Machine Learning (PEC3)

Máster de Bioestadística y Bioinformática - Diciembre 2023

Secuencias promotoras en E. Coli

Los promotores son secuencias de ADN que afectan la frecuencia y ubicación del inicio de la transcripción a través de la interacción con la ARN polimerasa.

Este estudio se basa en los ficheros obtenidos de:

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Para más información, se puede recurrir a la siguiente referencia acerca del estudio de promotores en E. Coli: Harley, C. and Reynolds, R. 1987. "Analysis of E. Coli Promoter Sequences." Nucleic Acids Research, 15:2343-2361

Los atributos del fichero de datos son:

1. Un símbolo de {+/-}, indicando la clase ("+" = promotor).
2. El nombre de la secuencia promotora. Las instancias que corresponden a no promotores se denominan por la posición genómica.
3. Las restantes 57 posiciones corresponden a la secuencia.

La manera elegida para representar los datos es un paso crucial en los algoritmos de clasificación. En el caso que nos ocupa, análisis basados en secuencias, se usará la transformación denominada **one-hot encoding**.

El one-hot encoding representa cada nucleótido por un vector de 4 componentes, con 3 de ellas a 0 y una a 1 indicando el nucleótido. Pongamos por ejemplo, el nucleótido T se representa por (1,0,0,0), el nucleótido C por (0,1,0,0), el nucleótido G por (0,0,1,0) y el nucleótido A por (0,0,0,1).

Por tanto, para una secuencia de 57 nucleótidos, como en nuestro caso, se obtendrá un vector de 4*57=228 componentes, resultado de concatenar los vectores para cada uno de los 57 nucleótidos.

Una vez realizada la transformación, one-hot encoding el objetivo se trata de predecir con SVM si la secuencia es un promotor o no, y comparar sus rendimientos.

Enunciado

1. Escribir en el informe una sección con el título: "Algoritmo Support Vector Machine" en el que se haga una breve explicación de su funcionamiento y sus características y, además, se presente una tabla de sus fortaleza y debilidades.

2. Implementar una función para realizar una transformación one-hot encoding de las secuencias del fichero de datos `promoters.txt`. En caso de no lograr la implementación de dicha transformación, se puede utilizar el fichero `promoters_onehot.txt` con las secuencias codificados según una codificación one-hot para completar la actividad.

3. Desarrollar un código en R (o en Python) que implemente un clasificador de SVM. El código debe:

(a) Leer y codificar los datos con la función one-hot desarrollada.
(b) Utilizando la semilla aleatoria 12345, separar los datos en dos partes, una parte para training (67%) y una parte para test (33%).
(c) Utilizar el kernel lineal y el kernel RBF para crear sendos modelos SVM basados en el training para predecir las clases en los datos del test.
(d) Usar el paquete `caret` con el modelo `svmLinear` para implementar un SVM con kernel lineal y 3-fold crossvalidation. Comentar los resultados.
(e) Evaluar el rendimiento del algoritmo SVM con kernel RBF para diferentes valores de los hiperparámetros `C` y `sigma`. Orientativamente, se propone explorar valores de `sigma` en el intervalo (0.005,0.5) y valores de `C` en el intervalo $(0.1, 2)$. Una manera fácil de hacerlo es utilizar el paquete `caret` con el modelo `svmRadial`. Mostrar un gráfico del rendimiento según los valores de los hiperparámetros explorados. Comentar los resultados.
(f) Crear una tabla resumen de los diferentes modelos y sus rendimientos. Comentar y comparar los resultados de la clasificación en función de los valores generales de la clasificación como `accuracy` y otros para los diferentes clasificadores obtenidos. ¿Qué modelo resulta ser el mejor?

Informe de la PEC

Las soluciones se presentarán mediante un informe dinámico R markdown (o Python notebook) con la estructura habitual de los ejercicios no evaluables realizados hasta ahora. En primer lugar, el informe tendrá un título (igual que el de la PEC), el autor, la fecha de creación y el índice de apartados de la PEC. En segundo lugar, se crea una sección con el título "Algoritmo Support Vector Machine" donde se haga una breve explicación de su funcionamiento y sus características. Además, se presenta la tabla de sus fortaleza y debilidades. En tercer lugar, se realizan los diferentes apartados de la PEC. Al final se crea una sección "Discusión final" para comentar todos los resultados obtenidos y escoger el mejor modelo. Una característica que se valorará es hasta qué punto es el informe "dinámico". En el sentido de adaptarse el informe a cambios en los datos. Se subirá al apartado de entregas un zip con los siguientes ficheros:

1. Fichero ejecutable (.Rmd o .ipynb) que incluya un texto explicativo que detalle los pasos implementados en el script y el código de los análisis. No olvidar de incluir todos los ficheros complementarios que hagan falta para la correcta ejecución: ficheros de datos, fichero de bibliografía, imágenes, etc. **NOTA**: Para facilitar la ejecución, no usar un ruta fija para la lectura del fichero, asociarlo al área de trabajo donde este el script (.Rmd o .ipynb).

2. Informe (pdf o html) resultado de la ejecución del fichero Rmd (o .ipynb) anterior. Antes de enviar el zip, se recomienda verificar la reproducibilidad del fichero (.Rmd o .ipynb) para obtener el informe en formato pdf o html sin ninguna dificultad.

En resumen, se puede entregar la PEC programando en R o Python, según vuestra conveniencia.

Puntuaciones de los apartados

Apartado 1 (5%), Apartado 2 (20%), Apartado 3 (65%), Calidad del informe dinámico (10%).