝐿} para una instancia 𝑥, considerando que las etiquetas pueden estar correlacionadas. Los métodos tradicionales, que asumen independencia entre etiquetas, pierden información clave, mientras que las cadenas de clasificadores aprovechan estas dependencias.

**Clasificación del Problema:**

* Aprendizaje Supervisado: Utiliza un conjunto de datos etiquetados.
* Clasificación Multietiqueta: Predice múltiples etiquetas para cada instancia.
* Secuencial/Dependiente: Cada etiqueta se predice basándose en las etiquetas previamente predichas.

**Viabilidad Computacional:**

Es computacionalmente viable para problemas con un número moderado de etiquetas. Su complejidad crece linealmente con el número de etiquetas, pero puede volverse costoso en problemas con cientos de etiquetas.

**Contexto del Problema:**

Este algoritmo es útil en dominios donde las etiquetas tienen relaciones estructurales o contextuales. Ejemplos incluyen:

* Clasificación de documentos (donde etiquetas representan temas relacionados).
* Reconocimiento de emociones (múltiples emociones pueden coexistir).
* Diagnóstico médico (varias condiciones pueden estar presentes simultáneamente).

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

**Datos:**

* Características 𝑋: Representación de las instancias.
* Etiquetas 𝑌: Matriz binaria indicando la presencia/ausencia de cada etiqueta.

Estructura:

* Secuencia de clasificadores 𝐶1,𝐶2,…,𝐶𝐿, donde 𝐶𝑖 predice la etiqueta 𝑦𝑖 basándose en 𝑋 y en las etiquetas previas 𝑦1,…,𝑦𝑖−1

**Evaluación de Complejidad:**

* Entrenamiento: 𝑂(𝐿⋅𝑇(𝐶)) donde 𝐿 es el número de etiquetas y 𝑇(𝐶) es el tiempo de entrenamiento para cada clasificador.
* Predicción: 𝑂(𝐿⋅𝑃(𝐶)) donde 𝑃(𝐶) es el tiempo de predicción para cada clasificador.

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

**Contexto general del Problema:**

Classifier Chains aborda problemas multietiqueta en los que la correlación entre etiquetas puede mejorar significativamente la precisión del modelo. Ejemplos incluyen:

* Sistemas de recomendación.
* Análisis de contenido multimedia.
* Bioinformática (como la predicción de funciones de proteínas).

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

* Conjuntos de datos:
  + X: Matriz de características de tamaño 𝑛×𝑑, donde 𝑛 es el número de instancias y 𝑑 el número de características.
  + 𝑌: Matriz de etiquetas de tamaño 𝑛×𝐿, donde 𝐿 es el número de etiquetas.
* Estructuras:
  + Cadena ordenada de clasificadores.
  + Dependencias entre etiquetas representadas explícitamente.

**Formalización del problema:**

El problema puede definirse como encontrar un conjunto de funciones 𝑓1, 𝑓2,…,𝑓𝐿 tal que:



Donde f i es un modelo entrenado para predecir la etiqueta 𝑦𝑖 considerando las etiquetas anteriores.

**Análisis de la Dinámica del Problema:**

1. Entrenamiento
   * Cada 𝑓𝑖 se entrena secuencialmente usando 𝑋 y las etiquetas previas 𝑦1,…,𝑦𝑖−1.
2. Predicción:
   * Cada instancia 𝑥 pasa a través de la cadena 𝑓1,…,𝑓𝐿 para generar las predicciones 𝑦1,…,𝑦𝐿 .

**Evaluación de la Complejidad del Contexto:**

* Ventajas:
  + Captura las dependencias entre etiquetas.
  + Flexible y extensible.
* Desventajas:
  + Sensible al orden de las etiquetas.
  + Errores acumulativos pueden propagarse a lo largo de la cadena.

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

* Casos límites:
  + Etiquetas completamente independientes (desperdicio de recursos computacionales).
* Excepciones:
  + Datos ruidosos o etiquetas altamente desequilibradas pueden afectar negativamente el rendimiento.

**Interacción con otros sistemas:**

* Puede integrarse con sistemas de clasificación de documentos, procesamiento de texto o análisis de imágenes.
* Compatible con otros modelos base (regresión logística, SVM, etc.).

**Especificación Formal:**

**Definición de las entradas y salidas:**

* Entradas:
  + Conjunto de características 𝑋: Matriz 𝑛×𝑑.
  + Conjunto de etiquetas 𝑌: Matriz 𝑛×𝐿.
* Salidas:
  + Predicción de etiquetas 𝑌^={𝑦^1,…,𝑦^𝐿}.

**Definición formal del problema mediante funciones:**

Para cada Para cada instancia 𝑥 y etiqueta 𝑦𝑖:



**Relaciones y Restricciones:**

* Cada clasificador depende de las predicciones previas.
* Restricción: Debe preservarse el orden de la cadena durante entrenamiento y predicción.

**Invariantes y propiedades:**

* Las etiquetas predichas deben ser coherentes entre sí según las dependencias.

**Control de Flujo:**

* Entrenar clasificadores 𝑓1,𝑓2,…,𝑓𝐿 secuencialmente.
* Durante la predicción:
* Predice 𝑦^1​ usando 𝑓1.
* Usa 𝑦^ para predecir 𝑦^2 con 𝑓2, y así sucesivamente.

**Pseudocodigo del Algoritmo**

Algoritmo ClassifierChains

// Entradas:

// datos\_entrenamiento: Matriz[n][m], n instancias y m atributos.

// etiquetas\_entrenamiento: Matriz[n][k], k etiquetas para cada instancia.

// cadena\_orden: Vector[k], orden de las etiquetas en la cadena (puede ser aleatorio o definido).

Funcion EntrenarCadenas(datos\_entrenamiento, etiquetas\_entrenamiento, cadena\_orden)

Definir clasificadores como ListaVacia

Definir datos\_actualizados como datos\_entrenamiento

Para i desde 1 hasta Longitud(cadena\_orden)

etiqueta\_actual <- cadena\_orden[i]

// Crear nuevas características usando etiquetas previas

datos\_actualizados <- ConcatenarColumnas(datos\_actualizados, etiquetas\_entrenamiento[, 1:i-1])

// Entrenar clasificador para la etiqueta actual

clasificador <- EntrenarClasificador(datos\_actualizados, etiquetas\_entrenamiento[, etiqueta\_actual])

AgregarALista(clasificadores, clasificador)

FinPara

Retornar clasificadores

FinFuncion

Funcion PredecirCadenas(clasificadores, instancia, cadena\_orden)

Definir predicciones como ListaVacia

instancia\_actualizada <- instancia

Para i desde 1 hasta Longitud(cadena\_orden)

clasificador\_actual <- clasificadores[i]

// Predecir la etiqueta actual

prediccion <- Predecir (clasificador\_actual, instancia\_actualizada)

AgregarALista(predicciones, prediccion)

// Actualizar la instancia con la predicción actual

instancia\_actualizada <- ConcatenarColumnas(instancia\_actualizada, prediccion)

FinPara

Retornar predicciones

FinFuncion

// Proceso principal

Escribir "Cargando datos..."

datos\_entrenamiento <- CargarDatos()

etiquetas\_entrenamiento <- CargarEtiquetas()

Escribir "Definiendo el orden de la cadena..."

cadena\_orden <- GenerarOrdenAleatorio(Longitud(etiquetas\_entrenamiento[0]))

Escribir "Entrenando cadena de clasificadores..."

clasificadores <- EntrenarCadenas(datos\_entrenamiento, etiquetas\_entrenamiento, cadena\_orden)

Escribir "Clasificadores entrenados. Realizando predicciones..."

nueva\_instancia <- CargarNuevaInstancia()

predicciones <- PredecirCadenas(clasificadores, nueva\_instancia, cadena\_orden)

Escribir "Predicciones para las etiquetas: ", predicciones

FinAlgoritmo

**Caso de Usos Posibles:**

1. Clasificación de texto:
   * Asignar múltiples temas a un artículo.
2. Análisis de contenido multimedia:
   * Etiquetar videos o imágenes con múltiples categorías (por ejemplo, "naturaleza", "animal", "paisaje").
3. Bioinformática:
   * Predicción de múltiples funciones proteicas simultáneamente.
4. Sistemas de recomendación:
   * Identificar múltiples preferencias de usuarios basadas en su historial.