PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
Escuela de Ingeniería
IIC2613 – Inteligencia Artificial
2023-2

Pauta Control Largo 2

Pregunta 1

a) Considere dos conjuntos de puntos en $\mathbb{R}^n$, C_1 y C_2, donde cada punto tiene una etiqueta binaria en $\{-1, 1\}$. ¿Es necesario que C_1 y C_2 sean iguales para que los SVM entrenados sobre ellos (uno para C_1 y otro para C_2) sean iguales? ¿Si no, qué condiciones deben cumplir? Justifique su respuesta.

Respuesta:
No, no es necesario que C_1 y C_2 sean iguales para que el SVM entrenado sobre sí sea idéntico, solamente basta con que los vectores de soporte entre ambos conjuntos sean compartidos.

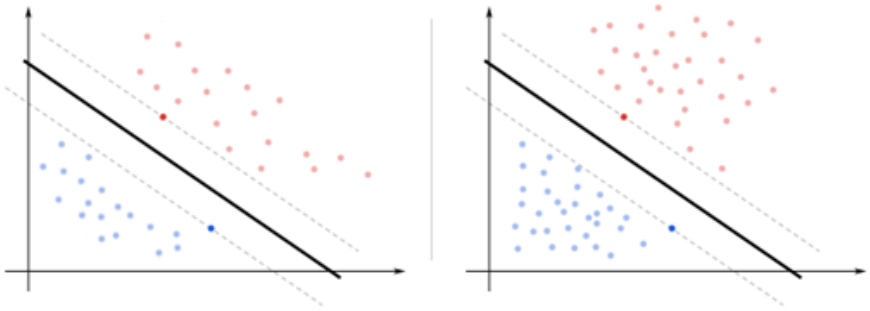

Figura 1: Dos conjuntos de puntos distintos con los mismos vectores de soporte

Esto se debe a que los vectores de soporte son aquellos puntos sobre los cuales se maximiza el margen de separación entre los conjuntos de datos, por lo que solamente necesitamos de estos puntos para modelar un clasificador de SVM sobre los datos.

b) Si se tiene la función $V^*(s)$ para un problema de decisión markoviano, ¿cómo se puede obtener la política óptima a partir de esta? ¿Se puede obtener la misma política óptima si ahora se usa Q-learning para estimar la función $Q^*(s, a)$ del problema?

Respuesta:
Dada la función $V^*(s)$, la política óptima π^* se obtiene buscando para cada estado, aquella acción que lleva al estado con mayor valor futuro, de acuerdo a la función de valor, es decir:

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma V^*(s'))$$

Por otro lado, Q-Learning es capaz de aproximar la política óptima dado infinito tiempo de entrenamiento y una política inicial aleatoria. Es importante notar que esta optimalidad está garantizada con $t \to \infty$, por lo que es posible que no obtengamos la política óptima cada vez que lo ejecutemos.

c) ¿Qué problema puede generar en un SVM multiclase el un problema con un alto desbalance de clases (número de ejemplos por clase muy distinto)? Indique cómo modificaría la función objetivo de un SVM multiclase para poder controlar este problema.

Respuesta:
Un SVM multiclase con gran desbalance de clases podría priorizar el separar mejor clases más frecuentes por sobre aquellas menos presentes debido a que existirán menores violaciones al margen por parte de aquellas con menor frecuencia.

Una forma de lidiar con esto es cambiar el ponderador C para ser específico en cada clase y de carácter inversamente proporcional a la frecuencia de la clase, a modo de que clases subrepresentadas tengan un valor C_i mayor.

La nueva función objetivo sería:

$$\min_{w_1, w_2 \ldots w_K, \xi} \quad \frac{1}{2} w_k^T w_k + C_i \sum_{(x_i, y_i) \in D} \xi_i$$

Pregunta 2

a) Considere un problema de procesamiento de lenguaje natural, donde se busca construir un modelo que permita predecir la siguiente palabra en una frase. Para esto se tiene un conjunto de datos de gran escala, con frases de distinto largo, provenientes de la literatura clásica española.

i) ¿Como resolvería el problema usando aprendizaje supervisado? Indique cómo representaría vectorialmente los datos y etiquetas, cómo generaría el conjunto de entrenamiento y qué tarea supervisada buscaría resolver.

Respuesta:
Se puede llevar cada texto a un vector utilizando técnicas similares a las de la Tarea 4 del curso tal como *Bag of Words*, una vez los textos se encuentran representados vectorialmente, podemos tomar fragmentos de estos textos (oraciones) y remover su última palabra.

La red deberá predecir qué palabra sigue en la oración en base a la entrada de la oración, siendo la etiqueta la palabra removida. La siguiente palabra a predecir sería entregada como un vector de K elementos, con K el número de palabras en el vocabulario, donde aquella palabra con mayor probabilidad sería utilizada para continuar el texto.

(Fun fact: A grandes razgos, así es como funciona ChatGPT)

Se aceptan otro tipo de respuestas siempre y cuando se encuentren bien argumentadas y hagan sentido.

ii) Suponiendo que el problema no es linealmente separable, diseñe una red neuronal que permita resolverlo, especificando cada uno de los elementos que la definen. Una vez hecho esto, diagrame el grafo de cómputo asociado al problema de optimización que permite obtener los parámetros de la red.

Respuesta:
La red deberá tener una capa de entrada equivalente al largo de la oración que recibirá para predecir, N capas intermedias y una capa de salida de K elementos, con K siendo la cardinalidad del vocabulario. Además, aplicaremos una función softmax en la salida para obtener una distribución de probabilidad sobre las salidas. Finalmente, luego de generar la salida, se debe incorporar una función de pérdida para poder entrenar el modelo. Para un problema simplificado, la red se vería algo así:

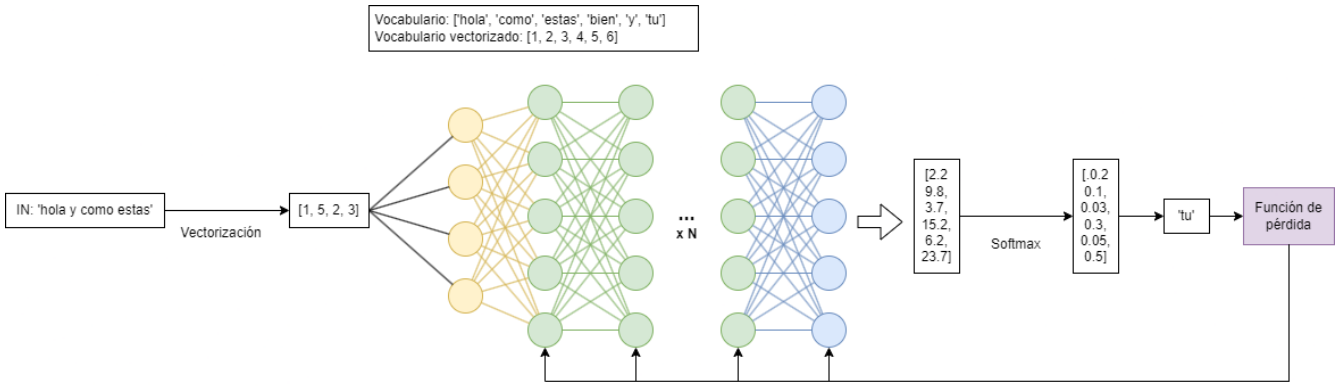

Figura 2: Grafo de cómputo de la red

b) Responda las siguientes preguntas sobre árboles y ensambles:

i) ¿Cómo se podría modificar un árbol de decisión (algoritmo y/o modelo) para que genere particiones basadas en rectas con pendiente arbitraria (no necesariamente ortogonales a los ejes)?

Respuesta:
Una forma de lograr esto es modificando el criterio de separación de las ramas del árbol en base a una operación entre dos características. Estas combinaciones también serían consideradas a la hora de computar la entropía, lo que si bien supone mucho mayor esfuerzo computacional, podría entregar mejores resultados.

A continuación se presenta un ejemplo particular, con un árbol simple que clasifica si la distancia Manhattan entre el origen y un punto es mayor a 5:

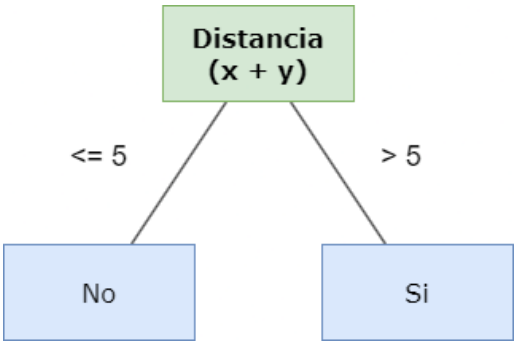

Figura 3: Árbol de decisión asociado al problema.

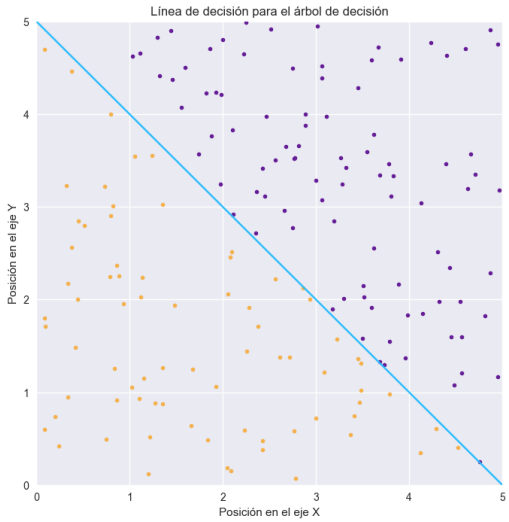

Figura 4: Línea de decisión para el árbol

Se aceptan otro tipo de respuestas siempre y cuando se encuentren bien argumentadas y hagan sentido.

ii) ¿Qué condiciones deben cumplir los clasificadores usados en un esquema de ensamble tipo *bagging*? ¿Cómo mediría estas condiciones, para dar garantía de que los modelos utilizados son adecuados?

Respuesta:
Los modelos deben tener baja correlación entre sí, de este modo deberían tener patrones de errores distintos que se cancelen con suficientes clasificadores en el ensamble.

De este modo, los clasificadores utilizados deben ser débiles, con rendimiento poco mejor al azar (si ajustan muy bien al conjunto de datos comenzarán a tener alta correlación entre sí), suelen ser entrenados con una fracción del set de datos total y un subconjunto de las características disponibles para describirlo.

Mecanismos para dar garantía de que los modelos son adecuados pueden ser estudios de diversidad, calculando la correlación entre las predicciones de los clasificadores y asegurándose de que sean bajas. Además, podemos validar el rendimiento del ensamble completo utilizando las métricas de rendimiento convencionales que conocemos tales como *accuracy*, precisión, *recall* y *F1-score*.