Selección de variables

García Veiga, Mariam (USC)
Lado González, Ignacio (USC)
Leyenda Rodríguez, María (USC)
López Veiga, David (USC)

27 de Noviembre de 2009

ÍNDICE

1. INTRODUCCIÓN

La selección de variables es un problema muy estudiado, aunque fundamentalmente abierto. Las fuertes interacciones entre las variables y la presencia de variables irrelevantes, redundantes, el ruido en la muestra, etc., dificultan aún más el problema.

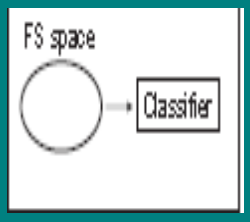
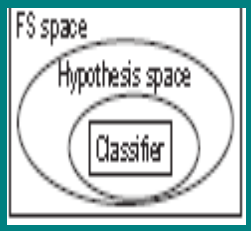
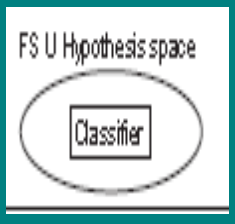
Los principales objetivos del problema de selección de variables son:

- Evitar el sobreajuste y mejorar el modelo obtenido, es decir, la predicción obtenida en caso de clasificación supervisada y mejor selección de variables ("clúster") en caso de que se quieran seleccionar variables.
- Proporcionar modelos más rápidos y con más coste-eficacia
- Obtener más profundidad en los procesos subyacentes que en el conjunto de datos generado.

Sin embargo, las ventajas de las técnicas de selección de variables tienen un cierto precio, por ejemplo, la búsqueda de un conjunto de variables relevantes introduce complejidad en la tarea de modelar un problema. En vez de solo optimizar los parámetros de todo el conjunto de datos, ahora encontraremos también los parámetros óptimos del modelo para el subconjunto óptimo de variables, porque no hay garantía de que los parámetros óptimos para todo el conjunto de variables sean igualmente óptimos para el subconjunto de óptimo de variables. Por tanto, la búsqueda del espacio de hipótesis del modelo aumenta en otra dimensión: encontrar el subconjunto óptimo de variables relevantes. Las técnicas de selección de variables se diferencian unas de otras por la manera en que ellas incorporan esta búsqueda del subconjunto de variables en el modelo de selección.

En el contexto de clasificación, las técnicas de clasificación de variables pueden ser organizadas en tres categorías, dependiendo de cómo combinan la selección de variables con la construcción del modelo de clasificación: *métodos filter* , *métodos wraper* y *métodos embedded.*

Modelos de búsqueda	
FILTER	Calculan la relevancia de las variables mirando solo las propiedades intrínsecas de los datos. En la mayoría de los casos, se calcula la relevancia de cada una de las variables y las variables con menor relevancia son eliminadas.
WRAPPER	Es definido el subconjunto de posibles variables en el espacio de búsqueda de un procedimiento, y varios subconjuntos de variables son generadas y evaluadas. La evaluación de un subconjunto especifico de variables es obtenido intentando y testeando un modelo de clasificación especifico, interpretando esta aproximación especificaremos el algoritmo de clasificación. La búsqueda de todos los subconjuntos de variables, es un algoritmo de búsqueda "wrapped" alrededor de la clasificación del modelo. Aunque, como el espacio de subconjuntos de variables crece exponencialmente con el número de variables, los métodos de búsqueda heurística son usados para guiar la búsqueda del subconjunto óptimo.
EMBEDDED	El subconjunto óptimo de variable es obtenido en la construcción del clasificador, y puede ser visto como una búsqueda en la combinación del espacio de los subconjuntos de características y hipótesis.

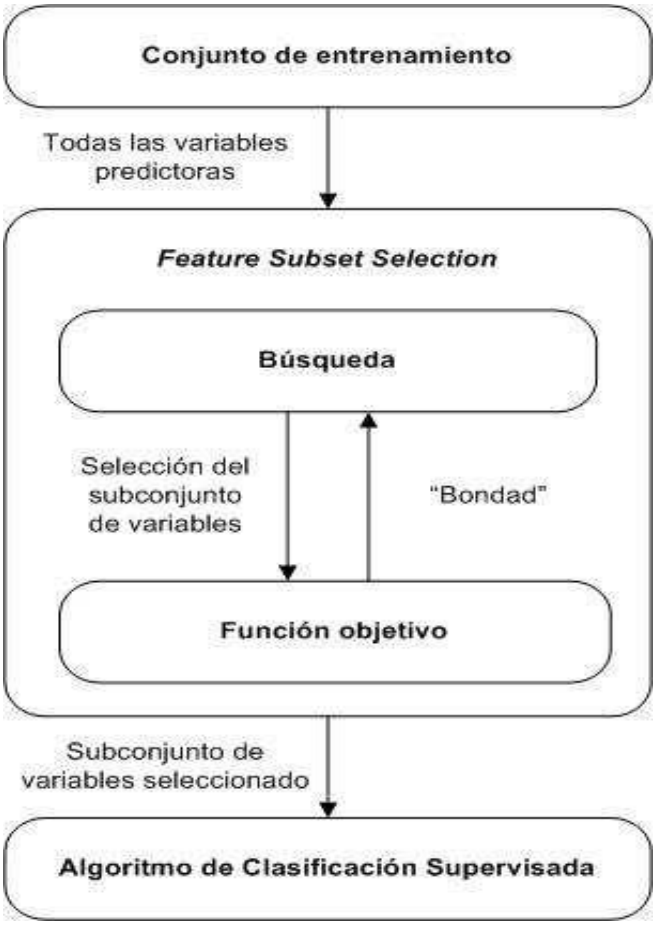
Modelo de búsqueda			Ventajas	Desventajas	Ejemplos
Filter		Univariante	Rápido Escalable Independiente del clasificador	Ignora las dependencias entre las variables Ignora la interacción con el clasificador	Chi-square Distancia euclidea t-test Information gain Gain ratio
		Multivariante	Modelos de variables dependientes Independiente del clasificador Computacionalmente es mejor que los métodos wapper	Más lento que las técnicas univariantesion Menos escalable que las técnicas univariantes. Ignora la interacción con el clasificador	selección de variables basada en correlación (CFS) Markov blanket filter (MBF) selección de variables basada en correlación rápida (FCBF)
Wrapper		Determinista	Simple Interactúa con el clasificador Modelos de variables dependientes Computacionalmente menos intensivo que los métodos aleatorizados.	Riesgo de sobreajuste Más propenso que los algoritmos aleatoriezados a quedarse atascados en un optimo local. (greedy search) El clasificador depende de la selección	Sequencial forward selection (CFS) Sequencial backward elimination (SBE) Añadir q quitar r Beam search
		Aleatorio	Menos propenso a la estimación local Interacciones con el clasificador Modelos de variables dependientes	Intensivo computacionalmente Clasificador depende de la selección El riesgo de sobreajuste es más elevado que en los algoritmos deterministas	Estimación de distribución de los algoritmos. Algoritmos genéticos
Embedded		Interacciona con el clasificador Computacionalmente es mejor que los métodos wapper Modelos de variables dependientes		El clasificador depende de la selección.	Arboles de decisión Selección de variables usando el peso del vector de SVM

Vamos a estudiar el problema de selección de variables a través de algoritmos de selección de variables. El problema de desarrollar un ASV es básicamente uno de búsqueda en un espacio de estados. Cada estado representa un subconjunto de variables ponderadas; el objetivo es encontrar el estado con la mejor medida de evaluación. El número de subconjuntos potenciales a evaluar es 2n en caso que la ponderación sea binaria.

Existen 2 tipos de ASV:

- Algoritmos que proporcionan un orden lineal de las variables (ponderación continua).
- Algoritmos que obtienen un subconjunto del conjunto original (ponderación binaria).

Esquema del procedimiento de selección de subconjuntos de variables para problemas de clasificación supervisada



El sistema Weka incorpora una gran cantidad de métodos para estudiar la **relevancia de atributos** y realizar una **selección automática de los mismos**. Estos métodos, están dentro de la entorno *Explorer* en la sección *Select Attributes*. Esta sección permite automatizar la búsqueda de subconjuntos de atributos más apropiados para "explicar" un atributo objetivo, en un sentido de clasificación supervisada: permite explorar qué subconjuntos de atributos son los que mejor pueden clasificar la clase de la instancia.

La selección supervisada de atributos tiene dos componentes:

Método de búsqueda(Search Method): es la forma de realizar la búsqueda de conjuntos. Como la evaluación exhaustiva de todos los subconjuntos es un problema combinatorio inabordable en cuanto crece el número de atributos, aparecen estrategias que permiten realizar la búsqueda de forma eficiente

"**SubSetEval"**: Evaluadores de conjuntos o selectores. Estos necesitan elegir un método o estrategia de búsqueda de los subconjuntos .

Método de Evaluación (**Attribute Evaluator):** es la función que determina la calidad del conjunto de atributos para discriminar la clase.

"AttributeEval": Porteadores de atributos. Estos solo pueden combinarse con un "Ranker" ya que no seleccionan atributos sino que solo los ordenan por relevancia.

Dentro los método de evaluación podemos distinguir dos tipos: Los métodos que directamente utilizan un clasificador específico para medir la calidad del subconjunto de atributos a través de la tasa de error del clasificador (métodos "wraper") y los que no.

Métodos "wrapper", porque "envuelven" al clasificador para explorar la mejor selección de atributos que optimiza sus prestaciones, son muy costosos porque necesitan un proceso completo de entrenamiento y evaluación en cada paso de búsqueda.

Métodos como el método **"CfsSubsetEval",** que calcula la correlación de la clase con cada atributo, y eliminan atributos que tienen una correlación muy alta como atributos redundantes.

Hay diferentes métodos de búsqueda de las variables más influyentes, como son:

"**ForwardSelection",** que es un método de búsqueda muy rápido que subóptima en escalada, donde elije primero el mejor atributo, después añade el siguiente atributo que más aporta y continua así hasta llegar a la situación en la que añadir un nuevo atributo empeora la situación.

"**BestSearch",** que permite buscar interacciones entre atributos más complejas que el análisis incremental anterior. Este método va analizando lo que mejora y empeora un grupo de atributos al añadir elementos, con la posibilidad de hacer retrocesos para explorar con más detalle.

"ExhaustiveSearch" simplemente enumera todas las posibilidades y las evalúa para seleccionar la mejor.

Por otro lado, en la configuración del problema debemos seleccionar que atributo objetivo se utiliza para la selección supervisada, en la ventana de selección y determinar si la evaluación se realizará con todas las instancias disponibles o mediante validación cruzada.

Vamos a estudiar la selección de atributos utilizando la herramienta Weka. Para ello vamos a usar dos conjuntos de datos:"Encuesta de accidentes.xls" y "credit.xls".

2. ALGORITMOS DE PONDERACIÓN BINARIA

2.1. CFS SUBSET EVAL

Evalúa el valor de un subconjunto de atributos, considerando la capacidad de predicción individual de cada función, junto con el grado de redundancia entre ellos.

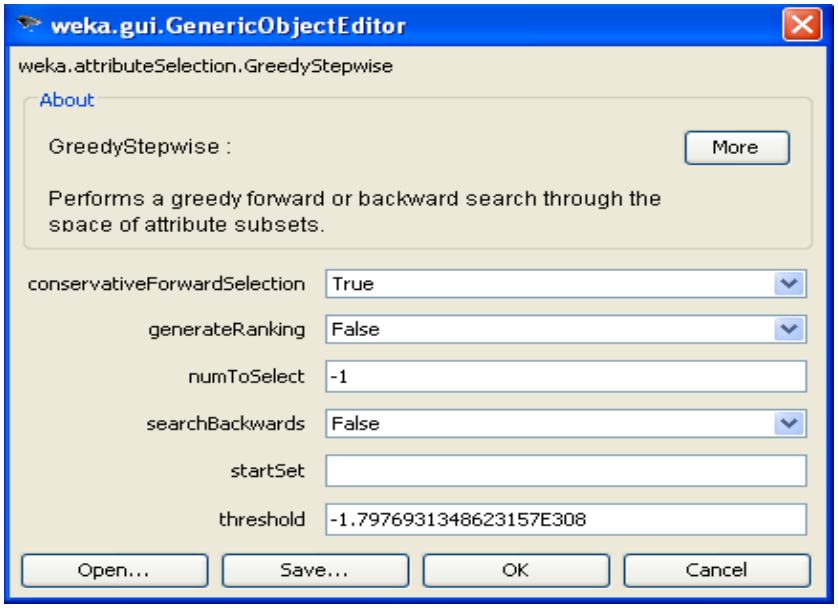
Elimina atributos que tienen una correlación muy alta como atributos redundantes.

Opciones:

- **Locally Predictive:**
  Identifica los atributos de predicción local. Iterativamente añade atributos con la mayor correlación con respecto a la clase siempre y cuando no haya ya un atributo en el subconjunto con una alta correlación con el atributo en cuestión.
- **Missing Separate:**
  Trata los valores perdidos como valores independientes. De lo contrario el recuento de los valores perdidos se distribuye a través de otros valores en proporción a su frecuencia

2.1.1. Método de búsqueda:

Funciona como un método de búsqueda rápido, hacia delante o hacia atrás, en función del espacio de subconjuntos de atributos. Puede comenzar con todos/ningún atributos o desde un punto arbitrario en el espacio. Se para cuando la adición/eliminación de cualquier atributo causa una disminución en la evaluación. También puede producir una lista ordenada de atributos atravesando el espacio de un lado al otro, registrando el orden de los atributos son seleccionados.



OPCIONES:

- **ConservativeForwardSelection** – Si marcamos con true esta opción, se irán añadiendo atributos uno a uno hasta obtener el mejor subconjunto posible de variables, siempre que el mérito no decrezca.
- **GenerateRanking** – Es opcional. Si se marca como true, lista un raking de atributos.
- **NumToSelect**- Especifica el número de atributos a retener. Por defecto viene el valor -1 que indica que todos los atributos son retenidos. Usa otra opción o un umbral para reducir la colección de atributos.
- **SearchBackwards** – Si marcamos con true esta opción, utilizará un método de búsqueda hacia atrás.
- **StarSet** – Establece un punto de partida para la búsqueda. Los atributos se separan por comas
- **Threshold –** Se fija el umbral por el cúal se pueden descartar atributos. El valor que viene dado por defecto no descarta ningún atributo. Se usa esta opción o numToSelect para reducir la colección de atributos.

2.2. CLASSIFIER SUBSET EVAL

Evalúa subconjuntos de atributos en los datos de entrenamiento.Utiliza un clasificador para estimar el "mérito" de un conjunto de atributos

2.2.1. Método de búsqueda, (RandomSearch):

Realiza una búsqueda aleatoria en el espacio de subconjuntos de atributos. Si no se proporciona ningún conjunto de inicio, la búsqueda aleatoria se inicia desde un punto de azar y los informes del mejor subconjunto encontrado.

Si se da un conjunto de partida, Random búsquedas al azar para subconjuntos que son tan buenos o mejores que el punto de partida con el mismo o menos, o atributos.

OPCIONES:

- **PearchPercent** –Porcentaje del espacio de búsqueda para explorar.
- **StartSet** - Establece el punto de partida para la búsqueda. Esto se especifica como una lista separada por comas a partir de 1. Puede incluir rangos. Por ejemplo. 1,2,5-9,17. Si se especificaron, registros aleatorios para subconjuntos de atributos que son tan buenos o mejores que el primer conjunto establecido, con la misma o menor cardinalidad.

- **Verbose** - Información sobre el progreso de impresión. Envía información de los avances de la terminal como la búsqueda avanza.

2.3. CONSISTENCY SUBSET EVAL

Evalúa el valor de un conjunto de atributos por el nivel de consistencia en la clase cuando los datos de entrenamiento son proyectados en el subconjunto de atributos.

2.3.1. Método de búsqueda, (Exhaustivesearch):

Realiza una búsqueda exhaustiva por el espacio de subconjuntos de atributos a partir del conjunto vacío de atributos. Informes del mejor subconjunto encontrado.

OPCIONES:

- **Verbose** - información sobre el progreso de impresión. Envía información de los avances de la terminal como la búsqueda avanza

2.4. COST SENSITIVE SUBSET EVAL

Se trata de un evaluador de conjuntos de atributos que establece el coste de dicho sunconjunto.

OPCIONES:

- **costMatrix –** Establece explícitamente cuál es el coste de la matriz. La matriz se utiliza si la propiedad *"costMatrixSource"* se proporciona "supplied".
- **costMatrixSource –** Indica donde conseguir la matriz de coste. Las dos opciones deben utilizar la matriz de coste proporcionada, o cargar una matriz de coste de un archivo cuando éste se requiera (este archivo se cargará desde el directorio establecido en la propiedad"onDemandDirectory" y se llamará "relation_name.cost").
- **evaluator** – Evaluador que va a ser utilizado.
- **onDemandDirectory –**Establece el directorio desde el que se cargan los archivos de coste. Se utiliza esta opción cuando el "costMatrixSource" se pone como"On Demand".
- **seed** – Semilla que utilizamos en el análisis.

2.4.1. Método de búsqueda, (Greedy Stepwise):

2.5. FILTERED SUBSET EVAL

Utilizado para controlar a un evaluador de un subconjuntos arbitrario de atributos sobre los datos a los que se le ha pasado el filtro (no se permiten filtros que alteran el orden o el número de atributos). Como el evaluador, la estructura del filtro está basada exclusivamente en los datos de entrenamiento

OPCIONES:

- **subsetEvaluator** – Se utiliza para seleccionar el evaluador de subconjuntos que se va usar.
- **filter**-- Determina el filtro utilizado en el análisis.

2.5.1. Método de búsqueda,(Greedy Stepwise):

2.6. WRAPPER SUBSET EVAL

El objetivo de la aproximación wrapper es seleccionar un subconjunto óptimo de variables o, en su defecto, un buen subconjunto de variables, que proporcionen el mejor conocimiento posible de la variable clase. A diferencia del enfoque filter, en el modelo wrapper el algoritmo de selección de subconjuntos de variables funciona como un envoltorio que rodea al algoritmo de inducción. Se trata pues de evaluar conjuntos de atributos utilizando un esquema de aprendizaje. El algoritmo utilizado para la selección de subconjuntos de variables lleva a cabo una búsqueda de subconjuntos de variables óptimos y, para ello, se sirve del algoritmo de inducción como una parte de la función que evalúa subconjuntos de variables.

El algoritmo de inducción se ocupa de extraer un clasificador. Éste ha de resultar útil en la tarea de clasificar nuevos casos. El algoritmo de inducción es aplicado al conjunto de los datos que, habitualmente, se encuentra particionado en diferentes subconjuntos de variables obtenidos de los datos. De esta aplicación se va a obtener el subconjunto de variables que presenta la mayor puntuación de evaluación. Y, sobre este subconjunto seleccionado, se procede a aplicar el algoritmo de inducción. Con ello se obtiene un clasificador. Siendo su objetivo y el último paso del modelo wrapper la obtención de la máxima precisión en la clasificación de un conjunto de test que no ha sido visto por el algoritmo con anterioridad.

En la aplicación de un modelo wrapper para selección de subconjuntos de variables se realiza una validación cruzada interna. El proceso de validación cruzada interna parte de una muestra de entrenamiento que se divide en varios subconjuntos de variables. En cada uno de estos subconjuntos se deja fuera una variable, y se aplica sobre ellos el algoritmo de inducción. Posteriormente se realiza una evaluación del clasificador obtenido con el algoritmo de inducción sobre la variable no vista por el mismo. Esto se conoce como fase de test, donde se valora la precisión de los clasificadores obtenidos.

Dos componentes importantes en el método wrapper son el espacio de búsqueda y el motor de búsqueda. El espacio de búsqueda está definido por elementos, cada uno de los cuales representa un subconjunto de variables. Si el conjunto original de los datos está formado por *n* variables, existen *n* bits en cada elemento que indican la presencia (1) o la ausencia (0) de cada variable en un elemento concreto. Al realizar una búsqueda sobre el espacio de búsqueda es necesario que los elementos del espacio se encuentren conectados entre sí. Esto se lleva a cabo empleando los operadores. Un ejemplo de operador es aquel que añade o elimina una variable de cada vez, permitiendo así el movimiento de un elemento a otro del espacio de búsqueda.

El motor de búsqueda es el algoritmo empleado como método para realizar la búsqueda entre los distintos subconjuntos de variables. Resulta importante la selección de un motor de búsqueda adecuado al problema que se está analizando. Atendiendo a que el método wrapper es computacionalmente más costoso que los métodos filter, especialmente cuando nos enfrentamos a conjuntos de datos de dimensionalidad muy elevada.

El algoritmo Wrapper Subset Eval en Weka presenta las siguientes opciones:

clasifier.- Permite seleccionar un clasificador que estime la precisión de los subconjuntos como resultado del algoritmo de inducción. Existe un gran número de clasificadores entre los que elegir, clasificados en las siguientes categorías: bayes, functions, lazy, meta, mi, misc, rules, scripting y trees. Al seleccionar cualquiera de los clasificadores disponibles es posible configurar las opciones del mismo.

evaluationMeasure.- Permite seleccionar las medidas empleadas en la evaluación de la representatividad de las combinaciones de variables que se utilizan en la tabla de decisión. Por defecto es la precisión para clases discretas y la RSME en clases continuas.

folds.- Permite configurar el número de grupos de validación cruzada que se van a utilizar en la estimación de la precisión de los subconjuntos.

seed.- Opción que permite determinar la semilla utilizada para generar cortes en la validación cruzada.

threshold.- Esta opción determina un umbral, que hace que se repita la validación cruzada si la desviación típica de la media supera este valor.

2.6.1. Método de búsqueda:

Un método de búsqueda que puede emplearse en un modelo wrapper de selección de subconjuntos de variables es el método Best First. Este método realiza una búsqueda en el espacio de subconjuntos con un "greedy hillclimbing" pero aumentando la facilidad de pasos hacia atrás. Puede configurarse el número de nodos consecutivos sin que se produzca un aumento permitiendo controlar el nivel retroceso. El algoritmo puede iniciarse con un conjunto de variables vacio y buscar hacia delante, empezar con un conjunto completo y buscar hacia atrás, o bien empezar en cualquier punto del espacio de búsqueda y buscar en ambas direcciones. La idea es encontrar el nodo más prometedor que se haya generado mientras que no haya sido mejorado. El menú opciones del método de búsqueda nos permite realizar algunas configuraciones:

- **direction.**- Permite decidir cual va a ser la dirección de la búsqueda, hacia delante, hacia atrás o en ambas direcciones.

- **lookupCacheSize**.- Esta opción sirve para configurar el tamaño máximo del caché de búsqueda de subconjuntos evaluados. Viene expresado como un multiplicador del número de variables en el conjunto de los datos.

- **searchTermination.**- Con esta opción puede especificarse el número de pasos hacia atrás permitidos al algoritmo antes de concluir la búsqueda.

- **startSet.**- Sí se desea elegir un punto concreto para el inicio de la búsqueda pude indicarse aquí.

2.7. SYMMETRICAL UNCERT ATTRIBUTE SET EVAL

Symmetrical Uncert Attribute Set Eval es un algoritmo empleado para el problema de selección de subconjuntos de variables. Para ello evalúa el valor, medido en incerteza simétrica con respecto a una variable clase, que presentan los distintos subconjuntos de variables. La medida de incerteza simétrica ha sido tratada anteriormente, y puede consultarse en el apartado destinado al algoritmo Symmetrical Uncert Attribute Eval. Además, el algoritmo SUASE tiene en cuenta la correlación entre las variables en los propios subconjuntos, para no considerar aquellas variables no relevantes por ser redundantes.

Para la aplicación de este algoritmo se ha seleccionado como método de búsqueda el método FCBF Search.

En el programa Weka el algoritmo presenta tan sólo una opción, Missing Merge, que permite decidir cual va a ser el tratamiento de los valores perdidos.

Por ello, para realizar los ajustes necesarios en la evaluación de las variables, se van a considerar las opciones que presenta el método de búsqueda. El método elegido, FCBF, es un método de selección de variables basado en la medida de correlación analizando la relevancia y la redundancia. Se utiliza conjuntamente con un evaluador de conjuntos de variables, en concreto, la medida de incerteza simétrica. Podemos configurar varios parámetros antes de aplicar el algoritmo con las siguientes opciones de las que dispone este método de búsqueda.

generateDataOuput.- Permite generar un nuevo conjunto de datos según las variables que hayan sido seleccionadas.

generateRanking.- Está opción, por defecto “true”, hace factible generar rankings de variables.

numToSelect.- Puede indicarse aquí el número de variables que queremos retener en nuestra selección. Por defecto, todas las variables son seleccionadas.

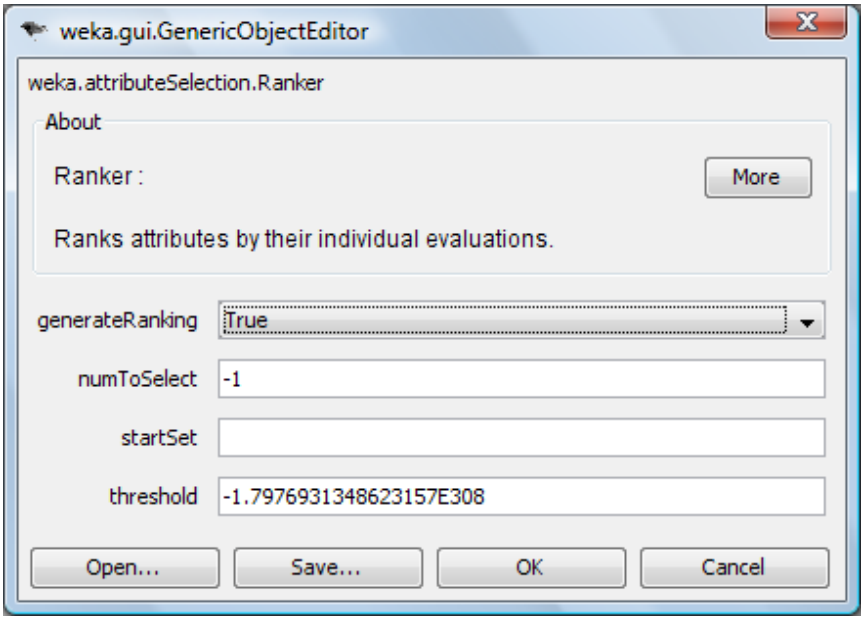
startSet.- Podemos definir aquí si queremos ignorar algún conjunto de variables.

Threshold.- En esta opción podemos determinar el valor de un umbral para la selección de variables. Aquellas variables que alcancen o superen este umbral serán incluidas en la selección. (Considerando también la configuración de la opción numToSelect). El valor que presenta la opción threshold por defecto incluye a todas las variables.

3. ALGORITMOS DE PONDERACIÓN CONTÍNUA

3.1. MÉTODO DE BÚSQUEDA: RANKER

Es una función de Weka que determina el rango de los atributos(variables) por sus evaluaciones individiduales. Se usa en conjunción con los evaluadores que ordenan los atributos por relevancia. (ReliefF, GainRatio, Entropy etc). Es decir, se usa junto con los algoritmos de ponderación contínua.



- **Determina el rango de los atributos(variables) por sus evaluaciones individiduales.** Se usa en conjunción con los evaluadores que ordenan los atributos por relevancia. (ReliefF, GainRatio, Entropy etc).

  OPCIONES

  - **generateRanking** – Es una constante opcional. Ranker es solo capaz de generar atributos ranking
  - **numToSelect** – Especifica el número de atributos a retener. Por defecto viene el valor -1 que indica que todos los atributos son retenidos. Usa otra opción o un umbral para reducir la colección de atributos.
  - **startSet** – Especifica una colección de atributos a ignorar. Cuando generamos el ranking, Ranker no evalúa los atributos en esta lista . Esto se especifica con una lista de atributos separada por comas, empezando en 1. Se puede incluir intervalos. Ejemplo. 1,2,5-9,17.
  - **Threshold –** Se fija el umbral por el cual se pueden descartar atributos. El valor que viene dado por defecto no descarta ningún atributo. Se usa esta opción o numToSelect para reducir la colección de atributos

3.2. CHI SQUARE ATTRIBUTE EVAL

Evalúa el valor de un atributo mediante el cálculo del valor del estadístico Chi-cuadrado con respecto a la clase.

OPCIONES:

- **Binarize Numeric Atributes:**

  Sólo binariza los atributos numéricos en lugar de discretizarlos.

- **Missingmerge**:

  Los valores perdidos son distribuidos a través de otros valores en función de su frecuencia. Por el contrario serán tratados como valores independientes.

3.3. FILTERED ATTRIBUTE EVAL

Clase que se utiliza para controlar a un evaluador de atributo arbitrario sobre los datos que han pasado un filtro arbitrario (nota: no se cambian ni el orden ni el número de atributos). La estructura del filtro está basada exclusivamente en la muestra de entrenamiento.

OPCIONES:

- **attributeEvaluator** – Se utiliza para seleccionar el evaluador de atributos que se va usar.
- **filter**-- Determina el filtro utilizado en el análisis.

3.4. GAIN RATIO ATTRIBUTE EVAL

Evalúa el valor de cada uno de los atributo en función del beneficio o proporción en que éste se incrementa con respecto a la variable clase.

OPCIONES:

- **missingMerge–** Distribuye los valores perdidos/valores faltantes. Este recuento se distribuye a otros valores en proporción a sus frecuencias. También se pueden tratar estos valores perdidos/faltantes como un valor por separado.

3.5. INFO GAIN ATTRIBUTE EVAL

Es una función que sirve para conocer el valor de una variable (atributo) midiendo la información que se gana respecto a la clase. Este algoritmo discretiza los atributos numéricos.

OPCIONES:

- **binarizeNumericAttributes** – Procede a binarizar los atributos numéricos
- **missingMerge** -- Distribuye los valores perdidos/valores faltantes. Este recuento se distribuye a otros valores en proporción a sus frecuencias. También se pueden tratar estos valores perdidos/faltantes como un valor por separado.

3.6. ONER ATTTRIBUTE EVAL

Es un algoritmo implementado en Weka que evalúa el valor de las variables usando un clasificador OneR.

OneR, es uno de los clasificadores más sencillos y rápidos, aunque en ocasiones, sus resultados son sorprendentemente buenos en comparación con algoritmos mucho más complejos. Este clasificador, simplemente selecciona el atributo que mejor "explica" la clase de salida. Si hay atributos numéricos, busca los umbrales para hacer reglas con mejor tasa de aciertos.

OPCIONES:

- **evalUsingTrainingData** -- Utiliza los datos de entrenamiento para evaluar atributos mejores que validación cruzada.
- **folds** -- Número de pliegues para validación cruzada.
- **minimumBucketSize** -- El número mínimo de objetos en una clase (pasado a OneR).
- **seed** --Semilla que usamos en validación cruzada.

3.7. RELIEVEFF ATTRIBUTE EVAL

Evalúa el valor de un atributo por muestreo repetidamente y considerando el valor de la atributo de la muestra de entrenamiento más cercana de la misma y clase diferente. Puede funcionar en ambas clases de datos discreta y continua. Utiliza el algoritmo RELIEF el cual es un algoritmo estadístico de selección de variables que usa muestras de entrenamiento para asignar peso relevante a cada característica.

Relief es un algoritmo de selección de variables predictivas inspirado en el conocimiento. Dado un conjunto de datos de entrenamiento S, muestra de tamaño m, y un umbral de relevancia ζ, Relief detecta esas variables predictivas que son estadísticamente relevantes.

Sea

- S denota una colección de datos de entrenamiento de tamaño n.
- F es una colección de variables predictivas dada $\{f_1, f_2, \ldots, f_p\}$.
- X es denotado por un vector p-dimensional $(x_1, x_2, \ldots, x_p)$

Donde x_j denota el valor de la variable predictiva f_j de X.

ζ codifica un umbral de relevancia ($0 \leq t \leq 1$). Se asume que la escala de las variables predictivas es nominal o numérica (entera o real). Los diferentes valores de las variables predictivas entre dos instantes X e Y son definidos por la siguiente función diff.

Cuando x_k e y_k son nominales:

$$\text{diff}(x_k, y_k) = \begin{cases} 0 & \text{si } x_k \text{ e } y_k \text{ son la misma} \\ 1 & \text{si } x_k \text{ e } y_k \text{ son diferentes} \end{cases}$$

Cuando x_k e y_k son numéricas:

$\text{diff}(x_k, y_k) = (x_k - y_k)/nu_k$ dónde nu_k es una normalización unidad para normalizar los valores de diff en el intervalo [0,1].

Relief escoge una muestra compuesta por m tripletes de un dato X, sus datos Near-hit y Near-miss dato.

- Near-hit: Si pertenece a los vecinos cercanos de X y tiene la misma categoría que X.
- Near-miss: Si pertenece a los vecinos cercanos de X pero no tiene la misma categoría que X.

Relief usa la distancia Euclidea p-dimensional para seleccionar Near-hit y Near-miss. Relief llama a una rutina para actualizar el peso del vector de las variables predictivas, W, para todos los tripletes y determinar el promedio del peso de la relevancia del vector de variables predictivas.

Relief selecciona las variables en las que el peso medio de relevancia ('nivel de relevancia') está por encima del umbral ζ .

Relief es válido solo cuando:

- El nivel de relevancia es alto para las variables relevantes y bajo para las variables irrelevantes.
- ζ puede ser escogido para retener las variables relevantes y descartar las irrelevantes.

El análisis teórico muestra que:

- La relevancia es positiva cuando la variable es relevante y próxima a cero o negativa cuando es irrelevante.
- Un método estadístico de intervalos estimados, puede ser usado para determinar el valor de ζ

La complejidad de Relief es $\theta(pmn)$ porque calcula la distancia entre X y cada uno de los n datos, tomando $\theta(p)$ veces, para determinar su Near-miss and Near-hit dentro de un bucle iterativo m veces. m es una constante que afecta a la exactitud de los niveles de relevancia. Luego, m es escogido independientemente de p y n, la complejidad está en $\theta(pn)$. De este modo el algoritmo puede seleccionar estadísticamente las variables relevantes en tiempo lineal en términos del número de variables y el número de datos de entrenamiento.

El pseudocódigo del algoritmo es el siguiente

Relief(S, m,ζ)

Separamos S en dos S^+={datos positivos} y S^-={datos negativos}

W=(0,0,…..,0)

Desde i =1 hasta m

Se escoge aleatoriamente XϵS

Se escoge aleatoriamente un dato positivo próximo a X, $Z^+\epsilon S^+$

Se escoge aleatoriamente un dato negativo próximo a X, $Z^-\epsilon S^-$

Si (X es positivo)

luego Near-hit= Z^+; Near-miss= Z^-

sino Near-hit= Z^-; Near-miss= Z^+

update-weight(W, X, Near-hit, Near-miss)

Relevancia=(1/m)W

Desde i=1 hasta p

si(relevancia$_i \geq \zeta$)

luego f_i es una variable relevante

sino f_i no es una variable relevante

update-weigth(W, X, Near-hit, Near-miss)

desde i=1 hasta p

$W_i=W_i$ – diff(x_i, near-hit$_i$)^2+ diff(x_i,near-miss$_i$)^2

¿Qué hace Weka?

- Evalúa el valor de un atributo por muestreo repetidamente y considerando el valor de la atributo de la muestra de entrenamiento más cercana de la misma y clase diferente. Puede funcionar en ambas clases de datos discreta y continua.
- RELIEF: describe un algoritmo estadístico de selección de características que usa muestras de entrenamiento para asignar peso relevante a cada característica.

OPCIONES:

- **numNeighbours** – Número de vecinos más cercanos para los atributos estimados.
- **sampleSize** – Número de casos de muestra. Por defecto(-1) indica que todos los casos serán utilizados para la estimación de los atributos.
- **seed –**Semillas aleatorias para el muestreo de casos.
- **sigma** –Conjunto de influencia de vecinos más cercanos. Utiliza una función exponencial para controlar la rapidez de disminución del peso de los casos más distantes. Uso junto con weightByDistance. Valores aconsejados= 1/5 a 1/10 el númeso de vecinos más cercanos
- **weightByDistance** –proporciona el peso para los vecinos más cercanos

3.8. SYMMETRICAL UNCERT ATTRIBUTE EVAL

Symmetrical Uncert Attribute Eval es un algoritmo que evalúa variables considerando, para ello, la medida de incerteza simétrica con respecto a la clase que presenta cada variable. Este algoritmo, de tipo filter, resulta útil para problemas de selección de variables de alta dimensionalidad, por la rapidez en su ejecución.

El algoritmo emplea como criterio para otorgar el peso a cada variable la Symmetrical Uncert (SU). La SU es una medida que relaciona la incerteza simétrica que presenta cada variable con respecto a la clase compensada por la medida InfoGain. En concreto.

$$SU(X,Y) = 2\,[\,IG(X|Y) / H(X) + H(Y)]$$

La incerteza simétrica parte de una medida de correlación basada en el concepto de información teórica de entropía, como la medida de la incerteza de una variable aleatoria. La entropía de una variable se define como,

$$H(X) = -\sum_{i} P(x_i)\log_2(P(x_i))$$

Además, una vez que se han observado los valores de de otra variable (Y) la entropía de X se define,

$$H(X \mid Y) = -\sum_{j} P(y_i) \sum_{i} (x_i \mid y_i) \log_2(P(x_i \mid y_j))$$

Donde P(x_i) es la probabilidad a priori para valores de X, y P(x_i|y_i) son las probabilidades a posteriori de X dados los valores de la variable Y. Cantidades para las cuales la entropía de Y decrece reflejan información adicional sobre X proporcionada por Y y se conoce como information gain:

$$IG(X \mid Y) = H(X) - H(X \mid Y)$$

De acuerdo a esta medida, una variable Y se considera más correlacionada con la variable X que la variable Z sí IG(X|Y)>IG(Z|Y). Sobre la medida Information Gain se presenta el siguiente teorema. Information Gain es simétrico para dos variables X e Y.

La simetría es una propiedad deseada para medir la correlación entre variables. Sin embargo, la Information Gain es segada a favor de variables con muchos valores. Además, los valores han de ser normalizados para asegurar que sean comparables y tengan la misma influencia. De ahí la definición de la incerteza simétrica.

Es compensado el sesgo por information gain hacia variables con más valores y normaliza sus valores en el rango [0,1] con el valor 1 indicando que el conocimiento del valor de cada una predice completamente al valor de la otra y el valor 0 indica que X e Y son independientes. Además, aún trata un par de variables simétricamente. La medida basada en entropía requiere variables nominales, pero puede ser aplicado para medir la correlación entre variables continuas, sí sus valores son discretizados adecuadamente con anterioridad.

El algoritmo Symetrical Uncert Attribute Eval requiere el uso de un método de búsqueda de tipo Ranker.

El algoritmo el cual está implementado en Weka, contiene una opción missingMerge para gestionar el tratamiento de los valores perdidos. Además, se pueden configurar algunos parámetros en el menú de opciones del método de búsqueda Ranker.

La opción generateRanking que aparece seleccionada como “true” por defecto permite generar una clasificación de las variables. Además, puede descartarse un conjunto de variables para que no sean evaluadas con la opción startSet.

Por último, resulta útil reducir el número de variables que van a ser seleccionadas. Para ello pueden ajustarse los parámetros de las opciones numToSelect y threshold. Con numToSelect es posible decidir de antemano el número de variables que van a ser seleccionadas. Mientras que la opción threshold permite determinar un umbral que las variables han de superar ara ser seleccionadas. Ambas opciones seleccionan todas las variables por defecto.

La salida del algoritmo en Weka puede proporcionar un ranking de las variables en función del valor de incerteza simétrica obtenido por cada una. Y, además, presentar una lista con las variables que han sido seleccionadas. Todo ello en función de los parámetros que hayan sido seleccionados previamente en el menú opciones del método de búsqueda Ranker. La salida con el ranking y la selección de variables depende también del modo de selección de variables que se ha utilizado. En concreto para toda la muestra de entrenamiento aparece un ranking con los valores de SU, mientras que, si se usa validación cruzada como modo de selección de variables se presentan el average merit y el average rank para cada una de las variables. El average merit representa una media de los valores de SU obtenidos en las repeticiones de la validación cruzada. Y, el average rank es un promedio de las posiciones que ha obtenido cada variable en cada una de las repeticiones.

En conclusión, el symmetrical uncert attribute eval, es un método rápido de selección de variables, pues se trata de un método de tipo filter permitiendo su aplicación a problemas de alta dimensionalidad. Necesita de un método de búsqueda Ranker, por lo que proporciona un evaluación de las variables basada en la medida de SU que presentan con respecto a la clase. Permitiendo realizar una selección de las variables más relevantes para conocer la variable clase, de utilidad en problemas de selección de variables.

4. LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis (LSA) es una técnica natural del procesamiento del lenguaje, en particular en vectores semánticos, que analiza las relaciones entre una colección de documentos y las palabras que ellos contienen por producir una colección de conceptos relacionados a los documentos y palabras.

LSA puede usar una matriz palabra-documento la cual describe cuantas veces aparecen las palabras en los documentos; es una matriz dónde las filas corresponden a las palabras y las columnas a los documentos.

Esta matriz también es común en los modelos semánticos estándar, sin embargo no es necesario expresarla explícitamente como una matriz, dado que las propiedades matemáticas de las matrices no son usadas.

LSA transforma la matriz de sucesos en una relación entre palabras y algunos conceptos, y una relación entre esos conceptos y los documentos. Así de esta manera, las palabras y los documentos están directamente relacionados a través de los conceptos.

Este nuevo espacio de conceptos puede ser usado para:

- Comparar los documentos en el espacio conceptual
- Encontrar similares documentos a través del lenguaje, después de analizar un conjunto base de documentos traducidos
- Encontrar relaciones entre palabras (sinonimia y polisemia)
- Proporcionar una búsqueda de los términos, traducirlos en el espacio conceptual, y encuentrar documentos parecidos.

Después de la construcción de la matriz de sucesos, LSA encuentra un menor rango aproximado a la matriz palabra-documento. Puede haber varias razones para esta aproximación:

- La original matriz palabra-documento es presuntamente grande para el cálculo; en este caso, la aproximación de la matriz con menos rango es interpretado como una aproximación.
- La original matriz palabra-documento tiene demasiado ruido(anécdotas, ejemplos…). En este caso, la aproximación es interpretada como una matriz “poco ruidosa”(mejor que la original).
- La matriz palabra-documento original es supuesta demasiado escasa en relación con la matriz de documento término "verdadera". La matriz original pone en una lista sólo las palabras en cada documento, mientras que nosotros podríamos estar interesados en todas las palabras relacionadas con cada documento - generalmente una colección mucho más de grande debido a la sinonimia.

4.1. EN WEKA

- Se realiza el “latent semantic analysis” y la transformación de los datos.
- Se usa junto una búsqueda Ranker. Un bajo rango aproximado de todo el conjunto de datos es encontrado por la dspecificación de valores singulares
- OPCIONES:
    - **maximumAttributeNames** –El máximo número de atributos a incluir en la transformación de los nombres de atributos .
    - **normalize** –Normaliza los datos.
    - **rank** – Rango de la matriz que se usa para la reducción de los datos. Puede ser una proporción indicada para la cobertura deseada

5. COMPONENTES PRINCIPALES

El Análisis de Componentes Principales (ACP) es una técnica estadística de síntesis de la información, o reducción de la dimensión (número de variables). Las nuevas componentes principales o factores serán una combinación lineal de las variables originales, y además serán independientes entre sí.

Un aspecto clave en ACP es la interpretación de los factores, ya que ésta no viene dada a priori, sino que será deducida tras observar la relación de los factores con las variables iniciales (habrá, pues, que estudiar tanto el signo como la magnitud de las correlaciones).

El análisis de componentes principales consta de las siguientes fases:

- **Análisis de la matriz de correlaciones**

Un análisis de componentes principales tiene sentido si existen altas correlaciones entre las variables, ya que esto es indicativo de que existe información redundante y, por tanto, pocos factores explicarán gran parte de la variabilidad total.

- **Selección de los factores**

La elección de los factores se realiza de tal forma que el primero recoja la mayor proporción posible de la variabilidad original; el segundo factor debe recoger la máxima variabilidad posible no recogida por el primero, y así sucesivamente. Del total de factores se elegirán aquéllos que recojan el porcentaje de variabilidad que se considere suficiente.

- **Análisis de la matriz factorial**

Una vez seleccionados los componentes principales, se representan en forma de matriz. Cada elemento de ésta representa los coeficientes factoriales de las variables (las correlaciones entre las variables y los componentes principales). La matriz tendrá tantas columnas como componentes principales y tantas filas como variables.

- **Interpretación de los factores**

Para que un factor sea fácilmente interpretable debe tener las siguientes características, que son difíciles de conseguir:

  - Los coeficientes factoriales deben ser próximos a 1.
  - Una variable debe tener coeficientes elevados sólo con un factor.
  - No deben existir factores con coeficientes similares.

Cálculo de las puntuaciones factoriales

Son las puntuaciones que tienen los componentes principales para cada caso, que nos permitirán su representación gráfica

$X_{ij} = \Sigma a_{is}\, Z_{sk\,;\,S=}1,\ldots\ldots,k$

Dónde a son los coeficientes y Z son los valores estandarizados que tienen las variables en cada uno de los sujetos de la muestra.

El objetivo de PCA es encontrar una nueva colección de atributos(esta nueva colección de atributos se denomina por componentes principales, PCs) que verifican las siguientes propiedades: Las PCs son

- Combinaciones lineales de los atributos originales
- Ortogonales entre si
- Capturan la máxima cantidad de variabilidad de los datos.

A menudo, la variabilidad de los datos puede ser capturada por un número relativamente pequeño de PCs, por consiguiente, PCA puede dar como resultado datos con poca dimensión con menos ruido que el modelo original.

PCA depende de la escala de los datos, y por lo tanto los resultados a veces no son concluyentes. Además , las componentes principales no son siempre fáciles para hacer de interpretar.

Además de obtener una nueva colección de variables, PCA también es útil en un problema para obtener mejoras en la clasificación.

Veamos las diferencias de tres variantes de la computación de PCA con el objetivo de obtener mejoras en la clasificación:

Para todos los subconjuntos basados en PCA primero realizamos un cambio en la media de todos los rasgos tal que la media se hace 0. Denotamos la matriz resultante como M.

- PCA1: Los autovalores y los vectores propios son calculados usando la covarianza de la matriz M. Los nuevos valores del atributo son luego calculados al multiplicar M con los vectores propios de Cov(M).
- PCA2: Los autovalores y los vectores propios son calculados usando la correlación de la matriz M. Los nuevos valores del atributo son luego calculados al multiplicar M con los vectores propios de Corr(M).
- PCA3: Cada variable de M es normalizada por la estandarización de su desviación. Estos valores normalizados son usados para calcular los autovalores y los vectores propios(no hay diferencia entre los coeficientes de covarianza y correlación) y también para el cálculo de los nuevos atributos.

5.1. EN WEKA

- Realiza un análisis de componentes principal y la transformación de los datos. Se emplea en conjunción con una búsqueda de Ranker. La reducción de dimensión se logra escogiendo bastantes vectores propios para considerar para algún porcentaje de la discrepancia en los datos originales---la falta 0.95 (el 95 %). El ruido de atributo puede ser filtrado por la transformación el espacio de la componente principal, eliminando algunos de los peores vectores propios, y luego devolviéndolos al ámbito original.

- Considera cada nivel de la variable como una variable, es decir, si tenemos una variable sexo con dos niveles: hombre y mujer. Pues al aplicar componentes principales consideramos que sexo_hombre es una variable y sexo_mujer sería otra variable.

OPCIONES

- **maximumAttributeNames** –El máximo número de atributos a incluir en la transformación de los nombres de atributos .
- **normalize** – Normalizar los datos
- **transformBackToOriginal** – Transformación del espacio de datos y devolviendolo al ámbito original. Si sólo son retenidas las n mejores componentes principales (poniendo varianceCovered < 1) entonces esta opción dará una colección en el espacio original, pero con menos ruido de atributo.
- **varianceCovered** --Conserve bastantes componentes principales de los atribuutos para considerar para esta proporción de discrepancia.

6. PROBLEMA 1: ENCUESTA SOBRE ACCIDENTES EN EMPRESAS MINERAS

Tenemos una tabla que recoge los resultados de una encuesta en varias empresas mineras, sobre las circunstancias que rodearon la ocurrencia de un suceso (variable S) que puede ser Accidente o Incidente en función de su gravedad.

OBJETIVO del análisis: Determinar las condiciones asociadas a uno u otro tipo de suceso con objeto de conocer su casuística y adoptar medidas preventivas en su caso. Para realizar el análisis tenemos que poner como variable respuesta la variable suceso.

En este problema tenemos las siguientes variables con sus correspondientes etiquetas:

- Variable respuesta: SUCESO (S)
- Variables explicativas: HORA (H), DÍA (D), MES (M), NACIONALIDAD (Na),HORAS DE OPERACIÓN (HO), TIPO DE CONTRATO (TC), TIEMPO EN OBRA (TO), PUESTO DE TRABAJO (PT) FORMACIÓN (F), COMUNIDAD AUTÓNOMA (CA) RÉGIMEN (R), PLAZO DE EJECUCIÓN (PE), DIRECCIÓN Y SUPERVISIÓN (DS)

Hay que tener especial cuidado con las variables COMUNIDAD AUTÓNOMA, RÉGIMEN y DIRECCIÓN Y SUPERVISIÓN pues ,en principio, se duda de que exista suficiente representación para cada tipo de suceso.

6.1. APLICACIÓN: ALGORITMOS DE PONDERACIÓN BINARIA

En este apartado comparamos los resultados arrojados en función de los evaluadores de conjuntos de atributos utilizados, empleando para ello toda la muestra de entrenamiento. Hay que señalar que para el caso del problema de Accidentes, los análisis se han realizado en un primer momento incluyendo a la variable encuesta (E) y posteriormente eliminándola de la muestra de entrenamiento, puesto que se ha observado que produce resultados no relevantes que pueden llevar a confusión.

Método	Solución
Cfs.Subset.Eval	14,16
Classifier.Subset.Eval	2,3,4,6,11,12,13,15,18
Consistency.Subset.Eval	1,2,3,4,6,9,12,13,14,15,16
Filtered Subset Eval	4,14

6.1.1. Wrapper Subset Eval

Aplicación del algoritmo Wrapper Subset Eval al problema de la encuesta sobre accidentes en empresas mineras, con todo el conjunto de datos.

En esta aplicación se ha elegido Naive Bayes como clasificador, con 5 grupos en la validación cruzada interna, y se ha reducido el valor de threshold. El método de búsqueda empleado es Best First. Como modo de selección de subconjuntos de variables se ha elegido sobre todo el conjunto de entrenamiento, ya que el método Wrapper tiene ya implementada una validación cruzada interna.

El número total de subconjuntos evaluados es de 104, y el mérito del mejor subconjunto encontrado es de 0.129. Han sido seleccionada una variable, CA (Comunidad Autónoma).

Resultados de la aplicación del algoritmo con validación cruzada de diez grupos en fase de test

Número de grupos	(%)	attribute
3	(30 %)	1 H
5	(50 %)	2 D
3	(30 %)	3 M
7	(70 %)	4 HO
3	(30 %)	5 Ed
4	(40 %)	6 Na
2	(20 %)	7 An
0	(0 %)	8 TC
3	(30 %)	9 TO
4	(40 %)	10 PT
5	(50 %)	11 F
1	(10 %)	12 RP
2	(20 %)	13 FP
2	(20 %)	14 ER
3	(30 %)	15 CT

Las variables que aparecen en un mayor número de grupos son: HO (70%), y, D y F en la mitad de los grupos.

6.1.2. Comparación de resultados eliminando y no eliminando las variables CA,R,DS:

6.1.2.1. Cfs.Subset.Eval:

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):

  14,16 : 2 (CA,ER)

- Selección de atributos eliminando las variables CA, R y DS:

  4,9,12 : 3 (HO,TO,RP)

6.1.2.2. Classificier.Subset.Eval

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):

  2,3,4,6,11,12,13,15,18 : 9

- Selección de atributos eliminando las variables CA, R y DS:

  2,3,4,6,11,12,13,15 : 8

6.1.2.3. Consistency.Subset.Eval

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):

  1,2,3,4,6,9,12,13,14,15,16 : 11

- Selección de atributos eliminando las variables CA, R y DS:

  2,3,4,5,6,12,13,14 : 8

6.1.2.4. Wrapper Subset Eval

La aplicación respetando los parámetros empleados con todo el conjunto de variables, proporciona los siguientes resultados:

El número total de subconjuntos evaluados es de 125, y el mérito del mejor subconjunto encontrado es de 0.21. Se han seleccionado 6 variables: H, M, HO, PT, ER y CT. Esto es: Hora, Mes, Horas de Operación, Puesto de Trabajo, Evaluación del Riesgo, Condiciones Adecuadas.

6.1.2.5. Filtered Subset Eval:

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):

  4,14 :2 (HO,CA)

- Selección de atributos eliminando las variables CA, R y DS:

  4 : 1

6.1.2.6. Symmetrical Uncert Subset Eval

Ranking de variables

J	SU(j,Class)	I	SU(i,j).
14	0.3384374	*	
4	0.16181	14	0.22352788579516433
15	0.1508562	14	0.5125757431658408
9	0.1407063	14	0.3985734748017905
1	0.0752126	14	0.1875358177411624
10	0.0699728	14	0.1280362572373483
11	0.0653226	14	0.24485914587573843
13	0.0586993	14	0.10242070069828914
18	0.0442482	14	0.06615039142203037
2	0.0413987	14	0.11456496786218001
16	0.0371833	*	
3	0.0279837	14	0.12329914628833184
17	0.0247748	14	0.10242070069828914
6	0.0204587	14	0.10523002099003279
8	0.0171163	14	0.06306200469783273
12	0.0075219	14	0.049211874834100416
7	0	14	0.0
5	0	14	0.0

Ranking de variables:

0.3384 14 CA

0.0372 16 ER

Variables seleccionadas: 14,16 : 2

Las variables seleccionadas son Comunidad Autónoma y Evaluación del riesgo.

6.1.3. Comparación de los resultados con y sin validación cruzada:

6.1.3.1. Cfs.Subset.Eval

- Selección de atributos sin validación cruzada:

  14,16 : 2

- Selección de atributos con validación cruzada:

number of folds (%)	attribute
0(0 %)	1 H
0(0 %)	2 D
0(0 %)	3 M
1(10 %)	4 HO
1(10 %)	5 Ed
0(0 %)	6 Na
0(0 %)	7 An
0(0 %)	8 TC
0(0 %)	9 TO
0(0 %)	10 PT
0(0 %)	11 F
0(0 %)	12 RP
0(0 %)	13 FP
10(100 %)	14 CA
0(0 %)	15 R
9(90 %)	16 ER
0(0 %)	17 CT
0(0 %)	18 DS

Obtenemos que la única variable que está seleccionada por todos los grupos es CA, la variable ER es seleccionada en el 90% de los grupos y las variables HO y ED tan solo son seleccionadas en un 1% de los grupos.

6.1.3.2. Clasificier.Subset.Eval

- Selección de atributos sin validación cruzada:
  2,3,4,6,11,12,13,15,18 : 9
- Selección de atributos con validación cruzada:

number of folds (%)	attribute
0(0 %)	1 H
10(100 %)	2 D
10(100 %)	3 M
10(100 %)	4 HO
0(0 %)	5 Ed
10(100 %)	6 Na
0(0 %)	7 An
0(0 %)	8 TC
0(0 %)	9 TO
0(0 %)	10 PT
10(100 %)	11 F
10(100 %)	12 RP
10(100 %)	13 FP
0(0 %)	14 CA
10(100 %)	15 R
0(0 %)	16 ER
0(0 %)	17 CT
10(100 %)	18 DS

Las variables que aparecen en todos los grupos son 2,3,4,6,11,12,13,15,18.
Por tanto, seleccionamos las mismas variables con o sin validación cruzada.

6.1.3.3. Consistency.Subset.Eval

- Selección de atributos sin validación cruzada:

  1,2,3,4,6,9,12,13,14,15,16 : 11

- Selección de atributos con validación cruzada:

number of folds (%)	attribute
8(80 %)	1 H
10(100 %)	2 D
10(100 %)	3 M
10(100 %)	4 HO
0(0 %)	5 Ed
10(100 %)	6 Na
0(0 %)	7 An
0(0 %)	8 TC
8(80 %)	9 TO
0(0 %)	10 PT
2(20 %)	11 F
10(100 %)	12 RP
10(100 %)	13 FP
8(80 %)	14 CA
10(100 %)	15 R
8(80 %)	16 ER
0(0 %)	17 CT
2(20 %)	18 DS

Las variables D, M, HO, Na, RP, FP y R entran a formar parte de todos los grupos (pliegues), mientras que las variables H, TO, CA, ER forman parte de un 80% de los grupos, y las variables F y DS tan solo en el 20%.

6.1.3.4. Filtered Subset Eval:

- Selección de atributos sin validación cruzada:
  4,14 : 2
- Selección de atributos con validación cruzada:
  Number of folds (%) Attribute

0(0 %)	1 H
0(0 %)	2 D
0(0 %)	3 M
9(90 %)	4 HO
0(0 %)	5 Ed
0(0 %)	6 Na
0(0 %)	7 An
0(0 %)	8 TC
0(0 %)	9 TO
0(0 %)	10 PT
0(0 %)	11 F
0(0 %)	12 RP
0(0 %)	13 FP
10(100 %)	14 CA
1(10 %)	15 R
0(0 %)	16 ER
0(0 %)	17 CT
0(0 %)	18 DS

Obtenemos que la variable CA está presente en todos los pliegues, seguida de la variable HO que se encuentra en el 90% de los pliegues y la variable R tan solo se encuentra en el 1% de los pliegues.

6.1.3.1. Wrapper Subset Eval

Resultados de la aplicación del algoritmo con validación cruzada de diez grupos en fase de test

Número de grupos	(%)	attribute
3	(30 %)	1 H
5	(50 %)	2 D
3	(30 %)	3 M
7	(70 %)	4 HO
3	(30 %)	5 Ed
4	(40 %)	6 Na
2	(20 %)	7 An
0	(0 %)	8 TC
3	(30 %)	9 TO
4	(40 %)	10 PT
5	(50 %)	11 F
1	(10 %)	12 RP
2	(20 %)	13 FP
2	(20 %)	14 ER
3	(30 %)	15 CT

Las variables que aparecen en un mayor número de grupos son: HO (70%), y, D y F en la mitad de los grupos.

6.1.3.2. Symmetrical Uncert Attribute Set Eval

average merit	average rank	attribute
0.016 +- 0.048	0.2 +- 0.6	5 Ed
0.345 +- 0.039	1 +- 0	14 CA
0.038 +- 0.025	1.9 +- 0.7	16 ER

Se han seleccionado tres variables: edad, Comunidad Autónoma y Evaluación del Riesgo.

Aplicación del algoritmo excluyendo las variables CA, R y DS con todo el conjunto da entrenamiento.

Attribute ranking.

J	SU(j,Class)	I	SU(i,j).
4	0.16181	*	
9	0.1407063	4	0.1953175436072784
1	0.0752126	*	
10	0.0699728	*	
11	0.0653226	4	0.0742096491513346
13	0.0586993	*	
2	0.0413987	4	0.08007700467068528
14	0.0371833	*	
3	0.0279837	4	0.06365284650564708
15	0.0247748	1	0.0313015150686073
6	0.0204587	1	0.1079142748320399
8	0.0171163	4	0.06515161963852294
12	0.0075219	4	0.0075219442823411635
7	0	4	0.0
5	0	4	0.0

Ranking de variables:

0.1618 4 HO

0.0752 1 H

0.07 10 PT

0.0587 13 FP

0.0372 14 ER

Attribute ranking.

J	SU(j,Class)	I	SU(i,j).
4 ;	0.16181 ;	*	
9 ;	0.1407063 ;	4 ;	0.1953175436072784
1 ;	0.0752126 ;	*	
10 ;	0.0699728 ;	*	
11 ;	0.0653226 ;	4 ;	0.0742096491513346
13 ;	0.0586993 ;	*	
2 ;	0.0413987 ;	4 ;	0.08007700467068528
14 ;	0.0371833 ;	*	
3 ;	0.0279837 ;	4 ;	0.06365284650564708
15 ;	0.0247748 ;	1 ;	0.0313015150686073
6 ;	0.0204587 ;	1 ;	0.1079142748320399
8 ;	0.0171163 ;	4 ;	0.06515161963852294
12 ;	0.0075219 ;	4 ;	0.0075219442823411635
7 ;	0 ;	4 ;	0.0
5 ;	0 ;	4 ;	0.0

Ranked attributes:

0.1618 4 HO

0.0752 1 H

0.07 10 PT

0.0587 13 FP

0.0372 14 ER

Variables seleccionadas: 4,1,10,13,14 : 5

Se han seleccionado las variables: Horas de Operación, Hora, Puesto de trabajo, Factores Personales y Evaluación del Riesgo.

Aplicación del algoritmo con validación cruzada de 10 grupos:

average merit	average rank	attribute
0.016 +- 0.048	0.2 +- 0.6	5 Ed
0.028 +- 0.058	0.2 +- 0.4	9 TO
0.006 +- 0.018	0.2 +- 0.6	12 RP
0.005 +- 0.014	0.3 +- 0.9	2 D
0.006 +- 0.018	0.4 +- 1.2	15 CT
0.16 +- 0.085	0.8 +- 0.4	4 HO
0.034 +- 0.042	0.9 +- 1.14	1 H
0.03 +- 0.037	1.1 +- 1.37	10 PT
0.031 +- 0.033	1.6 +- 1.69	13 FP
0.03 +- 0.031	2.1 +- 1.92	14 ER

Las variables han cambiado con respecto a la evaluación sobre toda la muestra de entrenamiento. Destacan: la Edad, el Tiempo en Obra y el Reconocimiento Previo del Peligro.

6.2. APLICACIÓN: ALGORITMOS DE PONDERACIÓN CONTÍNUA

En este apartado comparamos la solución dada por diferentes métodos con todas las variables y usando toda la muestra de entrenamiento. Además de esto también se muestran las diferencias que hay entre las soluciones:

Usando todas las variables y sin las variables CA,R y DS

Usando validación cruzada y sin usar validación cruzada.

Método	Solución
ChiSquared Attribute Eval	CA,HO,R,TO,H,PT,F,FP,D,M,ER,Na,DS,CT,TC,RP,Ed,An
One Attribute Eval	CA,R,TO,F,PT,H,TC,RP,HO,D,FP,M,DS,Ed,CT,An,ER,Na,E
RelievefF Attribute Eval	CA,H,ER,R,HO,TC,PT,Ed,TO,RP,D,FP,M,An,F,DS,E,Na,CT
Filtered Attribute Eval	An,Ho,Ed,CA,H,TO,R,Pt,D,F,M,FP,ER,DS,Na,CT,TC,RP
Gain Ratio Attribute Eval	CA,Ho,R,TO,An,Ed,FP,F,H,DS,PT,ER,D,CT,Na,M,TC,RP
Info Gain Ratio Attribute Eval	An,Ho,Ed,CA,H,TO,R,PT,D,F,M,FP,ER,DS,Na,CT,TC,RP
Symmetrical Uncert Attribute Eval	14,4,15,9,1,10,11,13,18,2,16,3,17,6,8,12,7,5 : 18

6.2.1. Comparación de resultados eliminando y no eliminando las variables CA, R y DS

6.2.1.1. Chi Square Attribute Eval

- Selección de atributos no eliminando las variables CA,R y DS:

  14,4,15,9,1,10,11,13,2,3,16,6,18,17,8,12,7,5 : 18

- Selección de atributos eliminando las variables Ca,R y DS:

  4,9,1,10,11,13,2,3,14,6,15,8,12,7,5 : 15

Podemos observar que el orden de las variables no cambia una vez se eliminan las variables CA,R,DS.

6.2.1.2. Filtered Attribute Eval

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):
  7,4,5,14,1,9,15,10,2,11,3,13,16,18,6,17,8,12 : 18
- Selección de atributos eliminando las variables CA, R y DS:
  7,4,5,1,9,10,2,11,3,13,14,6,15,8,12 : 15

6.2.1.3. Gain Ratio Attribute Eval

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):
  14,4,15,9,7,5,13,11,1,18,10,16,17,2,3,6,8,12 : 18
- Selección de atributos eliminando las variables CA, R y DS:
  4,9,7,5,13,11,1,10,14,15,2,3,6,8,12 : 15

6.2.1.4. Info Gain Attribute Eval

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):
  7,4,5,14,1,9,15,10,2,11,3,13,16,18,6,17,8,12 : 18
- Selección de atributos eliminando las variables CA, R y DS:
  7,4,5,1,9,10,2,11,3,13,14,6,18,8,12 : 15

Observamos que las variables siguen en el mismo orden excepto las variables señaladas en azul.

6.2.1.5. OneR Atttribute Eval

- Selección de atributos no eliminando las variables CA,R y DS(15,16,19):
  15,16,10,12,11,2,9,13,5,3,14,4,19,6,18,8,17,7,1 : 19
- Selección de atributos eliminando las variables CA, R y DS:
  12,10,11,2,9,13,5,14,3,4,6,8,18,17,7,1 : 16.

Observamos que las variables siguen en el mismo eliminando estas variables orden excepto las variables señaladas en azul.

6.2.1.6. RelievefF Attribute Eval

- Selección de variables

  15,2,17,16,5,9,11,6,10,13,3,14,4,8,12,19,1,7,18 : 19

- Selección de variables:

  2,17,5,9,11,6,10,13,3,14,4,8,12,1,7,18 : 16

  Las variables mantienen el orden de relevancia si se eliminan las variables Ca,R y DS

6.2.1.7. Symmetrical Uncert Attribute Eval

- Selección de variables

  14,4,15,9,1,10,11,13,18,2,16,3,17,6,8,12,7,5 : 18

- Selección de variables:

  4,9,1,10,11,13,2,14,3,15,6,8,12,7,5 : 15

Destacan principalmente las variables horas de operación y tiempo en obra. Un resultado mucho más lógico que el obtenido al incluir todas las variables.

6.2.2. Comparación de resultados con y sin validación cruzada

6.2.2.1. Chi Square Attribute Eval

- Selección de atributos sin validación cruzada:

  14,4,15,9,1,10,11,13,2,3,16,6,18,17,8,12,7,5 : 18

  Selección de atributos:

average merit	average rank	attribute
26.333 +- 2.616	1 +- 0	14 CA
11.201 +- 3.825	3.6 +- 4.48	4 HO
8.304 +- 1.337	3.8 +- 0.87	1 H
8.672 +- 2.148	4 +- 1.41	15 R
8.145 +- 1.261	4.3 +- 1	9 TO
6.542 +- 0.883	5.7 +- 1	10 PT
4.67 +- 1.025	8 +- 1.9	11 F
4.141 +- 0.531	8.9 +- 1.3	2 D
3.936 +- 1.391	9.1 +- 1.92	13 FP
3.474 +- 0.751	9.9 +- 0.83	3 M
2.858 +- 1.584	11.5 +- 2.58	16 ER
2.382 +- 1.016	12 +- 1.79	6 Na
2.041 +- 0.721	12.5 +- 1.36	18 DS
1.793 +- 1.077	13.2 +- 2.32	17 CT
0.921 +- 0.239	14.6 +- 1.11	8 TC
0.619 +- 0.842	15.1 +- 2.12	12 RP
0.677 +- 2.032	16.7 +- 3.58	5 Ed
0 +- 0	17.1 +- 0.54	7 An

  Podemos ver que la relevancia de las variables tanto con validación cruzada como sin ella se mantiene prácticamente igual.

6.2.2.2. Filtered Attribute Eval

- Selección de atributos sin validación cruzada:

  7,4,5,14,1,9,15,10,2,11,3,13,16,18,6,17,8,12 : 18

- Selección de atributos con validación cruzada:

Average merit	Average Rank	Atributos
0.491 +- 0.029	1 +- 0	7 An
0.379 +- 0.036	2.7 +- 0.64	4 HO
0.387 +- 0.034	2.8 +- 0.87	5 Ed
0.343 +- 0.039	3.5 +- 0.67	14 CA
0.111 +- 0.019	5.9 +- 1.04	1 H
0.106 +- 0.017	6.5 +- 1.2	9 TO
0.103 +- 0.029	6.8 +- 1.4	15 R
0.101 +- 0.011	7 +- 0.77	10 PT
0.075 +- 0.008	9.5 +- 1.12	2 D
0.061 +- 0.013	10.6 +- 1.28	11 F
0.057 +- 0.011	11 +- 1	3 M
0.047 +- 0.016	12.5 +- 1.86	13 FP
0.036 +- 0.02	13.8 +- 2.56	16 ER
0.035 +- 0.009	14 +- 1.48	18 DS
0.031 +- 0.013	14.6 +- 1.43	6 Na
0.022 +- 0.013	15.4 +- 2.01	17 CT
0.018 +- 0.004	16.4 +- 0.8	8 TC
0.009 +- 0.013	17 +- 2.05	12 RP

Se mantiene el mismo orden de relevancia de las variables tanto con cómo sin validación cruzada. Las que eran muy relevantes sin validación cruzada con validación cruzada lo siguen siendo y las que no eran muy relevantes sin validación cruzada, ahora tampoco lo son.

6.2.2.3. Gain Ratio Attribute Eval

- Selección de atributos sin validación cruzada:

  14,4,15,9,7,5,13,11,1,18,10,16,17,2,3,6,8,12 : 18

- Selección de atributos con validación cruzada:

Aerage merit	Average Rank	Atributos
0.306 +- 0.035	1.1 +- 0.3	14 CA
0.236 +- 0.021	2.2 +- 0.4	4 HO
0.225 +- 0.066	2.7 +- 0.64	15 R
0.171 +- 0.026	4 +- 0	9 TO
0.109 +- 0.007	5 +- 0	7 An
0.089 +- 0.008	6.4 +- 0.66	5 Ed
0.069 +- 0.023	8.3 +- 2.49	13 FP
0.063 +- 0.011	8.8 +- 1.54	11 F
0.056 +- 0.009	9.5 +- 1.43	1 H
0.056 +- 0.015	10.1 +- 1.76	18 DS
0.052 +- 0.005	10.2 +- 0.6	10 PT
0.039 +- 0.022	12.3 +- 3.16	16 ER
0.03 +- 0.003	13.3 +- 0.9	2 D
0.032 +- 0.019	13.7 +- 3.35	17 CT
0.021 +- 0.009	14.9 +- 1.51	6 Na
0.021 +- 0.004	15.5 +- 0.5	3 M
0.017 +- 0.004	16.2 +- 1.25	8 TC
0.012 +- 0.017	16.8 +- 2.64	12 RP

Observamos que en este caso sí se aprecian cambios en el orden de las variables si comparamos los resultados arrojados al utilizar la muestra de entrenamiento con y sin validación cruzada. Las variables a las cuales afecta este “cambio” son: la 17 (que tiene más peso sin validación cruzada) y la variable 3 (que le ocurre lo mismo, es decir, es más irrelevante si utilizamos la técnica de la validación cruzada).

6.2.2.4. Info Gain Attribute Eval

- Selección de atributos sin validación cruzada:

  7,4,5,14,1,9,15,10,2,11,3,13,16,18,6,17,8,12 : 18

- Selección de atributos con validación cruzada:

Average merit	Average Rank	Atributos
0.491 +- 0.029	1 +- 0	7 An
0.379 +- 0.036	2.7 +- 0.64	4 HO
0.387 +- 0.034	2.8 +- 0.87	5 Ed
0.343 +- 0.039	3.5 +- 0.67	14 CA
0.111 +- 0.019	5.9 +- 1.04	1 H
0.106 +- 0.017	6.5 +- 1.2	9 TO
0.103 +- 0.029	6.8 +- 1.4	15 R
0.101 +- 0.011	7 +- 0.77	10 PT
0.075 +- 0.008	9.5 +- 1.12	2 D
0.061 +- 0.013	10.6 +- 1.28	11 F
0.057 +- 0.011	11 +- 1	3 M
0.047 +- 0.016	12.5 +- 1.86	13 FP
0.036 +- 0.02	13.8 +- 2.56	16 ER
0.035 +- 0.009	14 +- 1.48	18 DS
0.031 +- 0.013	14.6 +- 1.43	6 Na
0.022 +- 0.013	15.4 +- 2.01	17 CT
0.018 +- 0.004	16.4 +- 0.8	8 TC
0.009 +- 0.013	17 +- 2.05	12 RP

6.2.2.5. OneR Atttribute Eval

- Selección de atributos sin validación cruzada:

  15,16,10,12,11,2,9,13,5,3,14,4,19,6,18,8,17,7,1

- Selección de atributos con validación cruzada:

average merit	average Rank	Atributos
87.097 +- 1.336	1 +- 0	15 CA
77.416 +- 1.48	2.1 +- 0.3	16 R
74.192 +- 0.917	4.2 +- 1.08	12 F
73.117 +- 1.609	5.3 +- 2.05	10 TO
72.737 +- 3.304	5.8 +- 3.52	11 PT
70.964 +- 4.154	8.4 +- 3.8	5 HO
70.968 +- 0.692	8.9 +- 1.7	13 RP
70.071 +- 4.011	9.2 +- 4.62	14 FP
69.89 +- 1.392	9.4 +- 2.76	9 TC
69.536 +- 2.228	10.3 +- 2.24	2 H
68.994 +- 1.999	11 +- 2.37	19 DS
67.558 +- 4.52	11.5 +- 4.1	17 ER
67.214 +- 3.001	12.2 +- 4.09	3D
66.675 +- 2.687	13.2 +- 3.37	18 CT
65.769 +- 3.751	14.1 +- 3.18	6 Ed
65.945 +- 3.438	14.4 +- 2.8	8 An
64.153 +- 3.952	14.6 +- 3.88	7 Na
63.792 +- 3.705	15.4 +- 3.14	4 M
29.032 +- 0.692	19 +- 0	1 E

Se mantiene en gran medida el orden de relevancia de las variables tanto con cómo sin validación cruzada. Las que eran muy relevantes sin validación cruzada con validación cruzada lo siguen siendo y las que no eran muy relevantes sin validación cruzada, ahora tampoco lo son. Aunque hay excepciones, por ejemplo las variables 3 y 4(MES (M), NACIONALIDAD (Na)) que son muy relevantes sin validación cruzada y poco relevantes sin validación.

6.2.2.6. RelievefF Attribute Eval

- Selección de variables sin validación cruzada:
  15,2,17,16,5,9,11,6,10,13,3,14,4,8,12,19,1,7,18 : 19
- Selección de variables usando validación cruzada

average merit	average rank	attribute
0.316 +- 0.039	1 +- 0	15 CA
0.135 +- 0.025	3.3 +- 0.9	2 H
0.129 +- 0.021	3.5 +- 1.2	17 ER
0.121 +- 0.024	4.1 +- 1.87	5 HO
0.124 +- 0.026	4.1 +- 1.81	16 R
0.104 +- 0.013	5.9 +- 1.04	11 PT
0.09 +- 0.014	7 +- 1.1	9 TC
0.077 +- 0.01	7.9 +- 1.14	6 Ed
0.062 +- 0.017	9.5 +- 1.8	10 TO
0.057 +- 0.018	10.5 +- 2.38	13 RP
0.048 +- 0.025	11.2 +- 2.89	14 FP
0.037 +- 0.012	12.6 +- 1.62	12 F
0.034 +- 0.009	13.1 +- 1.14	3 D
0.035 +- 0.013	13.1 +- 1.58	4 M
0.028 +- 0.004	14 +- 0.77	8 An
0.005 +- 0.009	16.5 +- 0.92	19 DS
0 +- 0	17.2 +- 0.6	1 E
0.005 +- 0.017	17.5 +- 1.2	7 Na
0.011 +- 0.018	18 +- 2.05	18 CT

Se mantiene en gran medida el orden de relevancia de las variables tanto con cómo sin validación cruzada. Las que eran muy relevantes sin validación cruzada con validación cruzada lo siguen siendo y las que no eran muy relevantes sin validación cruzada, ahora tampoco lo son.

6.2.2.7. Symmetrical Uncert Attribute Eval

- Selección de variables sin validación cruzada:

  14,4,15,9,1,10,11,13,18,2,16,3,17,6,8,12,7,5 : 18

- Selección de variables
  Aplicación con validación cruzada de 10 grupos:

average merit	average rank	attribute
0.345 - 0.039	1 +- 0	14 CA
0.155 - 0.045	3.2 +- 0.87	15 R
0.142 - 0.021	3.4 +- 0.8	9 TO
0.177 - 0.067	3.9 +- 4.72	4 HO
0.078 +- 0.013	6.1 +- 1.37	1 H
0.072 +- 0.007	6.5 +- 1.28	10 PT
0.066 +- 0.013	7.3 +- 1.73	11 F
0.061 +- 0.021	8.1 +- 2.47	13 FP
0.045 +- 0.004	9.5 +- 1.02	2 D
0.047 +- 0.012	10.3 +- 1.79	18 DS
0.04 +- 0.022	11.1 +- 3.11	16 ER
0.032 +- 0.006	12.1 +- 0.83	3 M
0.026 +- 0.011	12.8 +- 1.94	6 Na
0.028 +- 0.017	12.8 +- 2.48	17 CT
0.019 +- 0.004	14.4 +- 1.02	8 TC
0.011 +- 0.016	15 +- 2.41	12 RP
0.016 +- 0.048	16.2 +- 4.12	5 Ed
0 +- 0	17.3 +- 0.46	7 An

Destacan las variables Comunidad Autónoma y Régimen, aunque como se ha indicado se duda de que exista suficiente representación para cada tipo de suceso, por lo que tal vez resulte beneficioso excluirlas del análisis.

6.3. APLICACIÓN: LATENT SEMANTIC ANÁLISIS

La variable encuesta es la variable latente

6.4. APLICACIÓN COMPONENTES PRINCIPALES

Valores propios (eigenvalue)	Proporción varianza explicada(proportion)	Proporción acumulada(cumulative)	Componentes principales
7.19233	0.06365	0.06365	1
5.05384	0.04472	0.10837	2
4.54358	0.04021	0.14858	3
4.27477	0.03783	0.14858	4
3.97975	0.03522	0.18641	5
…	…	….	….
1.01639	0.00899	0.91005	51
1.01639	0.00899	0.91905	52
1.01639	0.00899	0.92804	53
1.01639	0.00899	0.93704	54
1.01639	0.00899	0.94603	55
1.01639	0.00899	0.95503	56

En este caso, tendríamos que quedarnos con 56componentes principales para poder explicar un 95,5% de la varianza.

Antes de realizar el análisis teníamos 116 variables. Contando que cada categoría es una nueva variable.

7. PROBLEMA 2:CREDIT-SCORING

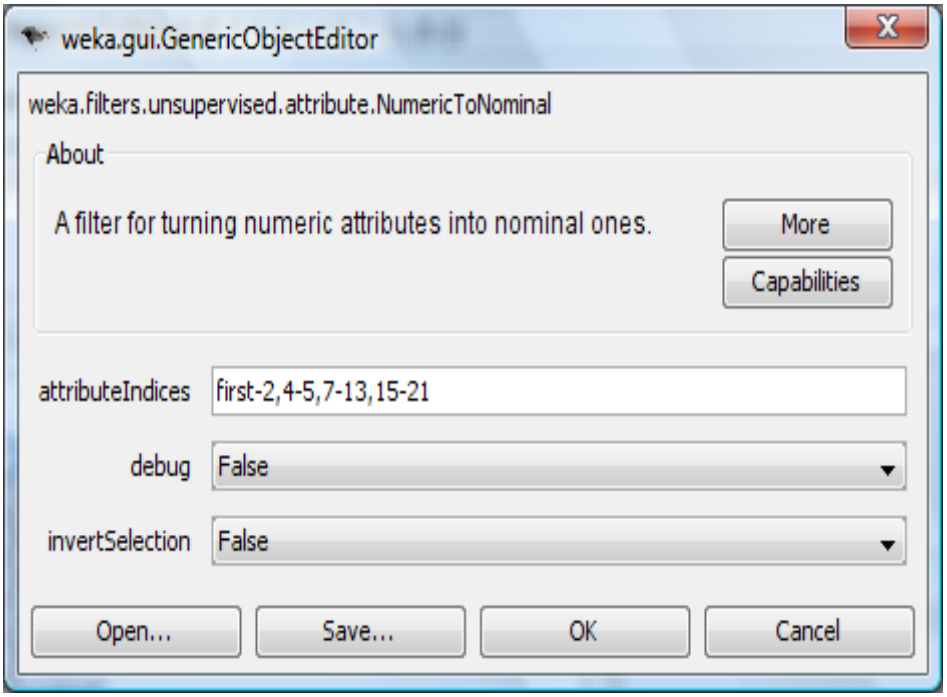
Los bancos están interesados en saber si los clientes le van a pagar el crédito o no.

El objetivo de credit-scoring es modelar o predecir la probabilidad de que un cliente con ciertas características esté considerado como un potencial riesgo.

Nuestro conjunto de datos consiste en 1000 personas que tienen un crédito en un banco alemán. Para cada cliente la binanaria variable respuesta "creditability" está disponible . Además, fueron registradas 20 covariables que influyen en la variable respuesta.

A la hora de tratar con este conjunto de datos tenemos que cambiar variables que están como numéricas y ponerlas como nominales.

Para ello usamos el filtro no supervisado atributo Numerical to nominal.



Las variables: Laufzeit, Hoehe y Alter son las únicas numéricas

Descripción de las variables

1. Laufkont: balance de la cuenta corriente
2. Laufzeit: duración en meses
3. moral : pagamiento de créditos previos
4. Verw: propósito del crédito
5. hoehe: cantidad de crédito en "Deutsche Mark" (metric)
6. sparkont: valores de los ahorros
7. Beszeit: Has estado empleado durante….

8. rate: plazo en % de ingresos seguros
9. famges :estado social/sexo
10. buerge : nuevos deudores/fiadores
11. Wohnzeit: viviendo en una casa familiar durante…
12. verm:posesiones
13. alter : edad en años
14. weitkred : nuevos créditos rápidos
15. Wohn: tipo de apartamento
16. bishkred: número de creditos previos pedidos a este banco(incluidos los créditos rápidos)
17. beruf : ocupación
18. pers : número de personas que mantienes
19. telef : ¿tienes teléfono?
20. Gastarb:¿trabajador extranjero?
21. kredit : buen crédito o no

7.1. APLICACIÓN: ALGORITMOS DE PONDERACIÓN BINARIA

En este apartado comparamos los resultados arrojados en función de los evaluadores de conjuntos de atributos utilizados, utilizando la muestra de entrenamiento.

Método	Solución
Cfs.Subset.Eval	1,2,3 : 3
Classifier.Subset.Eval	4 1
Consisitency.Subset.Eval	1,2,3,4,7,8,9,11,12,13,14 : 11
Filtered.Subset.Eval	5 1

7.1.1. Comparación de resultados sin y con validación cruzada:

7.1.1.1. Cfs.Subset.Eval

- Selección de atributos sin validación cruzada:

1,2,3 : 3

1.Laufkont: balance de la cuenta corriente

2.Laufzeit: duración en meses

3.moral : pagamiento de créditos previos

- Selección de atributos con validación cruzada:

number of folds (%)	attribute
10(100 %)	1 laufkont
9(90 %)	2 laufzeit
10(100 %)	3 moral
0(0 %)	4 verw
4(40 %)	5 hoehe
4(40 %)	6 sparkont
0(0 %)	7 beszeit
0(0 %)	8 rate
0(0 %)	9 famges
0(0 %)	10 buerge
0(0 %)	11 wohnzeit
0(0 %)	12 verm
0(0 %)	13 alter
0(0 %)	14 weitkred
5(50 %)	15 wohn
0(0 %)	16 bishkred
0(0 %)	17 beruf
0(0 %)	18 pers
0(0 %)	19 telef
0(0 %)	20 gastarb

La variable que está incluida en todos los grupos es 1 y 3. Seguida por la variable 2. La variable 15 está en la mitad de los grupos y las variables 5 y 6 están en el 40% de los grupos.
1.Laufkont: balance de la cuenta corriente
2.Laufzeit: duración en meses
3.moral : pagamiento de créditos previos
5.hoehe: cantidad de crédito en "Deutsche Mark" (metric)
6.sparkont: valores de los ahorros
15.Wohn: tipo de apartamento

7.1.1.2. Clasificier.Subset.Eval

- Selección de atributos sin validación cruzada:

  4 : 1

4.Verw: propósito del crédito

- Selección de atributos con validación cruzada:

number of folds (%)	attribute
0(0 %)	1 laufkont
0(0 %)	2 laufzeit
0(0 %)	3 moral
10(100 %)	4 verw
0(0 %)	5 hoehe
0(0 %)	6 sparkont
0(0 %)	7 beszeit
0(0 %)	8 rate
0(0 %)	9 famges
0(0 %)	10 buerge
0(0 %)	11 wohnzeit
0(0 %)	12 verm
0(0 %)	13 alter
0(0 %)	14 weitkred
0(0 %)	15 wohn
0(0 %)	16 bishkred
0(0 %)	17 beruf
0(0 %)	18 pers
0(0 %)	19 telef
0(0 %)	20 gastarb

La única variable representada es 4 y además está en todos los grupos.

4.Verw: propósito del crédito

7.1.1.3. Consistency.Subset.Eval

- Selección de atributos sin validación cruzada:

  1,2,3,4,7,8,9,11,12,13,14 : 11

1.Laufkont: balance de la cuenta corriente
2.Laufzeit: duración en meses
3.moral : pagamiento de créditos previos
4.Verw: propósito del crédito
7.Beszeit: Has estado empleado durante….

8.rate: plazo en % de ingresos seguros

9.famges :estado social/sexo

11.Wohnzeit: viviendo en una casa familiar durante…

12.verm:posesiones

13.alter : edad en años

14.weitkred : nuevos créditos rápidos

- Selección de atributos con validación cruzada:

number of folds (%)	attribute
10(100 %)	1 laufkont
5(50 %)	2 laufzeit
10(100 %)	3 moral
10(100 %)	4 verw
2(20 %)	5 hoehe
8(80 %)	6 sparkont
10(100 %)	7 beszeit
10(100 %)	8 rate
9(90 %)	9 famges
0(0 %)	10 buerge
10(100 %)	11 wohnzeit
10(100 %)	12 verm
3(30 %)	13 alter
9(90 %)	14 weitkred
1(10 %)	15 wohn
0(0 %)	16 bishkred
2(20 %)	17 beruf
0(0 %)	18 pers
3(30 %)	19 telef
0(0 %)	20 gastarb

Las variables que están incluidas en todos los pliegues son 1, 3, 4, 7, 8, 11 y 12. Las variables que estás incluidas en el 90% de los pliegues es 9 y 14. La variable 6 está incluida en el 80% de los pliegues. La variable 2 está en la mitad de los grupos de validación cruzada (pliegue). Las variables 13 y 19 están en el 30%de los grupos. Las variables 5 y 17 están en el 20% de los grupos y la variable 15 está en el 10% de los pliegues.

1.Laufkont: balance de la cuenta corriente
2.Laufzeit: duración en meses
3.moral : pagamiento de créditos previos
4.Verw: propósito del crédito
5.hoehe: cantidad de crédito en "Deutsche Mark" (metric)
6.sparkont: valores de los ahorros
7.Beszeit: Has estado empleado durante….
8.rate: plazo en % de ingresos seguros
9.famges :estado social/sexo
11.Wohnzeit: viviendo en una casa familiar durante…
12.verm:posesiones
13.alter : edad en años
14.weitkred : nuevos créditos rápidos
15.Wohn: tipo de apartamento
17.beruf : ocupación
18.pers : número de personas que mantienes
20.Gastarb:¿trabajador extranjero?

7.1.1.4. Filtered.Subset.Eval

- Selección de atributos sin validación cruzada:
  5: 1 hoehe

  5.hoehe: cantidad de crédito en "Deutsche Mark" (metric)

- Selección de atributos con validación cruzada:

number of folds (%)	attribute
0(0 %)	2 laufkont
0(0 %)	3 laufzeit
0(0 %)	4 moral
0(0 %)	5 verw
10(100 %)	6 hoehe
0(0 %)	7 sparkont
0(0 %)	8 beszeit
0(0 %)	9 rate
0(0 %)	10 famges
0(0 %)	11 buerge
0(0 %)	12 wohnzeit
0(0 %)	13 verm
0(0 %)	14 alter
0(0 %)	15 weitkred
0(0 %)	16 wohn
0(0 %)	17 bishkred
0(0 %)	18 beruf
0(0 %)	19 pers
0(0 %)	20 telef
0(0 %)	21 gastarb

La 6 es la única variable está en los grupos y además en todos.

6.sparkont: valores de los ahorros

7.1.1.5. Symmetrical Uncert Attribute Set Eval

Aplicación del algoritmo de selección de variables Symmetrical Uncert Attribute Set Eval a la base de datos kredit scoring. Con un método de búsqueda FCBF.

Ranking de variables:

J	SU(j,Class)	I	SU(i,j).
5	0.1541319	*	
1	0.0706128	5	0.29302597137504743
3	0.0336406	5	0.28105379041069467
2	0.0272728	5	0.5338536344168907
6	0.0218874	5	0.2768803889424587
13	0.0148849	5	0.6787249540567784
4	0.0140326	5	0.40672815260188966
15	0.0129631	5	0.19750414281621054
12	0.0120076	5	0.31235781263020473
20	0.0104951	5	0.04192032471142956
14	0.0102839	5	0.15079780578334376
7	0.0086299	5	0.34061437928323157
10	0.0067575	5	0.10007454116923312
9	0.005644	5	0.2529655798834511
8	0.002953	5	0.2930073905296647
16	0.0019627	5	0.1957164237540518
17	0.0011656	5	0.2365371407256128
19	0.0010392	5	0.1701969211477855
11	0.0003984	5	0.29807604704550356
18	0.0000087	5	0.11211704521816854

Ranking de atributos:

0.154 5 hoehe

Variables selecionadas: 5 : 1

Los primeros puestos del ranking los ocupan las variables hoehe, laufkont, moral, laufzeit y sparkont.

Nos encontramos con que el algoritmo sólo selecciona una variable (hoehe), que puede deberse a que las demás variables son redundantes o están correlacionadas con ésta.

Aplicación del algoritmo con validación cruzada de 10 grupos:

average merit	average rank	attribute
0.157 +- 0.001	1 +- 0	5 hoehe

Se ha obtenido de nuevo el mismo resultado con la variable hoehe como única seleccionada.

De la aplicación de Symmetrical Uncert Subset Eval podemos considerar que la variable hoehe debe incluirse en un buen subconjunto de variables que permita conocer mejor el comportamiento de la variable clase. Podrían incluirse más variables en el subconjunto pero hay que ser cuidadosos debido a que pudiesen resultar redundantes.

7.1.1.6. Wrapper Subset Eval

Se ha aplicando sobre los datos el algoritmo Wrapper Subset Eval, con Naive Bayes como clasificador y Best First como motor de búsqueda. Obteniéndose los siguientes resultados con un modo de selección de variables sobre toda la muestra de entrenamiento.

El número de subconjuntos evaluados es de 181 y el mérito del mejor subconjunto encontrado es 0.238.

Variables seleccionadas: 1,3,6,9,10,12,19 : 7

Laufkont (balance de la cuenta corriente)
Moral (pago de créditos previos)
Sparkont (valor de los ahorros)
Famges (Estado civil / sexo)
Buerge (nuevos deudores / fiadores)
Verm (Propósito del crédito)
Telef (Tener teléfono)

A continuación, se presentan los resultados de la aplicación del algoritmo pero con un modo de selección de variables consistente en una validación cruzada con diez grupos.

number of folds (%)	attribute
10(100 %)	1 laufkont
6(60 %)	2 laufzeit
10(100 %)	3 moral
1(10 %)	4 verw
0(0 %)	5 hoehe
8(80 %)	6 sparkont
5(50 %)	7 beszeit
4(40 %)	8 rate
6(60 %)	9 famges
10(100 %)	10 buerge
3(30 %)	11 wohnzeit
7(70 %)	12 verm

0(0 %) 13 alter

0(0 %) 14 weitkred

0(0 %) 15 wohn

2(20 %) 16 bishkred

2(20 %) 17 beruf

1(10 %) 18 pers

1(10 %) 19 telef

9(90 %) 20 gastarb

Han sido seleccionadas en todos los grupos las variables laufkont (balance de la cuenta corriente), moral (pago de créditos previos) y buerge (nuevos deudores / fiadores), las tres habían sido seleccionadas en la aplicación anterior del algoritmo. En un 90% de los grupos aparece la gastarb (trabajador nacional/extranjero) que no fue seleccionada sobre toda la muestra de entrenamiento. Las siguientes variables presentes en mayor porcentaje de grupos son sparkont (80%) y verm (70%) que ya eran incluidas en la selección sin validación cruzada. Por último, decir que las variables Famges y Telef que fueron seleccionadas en la primera aplicación están presentes en un 60% y un 10%, respectivamente; de los grupos de la validación cruzada.

Aplicando Wrapper Subset Eval a la base de datos Kredit Scoring, se ha obtenido que un buen subconjunto para explicar la variable kredit debería incluir las variables: Laufkont, Moral y Buerge. Y, puede considerarse la inclusión de Gastarb, Sparkont y Verm.

7.2. APLICACIÓN: ALGORITMOS DE PONDERACIÓN CONTÍNUA

En este apartado comparamos la solución dada por diferentes métodos con todas las variables y usando toda la muestra de entrenamiento. Además de esto también se muestran la diferencia que hay entre las soluciones.

Usando validación cruzada y sin usar validación cruzada.

En los casos en que la diferencia sea notable serán mencionados también en el documento.

Método	Solución
ChiSquared Attribute Eval	1,3,2,6,4,5,12,15,7,13,14,9,20,10,8,16,17,19,11,18
One Attribute Eval	3,2,9,11,10,6,8,7,18,17,20,19,12,13,16,14,15,4,1,5
RelievefF Attribute Eval	1,3,4,9,7,6,12,11,19,8,17,2,16,10,18,5,13,14,15,20
Filtered Attribute Eval	1,5,2,13,3,6,4,12,7,15,14,9,20,10,8,16,17,19,11,18
Gain Ratio Attribute Eval	5,1,20,3,2,6,15,14,4,10,12,13,7,9,8,16,19,17,11,18
Info Gain Ratio Attribute Eval	5,1,2,13,3,6,4,12,7,15,14,9,20,10,8,16,17,19,11,18
Symmetrical Uncert Attribute Eval	5,1,3,2,6,13,4,15,12,20,14 : 11

7.2.1. Comparación de resultados con y sin validación cruzada

7.2.1.1. Chi Square Attribute Eval

Selección de atributos sin validación cruzada:

1,3,2,6,4,5,12,15,7,13,14,9,20,10,8,16,17,19,11,18 : 20

Slección de atributos:

average merit	average rank	attribute
111.714 +- 6.025	1 +- 0	1 laufkont
55.838 +- 3.599	2 +- 0	3 moral
37.093 +- 6.289	3.7 +- 1	2 laufzeit
32.855 +- 3.986	4.1 +- 0.7	6 sparkont
30.862 +- 1.926	4.6 +- 0.8	4 verw
21.644 +- 3.2	7.2 +- 0.87	12 verm
24.281 +- 9.161	7.3 +- 4.38	5 hoehe
17.194 +- 4.503	8.5 +- 1.5	7 beszeit
17.105 +- 2.837	8.8 +- 0.87	15 wohn
11.769 +- 1.962	10.8 +- 0.87	14 weitkred
9.131 +- 2.204	11.7 +- 1.35	9 famges
11.343 +- 7.687	11.9 +- 5.36	13 alter
6.12 +- 1.196	13.2 +- 0.98	20 gastarb
6.133 +- 1.476	13.2 +- 0.75	10 buerge
5.101 +- 1.021	14.1 +- 1.14	8 rate
2.743 +- 1.032	16 +- 0.89	16 bishkred
1.956 +- 0.541	16.7 +- 0.78	17 beruf
1.237 +- 0.45	17.7 +- 1	19 telef
1.015 +- 0.636	18 +- 1.34	11 wohnzeit
0.144 +- 0.183	19.5 +- 0.5	18 pers

Tanto con validación cruzada como sin ella se mantiene el orden de relevacia de las variables. Las que con el conjunto de entrenamiento eran relevantes lo siguen siendo con validación cruzada.

7.2.1.2. Filtered Attribute Eval

- Selección de variables sin validación cruzada: 5,1,2,13,3,6,4,12,7,15,14,9,20,10,8,16,17,19,11,18 : 20
- Selección de variables con validación cruzada:

Average merit	Average Rank	Atributos
0.829 +- 0.005	1 +- 0	5 hoehe
0.095 +- 0.005	2 +- 0	1 laufkont
0.066 +- 0.003	3 +- 0	2 laufzeit
0.05 +- 0.003	4.1 +- 0.3	13 alter
0.044 +- 0.003	4.9 +- 0.3	3 moral
0.029 +- 0.004	6.3 +- 0.46	6 sparkont
0.026 +- 0.002	6.7 +- 0.46	4 verw
0.017 +- 0.003	8.3 +- 0.46	12 verm
0.014 +- 0.004	9.3 +- 1	7 beszeit
0.013 +- 0.002	9.5 +- 0.5	15 wohn
0.009 +- 0.001	11.4 +- 0.92	14 weitkred
0.007 +- 0.002	12.2 +- 1.08	9 famges
0.006 +- 0.001	12.9 +- 0.94	20 gastarb
0.005 +- 0.001	13.8 +- 0.98	10 buerge
0.004 +- 0.001	14.7 +- 0.64	8 rate
0.002 +- 0.001	16.4 +- 0.8	16 bishkred
0.002 +- 0	17.1 +- 0.83	17 beruf
0.001 +- 0	18.1 +- 0.94	19 telef
0.001 +- 0.001	18.4 +- 1.02	11 wohnzeit
0 +- 0	19.9 +- 0.3	18 pers

7.2.1.3. Gain Ratio Attribute Eval:

- Selección de variables sin validación cruzada:
  5,1,20,3,2,6,15,14,4,10,12,13,7,9,8,16,19,17,11,18 : 20
- Selección de variables con validación cruzada:

Average merit	Average Rank	Atributos
0.086 +- 0	1 +- 0	5 hoehe
0.053 +- 0.003	2 +- 0	1 laufkont
0.026 +- 0.002	3.3 +- 0.46	3 moral
0.026 +- 0.006	3.7 +- 0.46	20 gastarb
0.018 +- 0.001	5.3 +- 0.46	2 laufzeit
0.017 +- 0.002	5.7 +- 0.46	6 sparkont
0.012 +- 0.002	8 +- 1.1	15 wohn
0.011 +- 0.002	8.8 +- 1.6	14 weitkred
0.01 +- 0.001	9.6 +- 1.36	4 verw
0.01 +- 0.001	9.7 +- 1.42	13 alter
0.009 +- 0.002	10.3 +- 2.15	10 buerge
0.009 +- 0.001	11.3 +- 1.55	12 verm
0.006 +- 0.002	12.6 +- 1.02	7 beszeit
0.005 +- 0.001	13.7 +- 0.64	9 famges
0.002 +- 0	15.4 +- 0.49	8 rate
0.002 +- 0.001	16.1 +- 1.04	16 bishkred
0.001 +- 0	17.1 +- 0.94	17 beruf
0.001 +- 0	17.8 +- 0.87	19 telef
0 +- 0	19.1 +- 0.83	11 wohnzeit
0 +- 0	19.5 +- 0.81	18 pers

En este caso observamos que hay cambios importantes en el orden que siguen las variables en función de la explicación que aportan al modelo. Las variables 20 y 3 tienen el orden inverso con validación cruzada que sin ella. La variable 13 tiene más relevancia con validación cruzada, mientras que en el caso de la variable 19 ocurre lo contrario (es más relevante para un análisis sin validación cruzada).

7.2.1.4. Info Gain Attribute Eval

- Selección de variables sin validación cruzada:
- 5,1,2,13,3,6,4,12,7,15,14,9, 20,10,8,16,17,19,11,18 : 20
- Selección de variables con validación cruzada:

Average merit	Average Rank	Atributos
0.829 +- 0.005	1 +- 0	5 hoehe
0.095 +- 0.005	2 +- 0	1 laufkont
0.066 +- 0.003	3 +- 0	2 laufzeit
0.05 +- 0.003	4.1 +- 0.3	13 alter
0.044 +- 0.003	4.9 +- 0.3	3 moral
0.029 +- 0.004	6.3 +- 0.46	6 sparkont
0.026 +- 0.002	6.7 +- 0.46	4 verw
0.017 +- 0.003	8.3 +- 0.46	12 verm
0.014 +- 0.004	9.3 +- 1	7 beszeit
0.013 +- 0.002	9.5 +- 0.5	15 wohn
0.009 +- 0.001	11.4 +- 0.92	14 weitkred
0.007 +- 0.002	12.2 +- 1.08	9 famges
0.006 +- 0.001	12.9 +- 0.94	20 gastarb
0.005 +- 0.001	13.8 +- 0.98	10 buerge
0.004 +- 0.001	14.7 +- 0.64	8 rate
0.002 +- 0.001	16.4 +- 0.8	16 bishkred
0.002 +- 0	17.1 +- 0.83	17 beruf
0.001 +- 0	18.1 +- 0.94	19 telef
0.001 +- 0.001	18.4 +- 1.02	11 wohnzeit
0 +- 0	19.9 +- 0.3	18 pers

En este caso, tanto para el análisis con validación como para el análisis sin validación cruzada las variables/atributos siguen el mismo orden, es decir, son igual de relevantes en un caso y en otro.

7.2.1.5. OneR Atttribute Eval

- Selección de variables:

  3,2,9,11,10,6,8,7,18,17,20,19,12,13,16,14,15,4,1,5 : 20

- Selección de variables:

average merit	average rank	attribute
71.633 +- 0.258	1 +- 0	3 moral
70 +- 0	4.1 +- 0.3	11 wohnzeit
70.511 +- 0.495	4.3 +- 5.02	2 laufzeit
69.911 +- 0.267	5.6 +- 4.54	9 famges
70 +- 0	5.9 +- 0.3	6 sparkont
70 +- 0	7.1 +- 1.45	7 beszeit
70 +- 0	7.3 +- 0.64	8 rate
70 +- 0	8.6 +- 5.41	12 verm
70 +- 0	9.3 +- 0.64	18 pers
69.811 +- 0.233	9.5 +- 6.64	10 buerge
70 +- 0	9.9 +- 0.7	17 beruf
70 +- 0	11.3 +- 0.64	20 gastarb
70 +- 0	11.9 +- 0.7	19 telef
70 +- 0	13.2 +- 0.75	14 weitkred
70 +- 0	15 +- 0.89	15 wohn
69.122 +- 0.658	15.7 +- 5.08	1 laufkont
69.778 +- 0.131	16.1 +- 0.83	16 bishkred
69.511 +- 0.291	16.9 +- 1.64	13 alter
69.256 +- 0.636	17.3 +- 1.35	4 verw
66.122 +- 0.817	20 +- 0	5 hoehe

Se mantiene en gran medida el orden de relevancia de las variables tanto con cómo sin validación cruzada. Las que eran muy relevantes sin validación cruzada con validación cruzada lo siguen siendo y las que no eran muy relevantes sin validación cruzada, ahora tampoco lo son. Aunque hay excepciones como la variable 10 (buerge : nuevos deudores/fiadores) muy relevante sin validación cruzada y poco relevante con validación cruzada.

7.2.1.6. RelievefF Attribute Eval

- Selección de atributos:

  1,3,4,9,7,6,12,11,19,8,17,2,16,10,18,5,13,14,15,20 : 20

- Selección de variables

average merit	average rank	attribute
0.153 +- 0.011	1 +- 0	1 laufkont
0.066 +- 0.005	2 +- 0	3 moral
0.048 +- 0.004	3.1 +- 0.3	4 verw
0.039 +- 0.005	4.4 +- 0.8	9 famges
0.036 +- 0.004	5.4 +- 0.92	7 beszeit
0.032 +- 0.004	6 +- 1.18	6 sparkont
0.029 +- 0.005	6.7 +- 1.42	12 verm
0.024 +- 0.004	8.8 +- 2.09	11 wohnzeit
0.022 +- 0.005	9.4 +- 1.74	8 rate
0.019 +- 0.003	10.8 +- 1.66	19 telef
0.018 +- 0.004	11.4 +- 2.33	17 beruf
0.018 +- 0.002	11.5 +- 1.75	2 laufzeit
0.016 +- 0.002	12.5 +- 1.57	16 bishkred
0.016 +- 0.003	12.7 +- 2	10 buerge
0.012 +- 0.003	15.6 +- 1.85	18 pers
0.011 +- 0.001	16.2 +- 0.6	5 hoehe
0.01 +- 0.001	17 +- 0.89	13 alter
0.009 +- 0.003	17.5 +- 1.57	15 wohn
0.008 +- 0.003	18.2 +- 1.47	14 weitkred
0.003 +- 0.001	19.8 +- 0.4	20 gastarb

Se mantiene en gran medida el orden de relevancia de las variables tanto con cómo sin validación cruzada. Las que eran muy relevantes sin validación cruzada con validación cruzada lo siguen siendo y las que no eran muy relevantes sin validación cruzada, ahora tampoco lo son.

7.2.1.7. Symmetrical Uncert Attribute Eval

Aplicación del algoritmo Symmetrical Uncert Attribute Eval con método de búsqueda Ranker y sobre toda la muestra de entrenamiento. Se ha seleccionado un valor de threshold de 0.01 para seleccionar un subconjunto de las variables originales.

Ranking de variables:

0.1541 5 hoehe
0.0706 1 laufkont
0.0336 3 moral
0.0273 2 laufzeit
0.0219 6 sparkont
0.0149 13 alter
0.014 4 verw
0.013 15 wohn
0.012 12 verm
0.0105 20 gastarb
0.0103 14 weitkred

Variables seleccionadas: 5,1,3,2,6,13,4,15,12,20,14 : 11

En el ranking destacan las variables laufkont, moral, laufzeit y sparkont. Se ha seleccionado un total de once variables que superan el umbral de 0.01.

Aplicación del algoritmo Symmetrical Uncert Attribute Eval con validación cruzada como modo de selección de variables y un threshold de 0.01.

average merit	average rank	attribute
0.157 +- 0.001	1 +- 0	5 hoehe
0.071 +- 0.004	2 +- 0	1 laufkont
0.034 +- 0.002	3 +- 0	3 moral
0.028 +- 0.001	4 +- 0	2 laufzeit
0.022 +- 0.003	5 +- 0	6 sparkont
0.016 +- 0.001	6.2 +- 0.4	13 alter
0.014 +- 0.001	7.7 +- 1	4 verw
0.013 +- 0.002	8.4 +- 1.56	15 wohn
0.012 +- 0.002	9.1 +- 1.04	12 verm
0.011 +- 0.003	10.4 +- 1.62	20 gastarb
0.01 +- 0.002	10.5 +- 1.12	14 weitkred
0.009 +- 0.002	11.2 +- 1.94	7 beszeit
0.007 +- 0.002	12.8 +- 0.98	10 buerge
0.006 +- 0.001	13.7 +- 0.46	9 famges
0.003 +- 0.001	15.1 +- 0.3	8 rate
0.002 +- 0.001	16.4 +- 0.8	16 bishkred
0.001 +- 0	17.1 +- 0.83	17 beruf
0.001 +- 0	17.9 +- 0.83	19 telef
0.001 +- 0	18.8 +- 1.08	11 wohnzeit
0 +- 0	19.7 +- 0.64	18 pers

En la selección con validación cruzada con diez grupos destacan las variables:

Hoehe, laufkont, moral, laufzeit y sparkont.

De la selección de variables realizada con Symmetrical Uncert Attribute Eval podemos considerar un buen subconjunto el formado por las variables: hoehe, laufkont, moral, laufzeit y sparkont. Que pueden resultar útiles para determinar el valor que tomará un nuevo individuo en la variable clase kredit. Además, puede considerarse la inclusión en este subconjunto de las variables alter y verw.

7.3. APLICACIÓN: LATENT SEMANTIC ANÁLISIS

Se reduce la dimensión: pasamos de 21 a 2

Laufkont: balance de la cuenta corriente

Laufzeit: duración en meses

Son las variables escogidas

7.4. APLICACIÓN COMPONENTES PRINCIPALES

Valores propios(eigenvalue)	Proporción varianza explicada(proportion)	Proporción acumulada(cumulative)	Componentes principales
4.03815	0.05938	0.05938	1
3.30805	0.04865	0.10803	2
2.71972	0.04	0.1480	3
…	….	….	….
0.72783	0.0107	0.93203	45
0.70462	0.01036	0.94239	46
0.67681	0.00995	0.95234	47

En este caso, tendríamos que quedarnos con 47 componentes principales para poder explicar un 95,23% de la varianza.

Pasamos de 128 variables a 47 componentes principales.

8. APLICACIONES EN BIOINFORMÁTICA

8.1. SELECCIÓN DE VARIABLES PARA UN ANÁLISIS DE SECUENCIAS

El análisis de secuencias tiene una larga tradición en bioinformática. En este contexto de selección de variables, se pueden distinguir dos tipos de problemas:

Content analysis (análisis de contenidos), está enfocado hacia las características de una secuencia, codifica la tendencia de las proteínas o satisfacción de una cierta función biológica.

Signal analysis (análisis de señales), se centra en la identificación de importantes elementos en la secuencia, como son los elementos estructurales de un gen o elementos regulares.

8.1.1. Content analysis

La predicción de subsecuencias que codifican las proteínas ha sido un foco de interés desde el inicio de la bioinformática. Puesto que, muchas características pueden ser extraídas de una secuencia, y la dependencia más importante ocurre entre posiciones adyacentes; para ello se desarrollaron muchas variaciones en los modelos de Markov. Tratar con grandes cantidades de posibles características, introdujo el modelo interpolado de Markov (INM), el cual usó la interpolación entre diferentes ordenes del modelo de Markov con muestras de pequeño tamaño, y un método filtro(“filter method”) que selecciona solo las características relevantes.

Además, se extendió el INM a la situación de tratar con las dependencias de las características no adyacentes, resultando el modelo contextualmente interpolado (ICM), el cual mezcla un árbol de decisión Bayesiano con un método filter (Chi-square) para obtener las características relevantes. Recientemente, las técnicas de selección de variables(FS) para codificar la predicción potencial fueron obtenidas combinando diferentes medidas del código de la predicción potencial, y luego se usó la aproximación del filtro multivariante “blanket” de Markov (MBF) para retener solo las relevantes.

Una segunda clase de técnicas buscan la función predicción de las proteínas de la secuencia. Esto combina un algoritmo genético con un test Gamma para obtener el subconjunto de variables para la clasificación de grandes cantidades de subunidades de rRNA, inspirado en la rebúsqueda de técnicas FS para conseguir importantes subconjuntos de amino ácidos que describan la clase funcional de la proteína. Una interesante técnica usa el núcleo escalado para la SVM como un método que calcula el peso de las variables, y remueve las subsecuencias de variables con menor peso.

El uso de las técnicas FS en el ámbito del análisis de secuencias es introducido en numerosas aplicaciones recientes.

8.1.2. Signal analysis

Muchas metodologías de análisis de secuencias que reconocen más o las señales conservadas en la secuencia, representando sitios para varias proteínas o proteínas complejas. Una aproximación para encontrar motivos de reglamentación, es relacionar los motivos a los niveles de expresiones genéticas usando una regresión. La selección de variables puede ser usada para buscar los motivos que maximizan el ajuste de un modelo de regresión. Una clasificación aproximada es escogida para discriminar motivos. El método usa el umbral de números de errores de clasificación para valorar los genes mediante la relevancia en la clasificación. Del valor TNoM, se obtiene un p-valor que representa la significación de cada modelo. Los motivos son clasificados de acuerdo con su p-valor.

Otra línea de búsqueda es transformar en el contexto del conjunto de genes predicho, donde estructurales elementos como tales se trasladaron del sitio inicial (TIS) y los sitios de empalme son modelados como problemas específicos de clasificación. El problema de selección de variables para el reconocimiento de elementos estructurales. El problema de predicción de los sitios de empalme se resuelve combinando el método backward junto con el criterio de evaluación SVM para valorar la relevancia de las variables. Una estimación de la distribución del algoritmo fue usada para ganar más perspicacia de las relevantes características en la predicción de los sitios de embarque.

En el futuro, se espera que las técnicas de selección de variables mejoren las técnicas de predicción, las cuales identifican las relevantes variables relacionadas con los sitios de empalme y la alternativa TIS.

8.2. SELECCIÓN DE VARIABLES APLICADO AL ANÁLISIS DE MICROARRAY

Durante la última década, la aparición de conjuntos de datos de microarrays ha estimulado una nueva línea de investigación en bioinformática. El conjunto de datos de microarrays(“microarrays data”) constituye un gran desafío para las técnicas de cálculo, debido a su gran dimensionalidad (hasta varias decenas de miles de genes) y a su pequeño tamaño de muestra. Además, las complicaciones experimentales, como el ruido y la variabilidad hacen del análisis del conjunto de datos de microarrays un dominio emocionante.

Con el fin de hacer frente a estas características particulares del análisis del conjunto de datos de microarrays, se necesitaba emplear técnicas de reducción de la dimensión y pronto su aplicación se convirtió en una de estándar de facto en el campo. Para ello se han construido nuevas metodologías y adaptado las conocidas FS.

8.2.1. El paradigma del filtro univariado: simple pero eficiente

Debido a la alta dimensionalidad de la mayoría de los análisis de microarrays, las técnicas que han captado el mayor interés son las técnicas FS por su eficiencia y rápidez como los métodos de filtro univariantes. La prevalencia de estas técnicas univariantes han dominado el campo, y hasta ahora, se comparan las evaluaciones de diferentes clasificaciones y técnicas FS sobre el conjuntos de datos de microarrays de ADN centrados únicamente en el caso univariante. Este interés por la aproximación univariante se puede explicar por varias razones:

• La salida proporcionada por el ranking univariante de características es intuitiva y fácil de entender;

• el gen de la salida de clasificación podría cumplir los objetivos y las expectativas de los expertos en el campo biológico, quienes esperan tener el resultado de la secuencia validada mediante técnicas de laboratorio o explorar búsquedas bibliográficas. Los expertos no tienen la necesidad de emplear técnicas de selección que tengan en cuenta las interacciones entre los genes;

• la falta de conocimiento posible de los expertos en el campo de la expresión genética de la existencia de técnicas de análisis de datos multivariantes;

• el tiempo de cálculo extra necesario para las técnicas de selección genéticas multivariantes.

Se ha desarrollado una amplia gama de características nuevas o adaptadas técnicas univariantes de clasificación desde entonces se ha desarrollado. Estas técnicas se pueden dividir en dos clases: modelos paramétricos y los métodos de modelado libre.

Los métodos paramétricos asumen una determinada distribución a partir de la cual se han generado muestras(observaciones). El t-test y ANOVA se encuentran entre las técnicas más utilizadas en los estudios de microarrays, aunque se usan de forma básica, posiblemente sin justificación de sus principales hipótesis. Las modificaciones del t-test estándar para poder atender mejor con muestras de pequeño tamaño y el ruido inherente de conjuntos de datos de expresión genética incluye una serie de T o t-test como las estadísticos (se diferencian principalmente en la forma en que se estima la varianza) y una serie de Marcos Bayesiano . Aunque las hipótesis Gausianas han dominado el campo, otros tipos de enfoques paramétricos pueden también ser encontrados en la literatura, tales como modelos de regresión enfoques y modelos de distribución Gamma.

Debido a la incertidumbre acerca la distribución real subyacente de escenarios de expresión de muchos genes, y las dificultades para validar las suposiciones de distribución, debido al pequeño tamaño de muestral, los modelos no paramétricos o modelado de libre han sido ampliamente propuestos como una alternativa atractiva para hacer menos estrictos supuestos de distribución. Muchos de los parámetros del modelo libre, frecuentemente prestados del campo de la estadística, han demostrado su utilidad en muchos estudios de expresión génica, incluyendo la prueba de la suma de los rangos de Wilcoxon, la suma de los cuadrados entre-dentro de las clases(BSS / WSS) y el método de productos de los rangos.

Una clase específica de métodos de modelo libre estima la distribución de referencia del estadístico usando permutaciones aleatorias de los datos, permitiendo el cálculo de un modelo de versión libre asociado a pruebas no paramétricas. Estas técnicas han surgido como una sólida alternativa para hacer frente a las especificidades de los datos de microarrays de ADN. Su principio de permutación, en parte alivia el problema de las muestras de pequeño tamaño en los estudios de microarrays, la mejora de la robustez frente los valores extremos.

También menciona prometedoras métricas de tipo no-paramétrica las cuales, en lugar de tratar de identificar los genes expresados diferencialmente en toda la población(por ejemplo, la comparación de medias de la muestra), son capaces de capturar los genes que son significativamente más desregulados en sólo un subconjunto de muestras. Estos tipos de métodos ofrecen una aproximación más específica para la identificación de marcadores, y puede seleccionar genes que presentan patrones complejos. Además, también señalan la importancia de los procedimientos de para el control de los diferentes tipos de errores que se presentan en este complejo de escenario de múltiples pruebas de miles de gen, con un enfoque especial sobre las contribuciones para el control de la falsa tasa de descubrimiento (FDR).

8.2.2. Hacia modelos más avanzados: el paradigma multivariado para el filtro(filter), técnicas de envoltura e incrustados(wrapper, embedded)

Los métodos de selección univariantes tienen ciertas restricciones y pueden conducir a clasificadores menos precisos, no tienen en cuenta la interacción entre genes. Así, los investigadores han propuesto técnicas que tratan de capturar estas correlaciones entre los genes.

La aplicación de los métodos de filtro multivariante va de simples interacciones de dos variables, hacia soluciones más avanzadas resultado de explorar las interacciones de orden superior, tales como la correlación en base a características selección (CFS) y diversas variantes de la Markov método de filtro de manta (Markov blanquet filter method).

El procedimiento de selección de variables usando **métodos *wrapper*** (métodos de embase)o **métodos *embedded*** (métodos de incrustado) ofrecen una manera alternativa de realizar una selección múltiple subconjunto de genes, incorporando sesgo del clasificador en la búsqueda y ofreciendo así una oportunidad de construir clasificadores más precisos. En el contexto de el análisis de microarrays, la mayoría de los *métodos wrapper* están basados en búsquedas heurísticas de aleatorias, aunque en algunos casos también se usan técnicas de búsqueda secuencial. Es interesante la aproximación *filtro-wrapper* que es considerada como un híbrido ya que cruza una clasificación de genes pre-ordenados univariantemente con *método wrapper* que incrementa gradualmente.

Otra característica de cualquier procedimiento *wrapper* concierna a la función usada para evaluar cada subconjunto de genes encontrados. Esta medida de evaluación es usada para realizar comparaciones con trabajos previos. Sin embargo, los últimos propuestas que abogan por el uso de métodos para la aproximación de las área bajo la curva ROC, o la optimización de modelo LASSO (Contracción menos absoluto y selección de operador). Curvas ROC proporcionan una medida de evaluación interesante, especialmente adecuada a la demanda para la detección de los diferentes tipos de errores en muchos escenarios biomédica.

Los *métodos embedded* tienen la capacidad de usar varios clasificadores para descartar características y por lo tanto proponer un subconjunto de genes discriminativo. Los *métodos embedded* también utilizan el peso de cada característica en los clasificadores lineales como SVMS y de regresión logística. Estos pesos se utilizan para reflejar la importancia de la cada gen de una manera multivariante, y permitir así la eliminación de genes con poco peso.

En parte debido a la gran complejidad computacional de los *métodos wrapper* y en menor grado a las aproximaciones *embedded*, estas técnicas no han recibido tanto interés como las propuestas de filtro. Sin embargo, en la práctica es útil comprobar si es posible reducir el espacio de búsqueda utilizando un método de filtro univariante, y sólo entonces se aplican los *métodos wrapper o embedded,* ajustándonos al tiempo de computación de los recursos disponibles.

8.3. RELACIÓN CON LOS ÁMBITOS PEQUEÑA MUESTRA

Tamaños de muestra pequeños, y su riesgo inherente de la imprecisión y el sobreajuste, plantean un gran desafío para muchos problemas de modelización en bioinformática. En el contexto de la selección de características, dos iniciativas han surgido en respuesta a esta situación experimental: el uso de criterios de evaluación adecuados, y el uso de los modelos de selección de características robustos y estables.

8.3.1. Criterios de evaluación adecuados

Varios trabajos han advertido sobre el gran número de aplicaciones que no llevan a cabo una independiente y honesta validación de la exactitud de los porcentajes reportados. Pues no se realiza un proceso de selección de características externas en el entrenamiento de la regla de clasificación en cada etapa que estimamos la precisión de la estimación.

Además, nuevos métodos de estimación de la exactitud de predicción con características prometedoras, como la estimación de errores reforzado, han surgido para hacer frente a las especificidades de los dominios de la muestra pequeño.

8.3.2. Aproximación de emsemble selección de características

En lugar de optar por un método particular, FS, y se aceptar su resultado como el subconjunto final, pueden ser combinados diferentes métodos de FS usando la aproximación emsemble FS. Basado en que a menudo no existe una única técnica universalmente óptima de la selección de características, y debido a la posible existencia de más de un subconjunto de de las características que discrimina a los datos igual de bien, el modelo de combinación de enfoques ha sido adaptado para mejorar la robustez y estabilidad del resultado.

Las nuevas técnicas emsemble en el microarray incluyen un promedio de varios subconjuntos de características destacadas, integrando una colección de expresiones genéticas diferenciales univariantes, utilizando diferentes iteraciones de un algoritmo genético para evaluar la importancia relativa a cada característica, calculando el test de Kolmogorov-Smirnov en diferentes muestras bootstrap para calcular una probabilidad debe ser seleccionado, y un número de aproximaciones Bayesianas de la media. Además, los métodos basados en una colección de árboles de decisión pueden ser utilizado como una emsemble FS para evaluar la relevancia de cada característica.

Aunque el uso de aproximaciones emsemble requiera adicionales recursos computacionales, nos gustaría señalar que ofrecen un marco oportuno para hacer frente a los dominios de muestra pequeña ,proporcionando recursos adicionales de computación que son asequibles.

8.4. SELECCIÓN DE CARACTERÍSTICAS EN LAS PRÓXIMAS DOMINIOS

8.4.1. polimorfismo de nucleótido único análisis

Polimorfismos de nucleótido único (SNP) son mutaciones en una única posición de nucleótidos que se produjeron durante la evolución y se aprobaron a través de la herencia, que representan la mayoría de la variación genética entre los diferentes individuos. SNP están en el primer plano de muchos estudios sobre asociaciones entre enfermedad-gen, cuyo número se estima sobre de 7 millones en el genoma humano. Por lo tanto, la selección de un subconjunto de SNP que sea lo suficientemente informativo, pero lo suficientemente pequeño para reducir la sobrecarga del genotipo, es un paso importante hacia la la asociación enfermedad-gen. Normalmente, el número de SNP considerado no es superior a decenas de miles, con muestras de tamaño cien.

Varios métodos computacionales para la selección htSNP(holotipo SNP; un conjunto de SNPs se encuentra en un cromosoma) se han propuesto en los últimos años. Una aproximación se basa en la hipótesis de que el genoma humano puede ser visto como una serie de bloques discretos que comparten sólo un conjunto muy pequeño de los holotipos más comunes. Esta aproximación apunta a identificar un subconjunto de SNPs que permita distinguir todos los haplotipos más comunes, o al menos explicar un cierto porcentaje de ellos. Otra aproximación de la selección htSNP se basa en las asociaciones de pares de SNPs, y trata de seleccionar un conjunto de htSNPs de tal manera que cada uno de los SNPs de un haplotipo es altamente asociado con uno de los htSNPs . Una tercera aproximación considera a htSNPs como un subconjunto de todos los SNPs, a partir del cual pueden ser reconstruídos los restantes SNP. La idea es seleccionar htSNPs en base de lo bien que predigan el conjunto restante formado por los SNPs no seleccionados.

Una aproximación ensemble es aplicada con éxito a la la identificación de SNPs para el alcoholismo, mientras que proponen una sólida técnica de la selección de características basadas en un híbrido entre un algoritmo genético y la SVM. El algoritmo de selección de variables Relief-F, ha sido propuesto en relación con tres algoritmos de clasificación (K NN, SVM y naive Bayes). Algoritmos Genéticos han sido aplicados a la búsqueda del mejor subconjunto de SNPs, evaluandolas con un filtro de múltiples variables (CFS), y también de forma wrapper (con un árbol de decisión). La regresión lineal múltiple SNP (algoritmo de predicción) predice un genotipo completo basado en los valores de las SNPs informativas, sus posiciones entre todos los SNPs, y una muestra de genotipos completa.

8.5. CONCLUSIONES Y PERSPECTIVAS FUTURAS

Los principales problemas que aparecen en el campo de la bioinformática son: la gran dimensionalidad, y pequeños tamaños de muestra. Para hacer frente a estos problemas, una gran cantidad de técnicas de FS ha sido diseñado por los investigadores en bioinformática, el aprendizaje de máquinas y minería de datos.

Un esfuerzo amplio y fructífero se ha realizado durante los últimos años en la adaptación y la propuesta de técnicas FS de filtro univariante. En general, se observa que muchos investigadores en el campo todavía piensan que aproximaciones FS de filtro se limitan sólo a enfoques univariantes. La propuesta de algoritmos de selección múltivariantes puede ser considerada como una de las líneas de futuro prometedor de trabajo para la bioinformática de la comunidad.

Otra línea de investigación se basa en mejorar la robusted de la solución determinada por la aproximación ensemble. A fin de aliviarlos pequeños tamaños de muestra real de la mayoría de las aplicaciones en bioinformáticas, el desarrollo de estas técnicas, combinadas con los criterios de evaluación adecuados, constituye una interesante dirección para futuras investigaciones FS.

Otras oportunidades interesantes para la investigación futura FS será la extensión de las SNP hacia ámbitos como la bioinformática, y la combinación de heterogéneas fuentes de datos. Actualmente la selección de variables no son esenciales en este campo aunque se cree que su aplicación lo será para hacer frente a la alta dimensional de estas aplicaciones.

ANEXO I: PAQUETE 'WilcoxCV'

INTRODUCCIÓN

Hace pocos años, numerosos métodos están basados en microarrays para predecir clases. Aunque muchos de ellos han sido especialmente en el caso de que n<<p (muchas más variables que datos), anteriormente la selección de variables era casi siempre necesaria cuando el número de genes alcanza decenas de miles, es usual es recientes conjuntos de datos. El estadístico de la suma de los rangos de Wilcoxon es, junto con el t-estadístico, una de las estándar aproximaciones para la selección de variables. Es bien conocido que el paso de la selección de variables debe ser visto como una parte de la construcción del clasificador y , el cual, debe ser obtenido basándonos solamente en los datos de entrenamiento.

Cuando la exactitud del clasificador es evaluada vía validación cruzada o validación cruzada con Monte-Carlo, esto significa que tenemos que realizar p Wilcoxon o t-test para cada iteración, la cual comienza a ser una desalentadora tarea debido al incremento de p.

Como consecuencia de ello, muchos autores a menudo realizan la selección de variables usando sólo una vez con todos los datos disponibles, que pueden inducir una dramática subestimación de la tasa de error y así producir informes donde se pierda poder predictivo .Se propuso un método rápido de selección de variables basado en el test de Wilcoxon basado en la validación cruzada y de Monte Carlo de validación cruzada (también lo conocemos como división de azar en el aprendizaje y equipos de prueba). Esta implementación se basa en una simple fórmula matemática utilizando sólo el rango calculado del conjunto de datos original. ("new")

Test de la suma de los rangos de Wilcoxon

La idea de usar un test basado en rangos surge de aplicar la hipótesis de simetría a los test de signos.

La teoría de los tests basados en rangos es más complicada que la del test del signo. Bajo Ho, el estadístico de un test de rangos puede ser representado como una suma de v.a. independientes pero no idénticamente distribuidas y, bajo H1, se pierde inclusive la independencia. Por ello, necesitaremos nuevas versiones del teorema central del límite.
Se desea realizar el siguiente test

$$H_0: \theta = 0 \qquad \text{vs} \qquad H_1: \theta > 0$$

siendo θ el centro de simetría de X (será además la media si ésta existe).
Supondremos que X1,...,Xn es una muestra aleatoria de una distribución F(x-θ) con Fє Ωs, siendo

Ωs = {F / F es absolutamente continua con única mediana en 0 y simétrica}

El test del signo se basa en información sobre el signo de las observaciones y no utiliza información sobre la distancia de las observaciones al cero. Sin embargo, si la distribución es simétrica alrededor de 0, el vector de valores absolutos $|X_1|, |X_2|,...,|X_n|$ es un estadístico suficiente y por lo tanto, parece razonable tratar de incorporar esta información.

Sea $|X|^{(1)} \leq |X|^{(2)} .... \leq |X|^{(n)}$, la muestra de valores absolutos ordenados y

R_j = rango ($|X_j|$) es decir $|X_j| = |X|^{(Rj)}$

D_j = j-ésimo antirango es decir $|X_{Dj}| = |X|^{(j)}$

Estadístico del test: el estadístico del test de Wilcoxon (1945), T+, es la suma de los rangos de los valores absolutos de las observaciones mayores que 0 en la muestra original. Es decir, si definimos

$$W_j = \begin{cases} 1 & \text{si } |X|^{(j)} \text{ corresponde a una observación mayor que 0} \\ 0 & \text{en caso contrario} \end{cases}$$

$$T^+ = \sum_{j=1}^{n} j W_j$$

Pero $W_j = s(X_{D_j})$, con $s(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$.

entonces, podemos expresar al estadístico en la forma

$$T^+ = \sum_{j=1}^{n} R_j \, s(X_j)$$

Observación: Si θ > 0 y la distribución simétrica se halla desplazada hacia la derecha, las observaciones positivas tienden a estar más alejadas del 0 que las negativas, entonces T+ tiende a ser grande y se rechazaría Ho.

PAQUETE 'WilcoxCV'

Este paquete proporciona funciones que actúan rápido en la selección de variables, basadas en el test suma de los rangos de Wilcoxon en validación cruzada o validación cruzada mediante Monte-Carlo, para usar microarray basado en clasificación binaria.

generate.cv : Genera grupos mediante validación cruzada

Genera aleatoriamente m grupos para realizar de validación cruzada m veces:

- **Uso**

generate.cv(n,m)

- **Argumentos**

n el número total de observaciones en el conjunto de datos

m el número deseado de grupos

- **Importante**

Una matriz de dimensión m x (n/m) da el número máximo de índices de las observaciones incluidas en cada grupo.

La i-esima fila da los índices de observaciones incluidas en el grupo i-ésimo. Si los m grupos no son exactamente igual tamaño, la última columna incluye uno o varios de ceros.

EJEMPLO:

```
library(WilcoxCV)

#genera 10 grupos para un conjunto de datos de tamaño 95.

generate.cv(n=95,m=10)
[1] 10
[1] 10
[1] 10
[1] 10
[1] 10
[1] 9
[1] 9
[1] 9
[1] 9
[1] 9
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	4	5	7	20	25	26	39	42	53	93
[2,]	1	27	40	55	57	61	65	68	80	92
[3,]	15	16	35	44	74	76	84	85	89	91
[4,]	2	9	10	11	33	45	47	60	82	94
[5,]	8	24	28	37	46	56	59	70	73	87
[6,]	17	49	50	58	64	66	71	75	77	0
[7,]	14	18	21	23	34	38	43	51	83	0
[8,]	3	6	19	36	67	72	78	79	86	0
[9,]	12	13	29	31	41	48	52	63	95	0
[10,]	22	30	32	54	62	69	81	88	90	0

generate.split : Genera divisiones aleatorias en el aprendizaje y en conjuntos de datos de prueba.

La función generate.split genera *niter* divisiones aleatorias en el aprendizaje y en conjuntos de datos de prueba para su uso en Monte-Carlo de validación cruzada (MCCV).

- **Uso**

generate.split (niter, n, ntest)

- **Atributos**

niter El número de iteraciones (número de partes en el aprendizaje y partes de los conjuntos).

n El número total de observaciones en el conjunto de datos.

ntest El número de observaciones en los conjuntos de prueba.

- **Detalles**

Esta función está pensada para su uso en Monte-Carlo de validación cruzada (MCCV).

- **Importante**

Una matriz de dimensión niter x ntest da los índices de las observaciones incluidas en los conjuntos de prueba. La i-ésima fila da los índices de las ntest observaciones incluidas en el conjunto de prueba para la i-ésima iteración MCCV.

EJEMPLO:

```
Library(WilcoxCV)

#Genera 50 divisiones con relación 2:1 para el conjunto de datos incluyendo 90 observaciones

generate.split(niter=50, n=90, ntest=30)
```

wilcox.selection.split : Wilcoxon-based selecciona variables mediante validación cruzada (CV) y mediante validación cruzada de Monte-Carlo (MCCV).

La función wilcox.selection.split ordena las variables mediante el test Wilcoxon de suma de rangos para todas las iteraciones CV o MCCV.

- **Uso**

wilcox.selection.split (x,y, split, algo=”new”, pvalue=FALSE)

- **Atributos**

x una matriz o un data frame de tamaño n x p da la expresión de los niveles de las p variables (genes) para las n observaciones (arrays). Variables correspondientes a columnas, observaciones correspondientes a filas.

y un vector de longitud n da la clase del número de miembros para las n observaciones(arrays). **y** puede ser numérico o un factor pero debe ser codificado como 0,1.

split una matriz *niter x nest* da los índices de las *ntest* observaciones incluidas en cada uno de de las *niter* de los conjuntos de prueba, como los generados por las funciones anteriormente explicadas. La fila i-ésima de Split da los índices de las observaciones incluidas en el conjunto de datos de prueba para la i-ésima división iterada aleatoriamente.

algo “new” o “naive”. Si type=”new”, nuevo método. Si type=”naive”, los resultados son obtenidos tras recorrer la función Wilcox.test *niter* veces.

- **Detalles**

El estadístico suma de los rangos de Wilcoxon es definido como la suma del rango de X-rangos de observaciones con y=0. El test de la suma del rango Wilcoxon es equivalente al test Mann-Whitney. Está implementado en la función wilcox.test.

En el contexto de CV o MCCV, wilcox.selection.split calcula el estadístico de la suma de los rangos Wilcoxon para cada iteración y para cada variable. En cada iteración, un sujeto de las n observaciones es excluido del conjunto de datos y este será considerado como el conjunto de datos de prueba. Los índices de la observación considerada como el conjunto de prueba para cada para cada iteración está dando el *split* en la matriz de dimensión niter x ntest.

- **Importante**

Ordering Split.

Una matriz niter x p da los índices de los genes ordenados por el p-valor. Por ejemplo, la primera columna de *ordering.split* da los índices de las variables con el pvalor más bajo en cada una de las iterativas divisiones aleatorias, la segunda columna de *ordering.split* da los índices de las variables con el segundo p-valor más bajo en cada una de las iterativas divisiones aleatorias. Para la i-esima iteración , el índice de las 50 mejores variables están en las 50 primeras columnas de la fila i.

pvalue.split

Devuelve solamente si pvalue=TRUE. Una matriz niter x p de valores. El elemento de la i-esima fila y la j-esima columna es el p-valor de la variable j en la iteración i-esima.

EJEMPLO:

```
#Generamos un conjunto de datos
x<-matrix(rnorm(1000),100,10)
y<-sample(c(0,1), 100, replace=T)

#Generamos 50 divisiones MCCV con proporción 2:1 para el conjunto de datos incluyendo 90 observaciones
div<-generate.split(niter=50, n=90, ntest=30)
#calculamos la suma del rango del estadístico Wilcoxon para las 50 interacciones
solucion<-wilcox.selection.split(x=x,y=y, split=div, algo="new", pvalue=T)
attach(solucion)
ordering.split
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	6	9	2	3	4	8	10	1	7	5
[2,]	6	3	2	10	9	4	7	5	8	1
[3,]	6	9	8	2	7	10	3	5	1	4
[4,]	1	6	2	9	7	8	4	10	3	5
[5,]	6	3	8	1	9	10	7	4	2	5
[6,]	6	3	2	9	7	4	8	5	1	10
[7,]	6	2	1	9	3	4	7	8	5	10
[8,]	2	9	10	3	8	1	6	4	7	5
[9,]	6	2	9	8	10	3	7	1	4	5
[10,]	9	3	7	6	8	10	4	1	5	2
[11,]	9	3	1	7	10	5	4	2	8	6
[12,]	9	6	8	2	1	10	5	4	3	7
[13,]	4	6	9	10	2	3	8	7	5	1
[14,]	9	2	6	3	5	8	7	4	1	10
[15,]	9	6	8	2	3	1	5	7	10	4
[16,]	6	9	7	2	8	4	10	3	5	1
[17,]	9	7	5	6	3	1	8	4	2	10
[18,]	2	10	6	7	8	3	1	5	9	4
[19,]	6	2	3	9	7	5	10	8	1	4
[20,]	2	6	9	8	7	4	1	10	5	3
[21,]	6	2	5	10	9	4	7	3	8	1
[22,]	9	6	2	8	10	4	3	7	5	1
[23,]	6	2	9	7	4	10	8	1	5	3
[24,]	6	9	2	3	8	5	10	4	1	7
[25,]	8	3	2	6	10	5	9	4	1	7
[26,]	6	9	7	2	8	3	10	4	5	1
[27,]	6	3	9	7	2	1	10	4	8	5
[28,]	6	9	2	8	1	4	3	7	5	10
[29,]	6	3	2	8	9	1	10	4	5	7
[30,]	9	10	5	6	7	8	2	1	3	4
[31,]	8	2	7	6	1	9	3	10	5	4
[32,]	6	2	9	7	5	3	4	10	1	8
[33,]	7	6	8	9	2	3	4	10	5	1
[34,]	8	2	9	3	6	10	1	4	5	7
[35,]	6	7	2	3	1	9	5	10	8	4
[36,]	6	7	4	2	9	10	5	8	3	1
[37,]	9	8	2	4	6	7	1	5	10	3
[38,]	6	9	3	8	2	7	5	1	4	10
[39,]	7	9	6	2	3	8	4	1	10	5
[40,]	6	9	2	10	4	5	3	1	8	7
[41,]	9	6	2	1	3	8	5	4	10	7
[42,]	9	2	6	8	10	7	4	1	3	5
[43,]	3	9	2	8	10	6	4	5	7	1
[44,]	2	9	8	7	6	3	10	4	5	1
[45,]	2	8	3	9	6	10	4	7	1	5
[46,]	2	6	9	1	5	3	4	7	10	8
[47,]	6	9	2	7	1	3	8	10	5	4
[48,]	6	3	9	1	2	5	8	7	10	4
[49,]	3	9	7	6	8	4	10	2	5	1
[50,]	7	2	8	3	9	1	10	5	6	4

pvalue.split


```
> pvalue.split
             [,1]        [,2]       [,3]       [,4]      [,5]        [,6]
 [1,] 0.72505101 0.226224164 0.38737500 0.40731956 0.8533858 0.065469331
 [2,] 0.97622140 0.150811270 0.11145990 0.25486706 0.5793049 0.067232416
 [3,] 0.67987639 0.294056746 0.61219155 0.87819069 0.6121915 0.029179433
 [4,] 0.05037106 0.157680558 0.86387652 0.77206317 0.9575678 0.154232630
 [5,] 0.15422971 0.663435809 0.04207261 0.47071478 0.8627477 0.040882099
 [6,] 0.66058240 0.122875388 0.04617466 0.60154227 0.6265557 0.009676390
 [7,] 0.20413580 0.122875388 0.36081045 0.39944381 0.7307209 0.036731520
 [8,] 0.48782880 0.004664122 0.31370931 0.73094649 0.9855605 0.503090323
 [9,] 0.68086640 0.068547919 0.51050613 0.85086550 0.9251035 0.049716103
[10,] 0.80318740 0.971598558 0.10147204 0.65200610 0.9243605 0.264604202
[11,] 0.42425534 0.680866405 0.29024694 0.62163295 0.6051288 0.887861511
[12,] 0.63499179 0.545007154 0.84010992 0.82159926 0.7576538 0.226080460
[13,] 0.98591968 0.246503602 0.38074954 0.07859198 0.9484055 0.124692033
[14,] 0.81419493 0.097544667 0.39751702 0.66372001 0.4807770 0.172849997
[15,] 0.46627383 0.295639916 0.34718106 0.72444331 0.4662738 0.200247430
[16,] 0.97187827 0.115344372 0.80508725 0.41747400 0.8508655 0.006638483
[17,] 0.45500465 0.791225460 0.40684709 0.66760738 0.2188935 0.343583757
[18,] 0.46213715 0.124692033 0.42028657 0.81853964 0.7197132 0.276466318
[19,] 0.84939920 0.120021354 0.13794770 0.98106348 0.4265299 0.031707814
[20,] 0.69532575 0.075047737 0.90553034 0.46910453 0.8587094 0.099019351
[21,] 0.87819069 0.045047393 0.75024527 0.40256272 0.1751659 0.016170077
[22,] 0.87819069 0.443488170 0.73242997 0.69723705 0.8227572 0.370252495
[23,] 0.91577436 0.092883951 0.97187827 0.48811975 0.9251035 0.084099306
[24,] 0.94379145 0.226147837 0.28492057 0.86008706 0.7421369 0.020617025
[25,] 0.75024527 0.178948143 0.13741817 0.70597789 0.3278138 0.262713494
[26,] 0.92034020 0.159737390 0.23240771 0.83687446 0.8460759 0.046127891
[27,] 0.50411368 0.347231357 0.16118657 0.71838457 0.8360783 0.079122767
[28,] 0.36398919 0.294056746 0.53984131 0.45764077 0.8411512 0.046324757
[29,] 0.38149432 0.053592327 0.03698179 0.84788878 0.8572903 0.030015693
[30,] 0.48811975 0.459114320 0.51808579 0.65521170 0.2086287 0.212913384
[31,] 0.43398332 0.349611734 0.69347908 0.86454059 0.7552070 0.387175599
[32,] 0.70956366 0.032830277 0.41122632 0.58245958 0.3352364 0.013693202
[33,] 0.93444296 0.274467254 0.54116896 0.75996776 0.8508655 0.079966148
[34,] 0.75103569 0.239959134 0.58074845 0.81419493 0.9344430 0.588824053
[35,] 0.36809539 0.119018998 0.30326113 0.94840550 0.7109299 0.038940692
[36,] 0.95777659 0.361867675 0.94840550 0.29776858 0.6421248 0.031779729
[37,] 0.56499467 0.175520018 0.75527162 0.23059386 0.5813170 0.264887973
[38,] 0.69238976 0.548896006 0.19541099 0.80123090 0.6747845 0.028245488
[39,] 0.78695219 0.397517021 0.48811975 0.62163295 0.9437915 0.138709978
```

wilcox.split : El estadístico de la suma del rango Wilcoxon en validación cruzada (CV) y validación cruzada de Monte-Carlo (MCCV).

La función wilcox.split calcula el estadístico de la suma de los rangos de Wilcoxon para todas las iteraciones CV o MCCV definidas por la matriz *split*

- **Uso**

wilcox.split (x,y, split, algo=”new”)

- **Atributos**

x una matriz o un data frame de tamaño n x p da la expresión de los niveles de las p variables (genes) para las n observaciones (arrays). Variables correspondientes a columnas, observaciones correspondientes a filas.

y un vector de longitud n da la clase del número de miembros para las n observaciones(arrays). **y** puede ser numérico o un factor pero debe ser codificado como 0,1.

split una matriz niter x nest da los índices de las *ntest* observaciones incluidas en cada uno de de las niter de los conjuntos de prueba, como los generados por las funciones anteriormente explicadas. La fila i-ésima de Split da los índices de las obsevaciones incluidas en el conjunto de datos de prueba para la i-ésima división iterada aleatoriamente.

algo “new” o “naive”. Si type=”new”, nuevo método. Si type=”naive”, los resultados son obtenidos tras recorrer la función Wilcox.test *niter* veces.

- **Detalles**

El estadístico de la suma de los rangos de Wilcoxon es definido como la suma del rango de X-rangos de observaciones con y=0. El test de la suma del rango Wilcoxon es equivalente al test Mann-Whitney. Está implementado en la función wilcox.test.

En el contexto de CV o MCCV, wilcox.selection.split calcula el estadístico de la suma de los rango Wilcoxon para cada iteración y para cada variable. En cada iteración, un sujeto de las n observaciones es excluido del conjunto de datos y este será considerado como el conjunto de datos de prueba. Los índices de la observación considerada como el conjunto de prueba para cada para cada iteración está dando el *split* en la matriz de dimensión niter x ntest.

- **Importante**

Wilcox. Split.

Un numérico vector de longitud niter el cual su i-esima componente da el estadístico de la suma del rangp Wilcoxon obtenido en la i-ésima interacción.

EJEMPLO:

```
#Generamos un conjunto de datos
x<-rnorm(100)
y<-sample(c(0,1), 100, replace=T)

#Generamos 50 divisiones MCCV con proporción 2:1 para el conjunto de datos incluyendo 90 observaciones
div<-generate.split(niter=50, n=90, ntest=30)

#calculamos la suma del rango del estadístico Wilcoxon para las 50 interacciones
wilcox.split(x=x,y=y, split=div, algo="new")
```

```
[1] 1170 1232 1149 1146 1124 1188 1203 1103 1036 1146 1142 1227 1091 1134 1100
[16] 1167 1205 1229 1339 1167  984 1169 1188 1089 1013 1265 1141 1191 1053 1334
[31] 1197 1355 1239 1214 1249 1035 1082 1073 1080 1050 1159 1121 1309 1113 1183
[46] 1152 1116 1085  952 1223
```