CIMAT Centro de Investigación en Matemáticas, A.C.

Taller Interdisciplinario de Problemas

Teoria de Juegos aplicada a los datos de antibióticos del CAMDA 2024

por

Alejandro Sierra Conde
Kotaro Hata
Ronald Cárdenas Catota

Programa

Doctorado en Ciencias con orientación Probabilidad y Estadística

2024

Índice

1. Introducción

Uno de los problemas en análisis de datos de microbioma es la alta dimensionalidad, por lo cual se busca utilizar métodos de reducción de dimensionalidad para encontrar el conjunto de características más relevantes, no redundantes e interdependientes. La importancia de elegir un conjunto de características óptimo es clave al momento de entrenar el modelo de aprendizaje para clasificar debido a que la precisión del modelo depende del conjunto de características con el cual se va entrenar. Existen tres enfoques: inmersión, envolventes y filtro, durante el reto CAMDA 2024 se abordo diferentes métodos de reducción de dimensionalidad como son: PCA, Lasso y Ridge. Por lo cual es natural aplicar otro método de selección de características con el fin de mejorar la precisión del modelo de aprendizaje, para el presente trabajo utilización un método de filtro para selección de variables basado en la ponderación dinámica de teoría de juegos (**GTDWFE**). La motivación de utilizar este enfoque de teoría de juegos para selección de variables viene de los resultados obtenidos en articulo [Chowdhury et al, 2019] donde se observo que las métricas del modelo aumentaron para predecir cuando una secuencia de proteínas pertenece a la familia AMR(antimicrobial resistance) en bacterias Gram-negative.

El trabajo se desarrollo de la siguiente manera: en la sección 1 Introducción se presenta la motivación del enfoque a usar, en la sección 2 Definiciones se describen y enuncian resultados relevantes sobre objetos matemáticos a usar, en la sección 3 Filtro de variables se describe el índice de poder Banzaf que es el criterio para selección de variables, en la sección 4 Aplicación se presentan los resultados obtenidos de aplicar el enfoque de teoría de juegos a los datos del CAMDA 2024 para el problema de antibióticos usando SVM(Support Vector Machine).

2. Definiciones

Coeficiente de correlación de Pearson. Se tiene que medir alguna relación de dependencia entre las características, una opción es el coeficiente e correlación de Pearson entre una característica y las diferentes clases/grupos que componen las columnas de los datos; si f es una característica, C representa una clase, consideramos las correspondientes medias μ_f, μ_C y las desviaciones estándar σ_f, σ_f. El coeficiente de Pearson de f con respecto a la clase C se define como

$$R_\nu(f) = \frac{E\left[(f-\mu_f)(C-\mu_f)\right]}{\sigma_f \sigma_C}. \tag{2.1}$$

Índice de Tanimoto. El índice de Tanimoto, también conocido como coeficiente de Tanimoto o coeficiente de Jaccard-Tanimoto, es una generalización del coeficiente de Jaccard. Es una medida de similaridad, para características f, y $f_j \neq f$, se calcula

$$TC(f, f_j) = \frac{f \cdot f_j}{\|f\|^2 + \|f_j\|^2 - f \cdot f_j},$$

para medir la similaridad entre una característica fija y una clase de características se

puede obtener el promedio

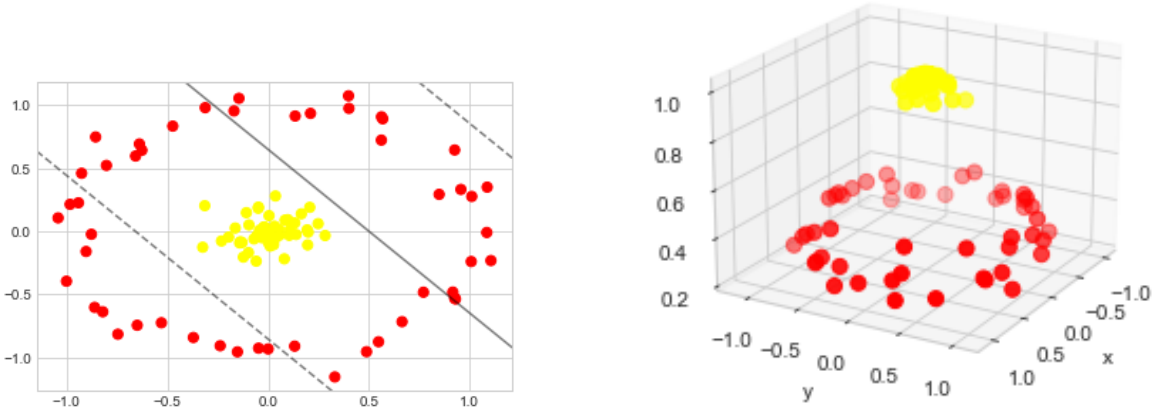
$$R_d(f) = \frac{1}{d-1} \sum_{f \neq f_j, f_j \in C} TC(f, f_j). \tag{2.2}$$

Información mutua y condicional. Queremos medir la interdependencia la relevancia entre características, una forma de medir esto es con la entropía conjunta y condicional, es decir, si U, V y Z son variables aleatorias, entonces la información conjunta y condicional se puede calcular como

$$I(U;V) = \sum_{u \in U} \sum_{v \in V} p(u,v) \log \left(\frac{p(u,v)}{p(u)\, p(v)} \right),$$

$$I(U;V|Z) = \sum_{u \in U} \sum_{v \in V} \sum_{z \in Z} p(u,v,z) \log \left(\frac{p(u,v|z)}{p(u|z)\, p(v|z)} \right)$$

Máquina de soporte vectorial. Una máquina de soporte vectorial (MSV) es un algoritmo de aprendizaje supervisado que representa cada elemento de datos como un punto en un espacio n-dimensional y construye un hiperplano (frontera de decisión) para separar los puntos de datos en dos grupos. El conjunto principal de vectores que identifican el hiperplano se conoce como vectores de soporte. A diferencia de muchos clasificadores, una MSV evita el sobreajuste regularizando sus parámetros. El sobreajuste ocurre cuando un clasificador modela tan bien los datos de entrenamiento que afecta la precisión del clasificador en datos nuevos. La MSV ha demostrado ser un buen clasificador para secuencias de proteínas, clasificándolas con alta precisión. Como predecir la resistencia antimicrobiana (AMR) es un problema de clasificación binaria, elegimos una MSV para este trabajo. Se utilizó una función de base radial como su núcleo/kernel, en particular **rbf** que corresponde a una función Gaussiana.



(a) MSV con kernel **lineal**.

(b) MSV con kernel **rbf**.

Figura 1: MSV implementando el uso del kernel

A la izquierda de la figura se observa que al aplicar MSV a datos que no son linealmente separables se obtiene un precisión baja al momento de clasificarlos. MSV implementa el uso de kernel para clasificar datos que nos son linealmente separables. De manera

intuitiva el kernel transforma nuestros datos proyectándolos a un espacio de dimensión superior donde podemos aplicar el clasificar lineal MSV. En la figura de la derecha usamos un kernel de base radial para proyectar los datos donde se observar que bajo este kernel el clasificador lineal SVM es aplicable.

3. Filtro de variables

El **índice de poder Banzhaf** mide el poder de un jugador dentro de un sistema de votación. Cada jugador(votante) tiene un peso de voto y se determina cuán a menudo cada jugador puede cambiar el resultado de su votación al cambiar su voto. Para calcular el poder de un votante, se listan todas las coaliciones ganadoras, luego se cuentan los votantes críticos. Un votante crítico es un votante que, si cambia su voto, causaría que la coalición pierda. Luego se debe convertir este conteo a fracciones al dividir por el total de veces que un jugar es crítico

$$\text{Índice de poder Banzhaf de } v = \frac{\text{Número de veces que } v \text{ es crítico}}{\text{Total de veces que todos los votadores son críticos}}.$$

Para evaluar el conjunto de características representativo, se selecciona un número pequeño de características basado en una estimación de relevancia, redundancia e interdependencia de todas las características. Se calcula el peso usando las estimaciones anteriores y se reajuste del peso de una característica se realiza de manera dinámica al añadir una nueva característica seleccionada al subconjunto de características seleccionadas previamente. De este modo, el peso de una característica se asemeja a la relación de interdependencia con las características seleccionadas anteriormente. El detalle del procedimiento se muestra en el **Algoritmo 1** el cual se presenta en el articulo [Chowdhury et al, 2019]:

Algoritmo 1. Evaluación de características basada en la ponderación dinámica de la teoría de juegos.

Entrada: Conjunto de datos D', conjunto de características F, clase C, evaluar T
Salida: Mejor subconjunto de características K

1: $K := \emptyset$;
2: $w(f) := 1$ para toda $f \in F$;
3: calcular $R_v(f)$ and $R_d(f)$ para toda $f \in F$ usando ecuaciones $(2.1) - (2.2)$
4: $R_{\text{sum}}(f) := R_v(f) + R_d(f)$ para toda $f \in F$;
5: **Para** $j \leftarrow 1$ to T **Hacer**
6: **Para** $f \in F$ **Hacer**
7: $L(f) := R_{sum}(f) \times w(f)$;
8: **Fin Para**
9: $select f_h$ con más grande $L(f)$;
10: $K := K \cup \{f_h\}$;
11: $F := F \setminus \{f_h\}$;
12: **Si** $|K| \neq T$ **Entonces**
13: **Para** $f \in F$ **Hacer**
14: calcular el índice de potencia de Banzhaf$\phi_B(f)$ sobre$K$ usando ecuaciones $(3.1)-(3.3)$;
15: $w(f) := w(f) \times (1 + \phi_B(f))$;
16: **Fin Para**
17: **Fin Si**
18: **Fin Para**
19: **retorna** K ▷ Mejor subconjunto de características

Para implementar el Algoritmo se inicializan los pesos $w(f)$ para cada característica f igual a 1. La relevancia para la clase objetivo $R_v(f)$ y el valor de similaridad de la característica $R_d(f)$ son calculados usando las ecuaciones (2.1)-(2.2) respectivamente. De manera intuitiva, cuanto mayor sea la relevancia de una característica para la clase objetivo (secuencia AMR o no-AMR), más puede contribuir a la predicción compartiendo información con la clase objetivo. Además, cuanto mayor sea la distancia entre una característica y todas las demás, menor será la similitud de la característica con las demás, lo que indica una menor redundancia.

Luego el algoritmo itera hasta seleccionar T características. Para cada iteración el valor de $L(f)$ es calculado y la característica con el mayor $L(f)$ es seleccionada y añadida dentro del subconjunto K y entonces eliminada del conjunto de características F. Los pesos de las características candidatas restantes se recalculan en cada iteración para determinar el impacto de las características candidatas en las características seleccionadas anteriormente (líneas 13-16). Utilizamos el índice de poder de Banzhaf para reajustar el peso $w(f)$ de una característica f. Para cada coalición ganadora de $S \cup \{r\}$ si S perdería sin el jugador r, entonces$r$ es crucial para ganar el juego. Dado que el jugador r es una característica, se hace una ligera modificación del índice de poder de Banzhaf original, y la definición actualizada del índice de poder de Banzhaf para un jugador r se da como sigue

$$\phi_B(r) = \frac{1}{|\Pi_\delta|} \sum_{S \subset \Pi_\delta} \Delta_r(S), \tag{3.1}$$

donde la contribución marginal de la característica r a todas las coaliciones es $\Delta_r(s)$ donde $\Delta_r(S) = \nu(S \cup r) - \nu(S)$. Aquí δ es la cota superior de la cardinalidad de S, $|\Pi_\delta|$ da el número total de subconjuntos de $F \setminus r$ acotados por δ. Esto significa que $\{\Pi_g\} \in S$, y g es la cardinalidad de un subconjunto de características con $g = 1, 2, \ldots, \delta$.

Si consideramos dos características r y t como jugadores, podemos calcular su interdependencia usando (3.2) donde C representa la clase binaria 1 (AMR ó clase positiva) y -1 (no-AMR ó clase negativa):

$$\tau(r,t) = \begin{cases} 1 & , \text{ si } I\left(f_t; C \mid f_r\right) > I\left(f_t; C\right) \\ 0 & , \text{en otro caso.} \end{cases} \tag{3.2}$$

Podemos formular $\Delta_r(S)$ como

$$\Delta_r(S) = \begin{cases} 1 & , \text{ si } I\left(S; C \mid f_r\right) \geq 0, \sum_{f_t \in S} \tau(r,t) \geq \frac{|S|}{2} \\ 0 & , \text{ en otro caso.} \end{cases} \tag{3.3}$$

4. Aplicación

Recolección de datos. En el artículo se menciona que las secuencias de aminoácidos para los genes de resistencia a antimicrobianos fueron obtenidas de la Base de Datos de Genes de Resistencia a Antibióticos (ARDB), y las secuencias no resistentes a AMR se obtuvieron de PATRIC. Realizaron una búsqueda BLASTp utilizando la configuración

predeterminada para encontrar todas las secuencias coincidentes. Las secuencias iniciales de AMR para las bacterias Gram-negativas *Acinetobacter* spp., *Klebsiella* spp., *Campylobacter* spp., *Salmonella* spp., y *Escherichia* spp. fueron 387 para *acetyltransferase* aac, 1113 para *β-lactamase* bla y 804 para *dihydrofolate reductase* dfr; hubo 159 secuencias no AMR (73 genes esenciales y 86 histonas acetiltransferasas) elegidas al azar.

Adaptamos el algoritmo (el script original se encuentra en R) en Python para replicar la clasificación binaria (AMR+ y AMR-). Usamos una coalición de 3 y le pedimos al algoritmo que escoja 10 características de un total de 621, combinado con una maquina de soporte vectorial con kernel Gaussiano obtuvimos un accuracy de 0.83.

Cuadro 1: Reporte de clasificación AMR+ y AMR-

Report	precision	recall	f1-score	support
AMR-	0.89	0.77	0.87	13
AMR+	0.75	0.87	0.86	9
accuracy			0.83	22
macro avg	0.88	0.88	0.86	22
weighted avg	0.90	0.96	0.86	22

Utilizamos los datos de CAMDA 2024 correspondientes a la resistencia a antibióticos. Consideramos dos tablas que contienen recuentos de SNPs de resistencia antimicrobiana (AMR) para diferentes aislamientos bacterianos **Resistance SNP count tables**. Ambas tablas contienen las siguientes columnas de metadatos: **accession, genus, species y antibiotic.** El objetivo de predicción es la columna **phenotype**, esta característica puede tener uno de dos valores: Resistente o Susceptible, nosotros la codificamos como -1 y 1, respectivamente. El segundo objetivo de predicción se encuentra en la columna llamada **measurement_value**, almacena valores numéricos positivos que representan la concentración mínima inhibitoria. Contiene recuentos de SNPs que confieren resistencia a antibióticos. Las columnas están etiquetadas usando identificadores ARO seguidos de la sustitución de aminoácidos.

Las tablas de **Resistance gene count tables** contienen el conteo de genes AMR para diferentes aislados bacterianos. Incluyen metadatos de la muestra (número de acceso, género, especie, fenotipo, antibiótico y valor de medición) y recuentos de genes AMR etiquetados con identificadores ARO. Durante la semana del Hackatón en el grupo de resistencia se obtuvieron nuevas tablas tomando en cuenta el mecanismo de resistencia a los antibióticos (meropenem-ciprofloxacin) y el recuento de genes asociado con los indentificadores ARO. El resultado de este cruce de tablas son los datos filtrados **ResistanceJoinedLooseBiofiltered** y **ResistanceJoinedStrictBiofiltered**. En particular trabajamos en los datos filtrados **ResistanceJoinedStrictBiofiltered**. Tomamos una coalición de 3 e indicamos al algoritmo que seleccione 10 y posteriormente 20 características más relevantes de un total de 348, los resultados se pueden observar en la cuadros 2 y 3 respectivamente, y en la Figura 2 vemos la matriz de confusión para 10 características, en la Figura 3 tenemos la correspondiente matriz de confusión para 20 características.

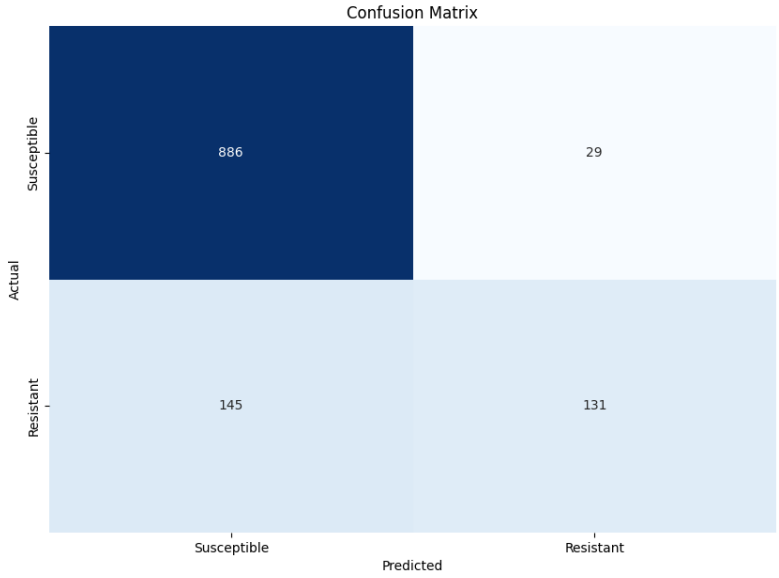


Figura 2: Matriz de confusión datos CAMDA 2024 para 10 características

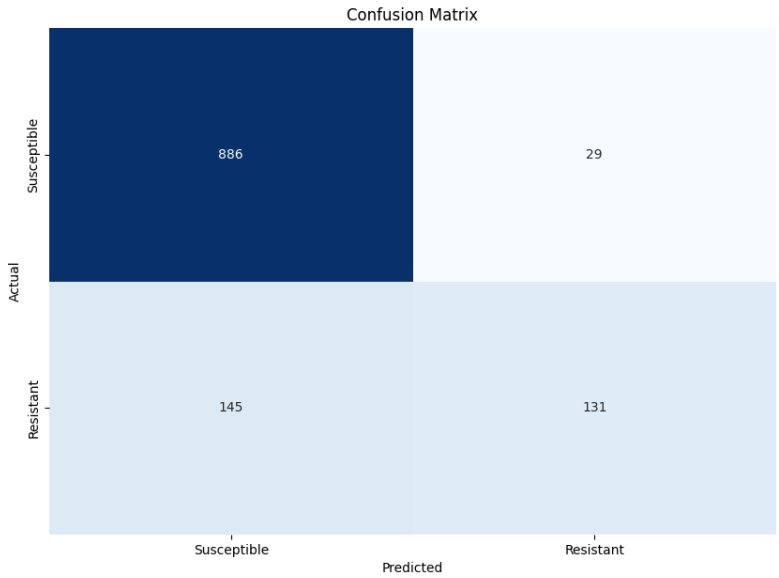


Figura 3: Matriz de confusión datos CAMDA 2024 para 20 características

Cuadro 2: Reporte de clasificación Resistente y Susceptible

Report	precision	recall	f1-score	support
Susceptible	0.85	0.97	0.91	915
Resistente	0.80	0.45	0.47	276
accuracy			0.81	1191
macro avg	0.83	0.71	0.74	1191
weighted avg	0.84	0.85	0.82	1191

Cuadro 3: Reporte de clasificación Resistente y Susceptible

Report	precision	recall	f1-score	support
Susceptible	0.86	0.97	0.91	915
Resistente	0.82	0.47	0.60	276
accuracy			0.85	1191
macro avg	0.84	0.72	0.76	1191
weighted avg	0.85	0.85	0.84	1191

5. Script

El código usando para generar los resultados que se observan en las tablas y gráficas está en el siguiente enlace `https://github.com/ccm-bioinfo/Camda24_resistance/blob/main/Scripts/preprocessing/GTDWFE.py`

Referencias

[Chowdhury et al, 2019] Chowdhury, A. S., Call, D. R., & Broschat, S. L. (2019). Antimicrobial Resistance Prediction for Gram-Negative Bacteria via Game Theory-Based Feature Evaluation. Scientific Reports, 9(1). doi:10.1038/s41598-019-50686-z.