



Modelos Gráficos Probabilísticos - 2024-2025

MASTER CIENCIA DE DATOS

UNIVERSIDAD DE GRANADA

Práctica 4

MIGUEL GARCÍA LÓPEZ

Índice

1. Introducción	3
2. Problema	3
3. Redes	3
3.1. Hill-Climbing con <i>Score</i> K2 (<code>hc_k2</code>)	4
3.2. Hill-Climbing con <i>Score</i> BIC (<code>hc_bic</code>)	4
3.3. Hill-Climbing con <i>Score</i> BDE (<code>hc_bde</code>)	5
3.4. Tabu Search (<code>tabu</code>)	5
3.5. Incremental Association Markov Blanket (IAMB) (<code>iamb</code>)	5
3.6. Max-Min Hill-Climbing (MMHC) (<code>mmhc</code>)	5
3.7. Restricted Maximization 2 (RSmax2) (<code>rsmax2</code>)	6
4. Pre-procesado	6
4.1. Selección de características con FSelector: Ganancia de información . .	6
5. Resultados	7
5.1. Sin selección de características	7
5.1.1. Estructuras obtenidas	9
5.2. Con selección de catacterísticas	11
5.2.1. Estructuras obtenidas	11

Índice de figuras

1. Correlación de variables en ledLXMn10.	4
2. Matrices de confusión del top.	8

3.	IAMB	9
4.	HC_BDE	10
5.	HC_K2	10
6.	Matrices de confusión del top (filtro IG).	12
7.	IAMB (filtro IG)	12
8.	HC_BDE (filtro IG)	13
9.	HC_K2 (filtro IG)	13

Índice de cuadros

1.	Top tres mejores algoritmos de aprendizaje de estructuras.	7
2.	Métricas del top 3 de algoritmos (HC_K2, IAMB, HC_BDE)	9
3.	Top tres mejores algoritmos de aprendizaje de estructuras (filtro IG). . .	11

1. Introducción

El objetivo de esta práctica es el de entrenar varias redes bayesianas para ajustar los datos del conjunto de datos *ledLXMn10.arff*. Previamente se deberán pre-procesar los datos, si fuese necesario, para poder conseguir un clasificador basado en redes bayesianas de la máxima calidad.

Se escogerá un sub-conjunto de las tres mejores redes obtenidas y se mostrarán los resultados en las métricas de referencia, las redes obtenidas y métricas. Se ha utilizado la restricción de *blacklist* para que la clase V25 (la clase objetivo) no pueda generar arcos a otras variables, es un conocimiento a priori válido.

2. Problema

El conjunto de datos es *ledLXMn10* el cual ha sido generado con el generador de flujo de datos LED (*LEDGenerator*). Este generador simula la representación de dígitos en un display de siete segmentos, donde el objetivo principal es predecir correctamente el número que está siendo representado.

Este conjunto de datos se caracteriza por tener un total de 24 atributos binarios, de los cuales únicamente 7 son relevantes para la tarea de clasificación, y los 17 restantes corresponden a atributos irrelevantes que añaden ruido y dificultan la tarea de aprendizaje.

Además, se introduce un porcentaje de ruido, concretamente del 10%, mediante la inversión aleatoria de los valores de los atributos, con el fin de simular condiciones de datos ruidosos y evaluar la robustez de los clasificadores.

Como puede verse en 1, las variables ruidosas presentan una correlación de 0 con respecto al resto de variables, solo la variable objetivo así como las 7 variables presentan coeficientes de relación distintos de cero, por lo que esta gráfica podría servir en un inicio para darse cuenta de esto.

El conjunto de datos presenta una tasa de clasificación Bayesiana óptima del 74%, lo que implica que incluso un clasificador ideal tendría un margen de error debido a la naturaleza ruidosa de los datos.

3. Redes

Para el aprendizaje de la estructura de las redes bayesianas se han empleado diversos algoritmos implementados en el paquete *bnlearn* de R. Estos algoritmos permiten la

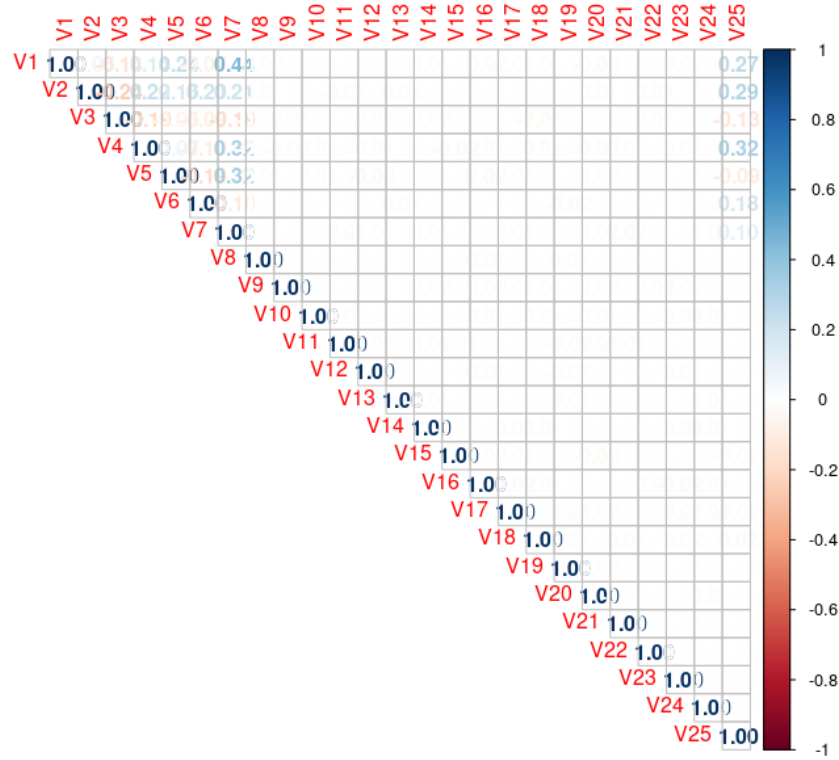


Figura 1: Correlación de variables en ledLXMn10.

identificación de la estructura subyacente que conecta las variables del conjunto de datos. A continuación, se describen detalladamente los algoritmos utilizados:

3.1. Hill-Climbing con *Score* K2 (hc_k2)

El algoritmo *Hill-Climbing* (HC) es un método de búsqueda heurística que explora el espacio de estructuras posibles mediante movimientos locales (adición, eliminación o inversión de aristas) con el objetivo de maximizar una función de puntuación. En este caso, se ha utilizado la puntuación *K2*, la cual está basada en la probabilidad a posteriori de la estructura dada una muestra de datos, asumiendo independencia a priori entre los nodos y utilizando una estimación de máxima verosimilitud.

3.2. Hill-Climbing con *Score* BIC (hc_bic)

Este algoritmo sigue el mismo procedimiento de búsqueda local que *Hill-Climbing*, pero utilizando como criterio de evaluación el **Bayesian Information Criterion** (BIC). Este criterio introduce una penalización por el número de parámetros del modelo, lo que favorece la obtención de redes más simples y generalizables. Al balancear la bondad de ajuste con la complejidad de la red, BIC busca evitar el sobreajuste, seleccionando redes

bayesianas que expliquen adecuadamente los datos sin añadir relaciones innecesarias.

3.3. Hill-Climbing con *Score* BDE (hc_bde)

En este caso, el algoritmo *Hill-Climbing* emplea la puntuación **Bayesian Dirichlet equivalent** (*BDE*). Esta puntuación corresponde a una familia de puntuaciones bayesianas que incorporan tanto la información de los datos como una distribución a priori sobre las redes.

3.4. Tabu Search (tabu)

El algoritmo **Tabu Search** es una técnica metaheurística que mejora el proceso de búsqueda local del *Hill-Climbing* mediante la incorporación de una memoria a corto plazo conocida como “lista tabú”. Esta lista contiene las últimas modificaciones realizadas a la estructura, prohibiendo deshacerlas durante un número determinado de iteraciones, lo que permite al algoritmo escapar de óptimos locales y explorar zonas más amplias del espacio de soluciones. *Tabu Search* es especialmente útil en problemas donde la superficie de la función objetivo presenta múltiples óptimos locales.

3.5. Incremental Association Markov Blanket (IAMB) (iamb)

El algoritmo **IAMB** se basa en la construcción del *Markov Blanket* de cada variable, es decir, el conjunto mínimo de variables que, condicionadas, hacen independiente a la variable objetivo del resto. *IAMB* opera mediante una fase de adición, en la cual se incorporan variables que tienen una fuerte dependencia condicional con la variable objetivo, y una fase de eliminación, donde se descartan variables redundantes. Este enfoque permite identificar las relaciones de dependencia directa entre variables de forma eficiente, especialmente útil en conjuntos de datos de alta dimensionalidad. Posteriormente, se ha aplicado la extensión **cextend** para garantizar que la red obtenida sea completa y consistente.

3.6. Max-Min Hill-Climbing (MMHC) (mmhc)

MMHC es un algoritmo híbrido que combina dos fases: una primera fase basada en pruebas de independencia condicional (*Max-Min Parents and Children*) para obtener una red inicial de adyacencias, y una segunda fase en la que se aplica el algoritmo *Hill-Climbing* para determinar la orientación de las aristas, respetando las relaciones de dependencia descubiertas. Esta combinación permite a *MMHC* escalar mejor a grandes conjuntos

de datos, reduciendo el espacio de búsqueda a partir de las relaciones más relevantes identificadas en la primera fase.

3.7. Restricted Maximization 2 (RSmax2) (rsmax2)

El algoritmo **RSmax2** es otro enfoque híbrido que también divide el aprendizaje de la estructura en dos etapas. En la primera fase, al igual que *MMHC*, se realiza una búsqueda restringida del conjunto de adyacencias mediante pruebas de independencia. En la segunda fase, se lleva a cabo una búsqueda orientada por puntuación (como *Hill-Climbing*) para definir la dirección de las relaciones, pero limitada al grafo descubierto inicialmente. Este enfoque permite balancear la eficiencia computacional con la calidad de la estructura aprendida, al reducir el número de candidatos a evaluar en la fase de optimización.

4. Pre-procesado

Para el pre-procesado se han convertido las variables a tipo **factor**, ya que los algoritmos de aprendizaje no son capaces de procesar el tipo de dato base en el que vienen descritas las variables. Cada variable toma dos posibles valores, 0 o 1 y la variable objetivo puede tomar valores desde 1 a 9. Dada la descripción del problema, se conoce que hay 17 variables totalmente inútiles, ya que aportan solo ruido, por lo que una selección de características previas podría ser interesante.

En este caso es interesante eliminar características irrelevantes, ya que se conoce que existen variables totalmente ruidosas. El propósito de esto es facilitar la tarea de aprendizaje del clasificador bayesiano, no es totalmente obligatorio, pues la red podría inferir una estructura válida a partir de los datos, ignorando enlaces en aquellas variables que no son necesarias. Por esta razón se compararán las métricas y las estructuras de las top 3 redes con mejores valores para la métrica *accuracy*.

Además se realiza un particionamiento de los datos en dos conjuntos, entrenamiento y prueba (*train* y *test*). La idea es entrenar y encontrar la estructura dado un sub-conjunto y validar los resultados sobre datos nunca vistos con *test*.

4.1. Selección de características con FSelector: Ganancia de información

FSelector es un paquete de R que implementa métodos de selección de características (*feature selection*). Concretamente se va a utilizar el método de **ganancia de información**.

La ganancia de información es una métrica proveniente de la teoría de la información de *Shannon*. Se utiliza principalmente para seleccionar las mejores variables predictoras en tareas de clasificación. La idea de *IG* (*information gain*) es medir cuánto conocimiento sobre la variable objetivo (*target*) se obtiene al conocer el valor de una variable (*feature*). Es decir, cuánto se reduce la incertidumbre sobre el *target* cuando se conoce esa *feature*.

Formalmente, se basa en la entropía.

$$H(Y|X) = \sum_{j=1}^m P(x_j) H(Y|X = x_j)$$

La ganancia de información es la reducción de incertidumbre (entropía) sobre Y al conocer X :

$$IG(Y, X) = H(Y) - H(Y|X)$$

Utilizando la función `information.gain` calcula la ganancia de información de todas las variables independientes respecto a la variable objetivo (V25).

5. Resultados

Tras la construcción de las estructuras de las redes y el ajuste de las mismas por medios de los datos para el aprendizaje de los parámetros, se procede a analizar y exponer los resultados de las tres redes con mejores resultados de tasa de acierto en clasificación.

5.1. Sin selección de características

Algorithm	Accuracy
HC_K2	0.727
IAMB	0.725
HC_BDE	0.719

Cuadro 1: Top tres mejores algoritmos de aprendizaje de estructuras.

Como puede observarse en la tabla 3, los las redes con mejores tasas de *accuracy* son `hc_k2`, `hc_bde` e `iamb`. En la figura 2 se muestran las matrices de confusión para cada red, siendo todas muy buenas. De hecho, se puede observar como las redes se confunden con frecuencia entre los mismos dígitos, indicando que hay cierta dificultad en esas partes concretas.

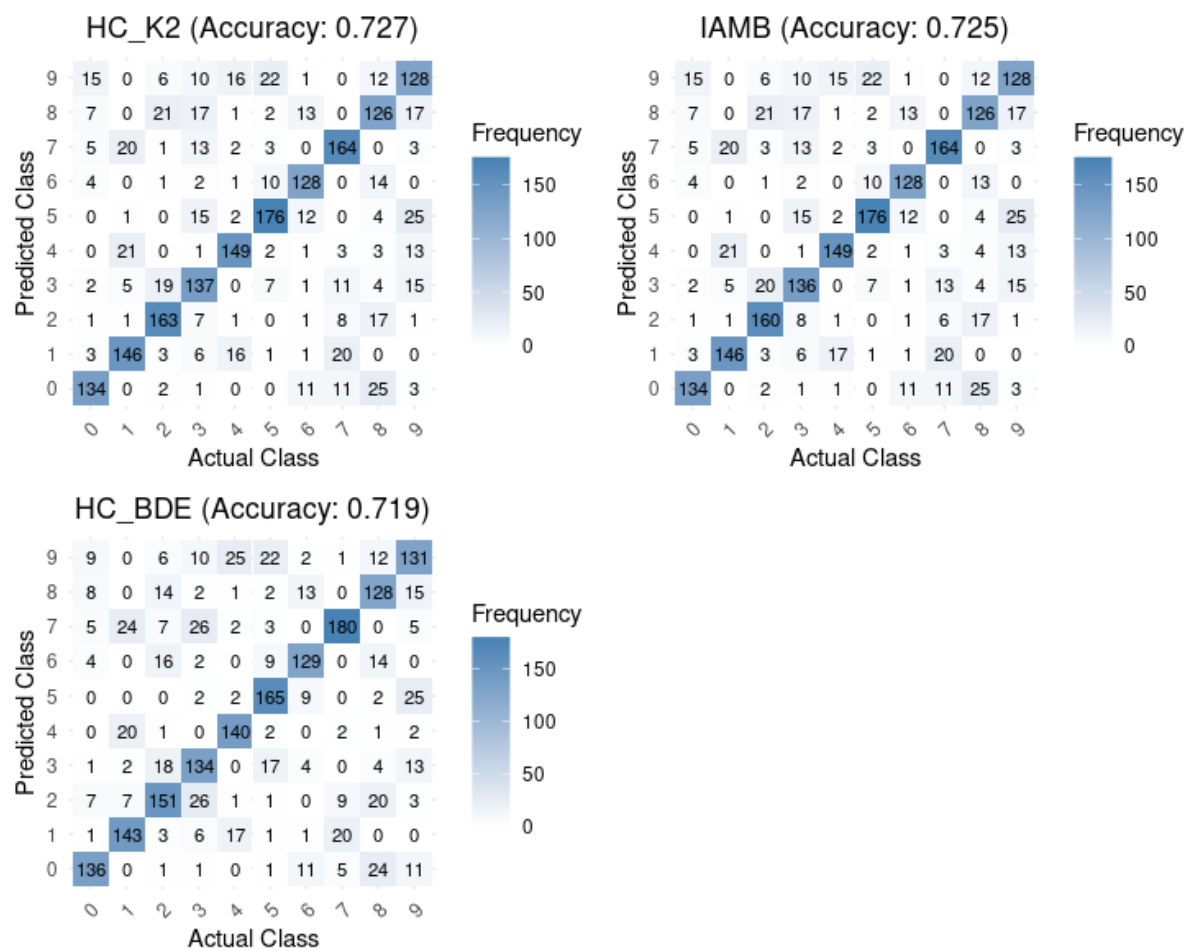


Figura 2: Matrices de confusión del top.

Este entrenamiento se ha llevado a cabo sin seleccionar características para disminuir el tamaño de las variables y ayudar a los algoritmos de aprendizaje, pero pese a eso se han alcanzado tasas de *accuracy* muy cercanos al límite teórico que indicaba la documentación del generador de datos LED, que en teoría era de un 74 %.

En la tabla 2 se pueden ver las métricas de *recall*, *f1-score* y *precision* desglosadas por cada clase.

	Precision			Recall			F1-Score		
	HC_K2	IAMB	HC_BDE	HC_K2	IAMB	HC_BDE	HC_K2	IAMB	HC_BDE
0	0.717	0.713	0.716	0.784	0.784	0.795	0.749	0.747	0.753
1	0.745	0.741	0.745	0.753	0.753	0.730	0.749	0.747	0.737
2	0.815	0.816	0.671	0.755	0.741	0.696	0.784	0.777	0.683
3	0.682	0.670	0.694	0.656	0.651	0.641	0.668	0.660	0.667
4	0.772	0.768	0.833	0.793	0.793	0.745	0.782	0.780	0.787
5	0.749	0.749	0.805	0.789	0.789	0.740	0.769	0.769	0.771
6	0.800	0.810	0.741	0.757	0.757	0.763	0.778	0.783	0.752
7	0.777	0.770	0.714	0.756	0.756	0.829	0.766	0.763	0.768
8	0.618	0.618	0.699	0.615	0.615	0.624	0.616	0.616	0.660
9	0.610	0.612	0.601	0.624	0.624	0.639	0.617	0.618	0.619

Cuadro 2: Métricas del top 3 de algoritmos (HC_K2, IAMB, HC_BDE)

5.1.1. Estructuras obtenidas

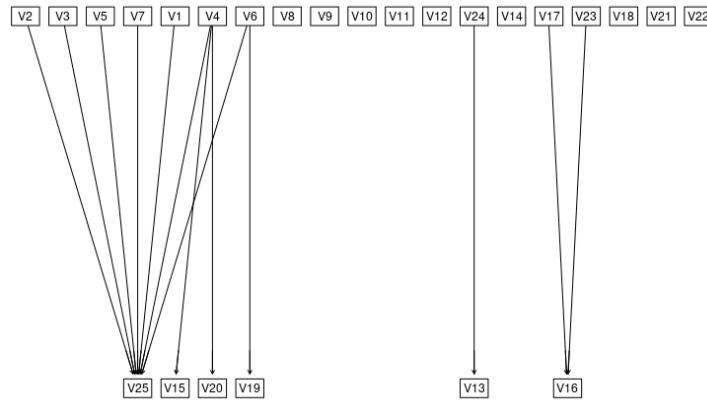


Figura 3: IAMB

Como puede observarse, en las tres redes (3, 4, 5) tienen muchísimas variables sin conectar con el resto de nodos, esto es porque corresponden a variables totalmente rui-

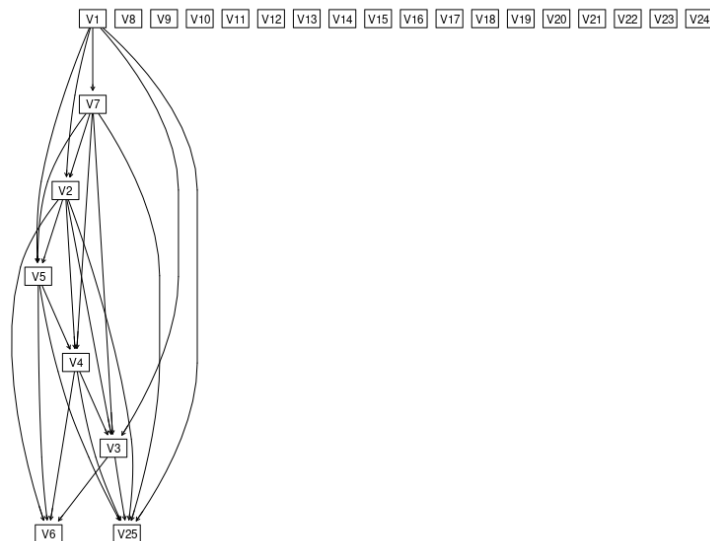


Figura 4: HC_BDE

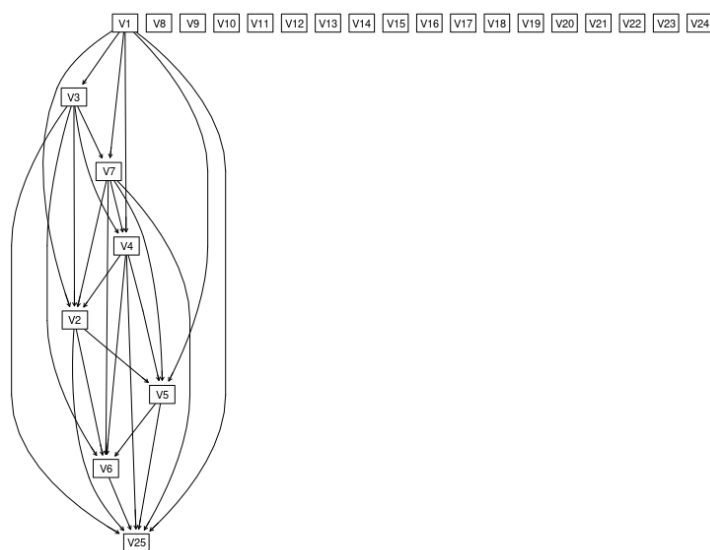


Figura 5: HC_K2

dosas y el algoritmo de cada una de las redes ha sido capaz de identificar que no aportan nada. Las redes correspondientes a los algoritmos hc presentan mayor cantidad de arcos en relación a la red del algoritmos de test de independencia.

5.2. Con selección de catacterísticas

Ahora se aplica un filtrado de características inicial para quitar aquellas que, bajo el criterio de ganancia de información, no aportan. Para ello se intenta eliminar 17, que serían, idealmente, aquellas 17 que fueron generadas con la finalidad de aportar ruido.

Algorithm	Accuracy
HC_K2	0.727
IAMB	0.725
HC_BDE	0.719

Cuadro 3: Top tres mejores algoritmos de aprendizaje de estructuras (filtro IG).

Los resultados son en términos de *accuracy* (table 6 y tabla 6) exáctamente iguales dado un filtrado previo de las variables ruidosas, de hecho cabe destacar que el filtro de ganancia de información ha conseguido identificar perfectamente a todas ellas. Dados estos resultados, se considera redundant añadir tabla de métricas adicionales.

5.2.1. Estructuras obtenidas

Como puede observarse (figuras 7, 8, 9), las estructuras son mucho más limpias al no contener las variables ruidosas, pero obviando ese detalle, son exactamente iguales que las obtenidas anteriormente, todas ellas presentan los mismos arcos entre las mismas variables.

En este experimento, parece que la selección de características no ha aportado nada a la mejora de las redes.

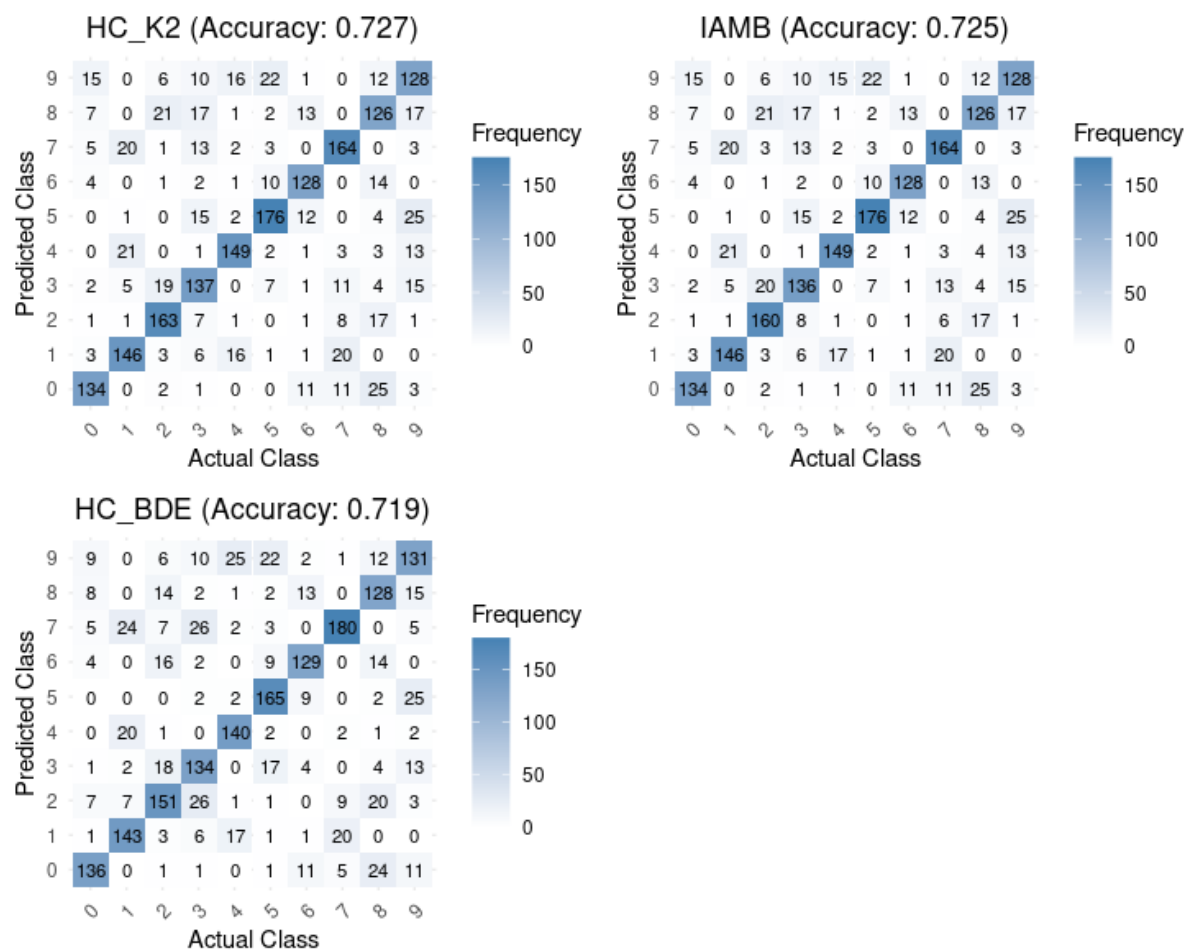


Figura 6: Matrices de confusión del top (filtro IG).

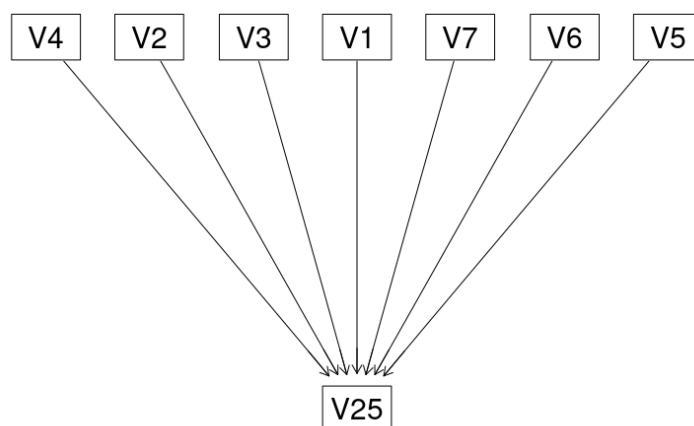


Figura 7: IAMB (filtro IG)

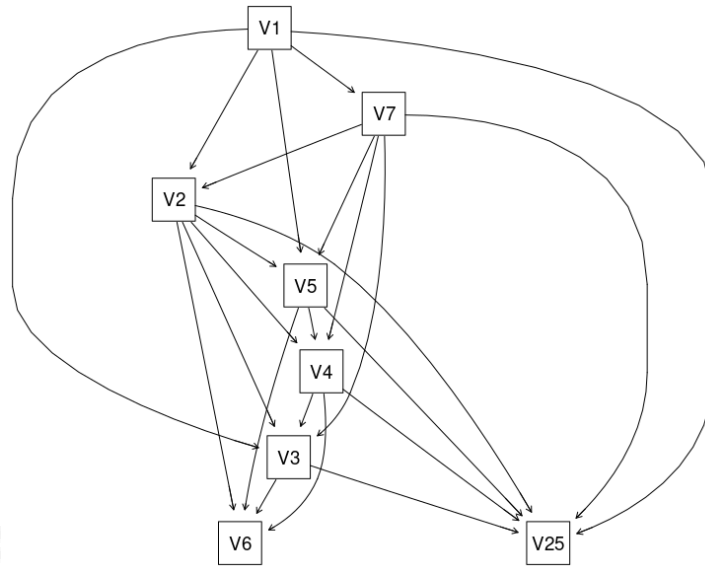


Figura 8: HC_BDE (filtro IG)

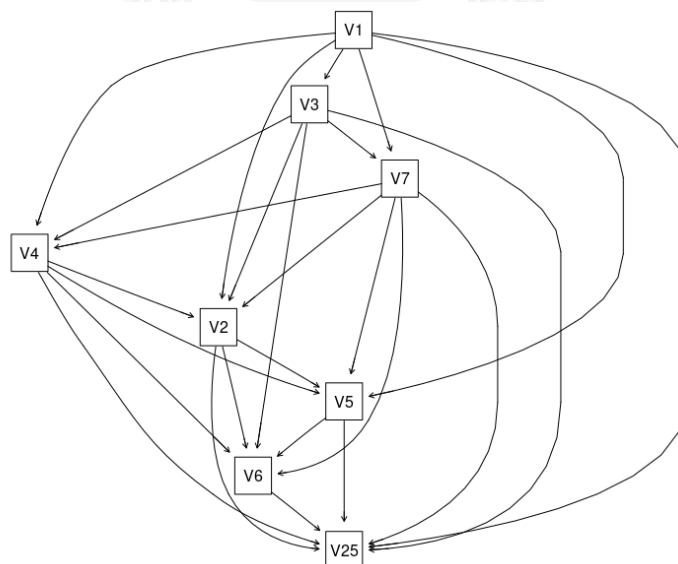


Figura 9: HC_K2 (filtro IG)