Un análisis crítico del clasificador AkDE como *ensemble* y sus implicaciones para tratar con grandes volúmenes de datos

Jacinto Arias
Dpto. de Sistemas Informáticos
Universidad de Castilla-La Mancha
Albacete, 02071, España
jacinto.arias@uclm.es

José A. Gámez
Dpto. de Sistemas Informáticos
Universidad de Castilla-La Mancha
Albacete, 02071, España
jose.gamez@uclm.es

José M. Puerta
Dpto. de Sistemas Informáticos
Universidad de Castilla-La Mancha
Albacete, 02071, España
jose.puerta@uclm.es

***Abstract*—En clasificación supervisada el tamaño muestral condiciona enormemente el modelo a utilizar, especialmente para volúmenes de datos masivos donde la eficiencia del modelo y su potencia predictiva constituyen un equilibrio entre rendimiento y complejidad computacional. Los clasificadores basados en Redes Bayesianas permiten ajustarlo parametrizando su aprendizaje para estimar distribuciones de probabilidad cada vez más complejas. El rendimiento puede descomponerse en sesgo, que se reduce al aumentar la complejidad, y varianza, que aumenta de manera inversamente proporcional. El clasificador AkDE es uno de los ejemplos más estudiados ya que puede aprenderse en una única pasada sobre los datos y al tratarse de un modelo de tipo ensemble reduce la varianza agregando las predicciones de clasificadores individuales. En la práctica es necesario reducir su complejidad espacial y es utilizado junto a técnicas de selección de modelos basadas en la Teoría de la Información lo que implica pasadas adicionales sobre los datos. Este trabajo estudia el rendimiento de este clasificador comparándolo a otras técnicas de ensemble populares y cuestiona el impacto real de la agregación en la reducción del sesgo y la varianza. Comprobaremos empíricamente como en problemas con muestras grandes los resultados no se ajustan al modelo teórico y cómo la selección de modelos difiere del comportamiento básico de un ensemble. Los resultados obtenidos se utilizarán para proponer un modelo alternativo que sí capture las propiedades deseadas para un ensemble.**

***Index Terms*—Clasificación supervisada; Clasificadores basados en redes Bayesianas; Clasificadores ensemble.**

I. Introducción

La actual abundancia de datos y potencia computacional motiva a los investigadores en aprendizaje automático a desarrollar nuevos métodos buscando el balance entre escalabilidad y precisión. Sin embargo, no deberíamos caer en el error de conseguir este balance a cambio de producir algoritmos complejos de usar y que generen modelos difíciles de interpretar, o de otra manera, no tendrán aceptación por parte de la industria.

De hecho, gran parte de los algoritmos incluidos en los paquetes software que son referencia en la actualidad (e.g. [12]), fueron propuestos hace más de una década, pero todavía son competitivos. Parte de su éxito, sin duda se debe a sus fundamentos teóricos, pero también a su funcionamiento intuitivo y a la facilidad de interpretar y fijar los hiperparámetros necesarios para su uso. Un ejemplo claro son los clasificadores tipo *ensembles* [9] y, en particular, Random Forest (RF) [5], bien considerados tanto por investigadores como por usuarios. Desde el punto de vista de su usabilidad, el éxito recae en que un único parámetro, el número de modelos en el ensemble, sirve para controlar tanto la complejidad del modelo resultante como su nivel de precisión. P.e. en RF está demostrado que incrementar el número de árboles reduce la varianza en la clasificación manteniendo un sesgo estable. Esto garantiza que el rendimiento del clasificador mejorará o se estabilizará con un mayor número de árboles, lo que permite al usuario fijarlo basándose únicamente en la complejidad del problema abordado y/o en los recursos disponibles (tiempo y espacio).

En este trabajo nos centramos en los clasificadores basados en redes Bayesianas (BNCs), los cuáles al poderse aprender *out of core* son, en principio, excelentes candidatos para abordar grandes volúmenes de datos [1]. Además, en algunos de estos algoritmos, un único parámetro, k, controla la complejidad de los modelos aprendidos. Por ejemplo en el algoritmo kDB [14], incrementar el valor de k permite aumentar el número de dependencias permitidas y, por tanto, la complejidad de la red resultante. De forma similar, en el clasificador tipo ensemble AkDE [18], aumentar el valor de k implica también aumentar el número de dependencias, pero también el número de modelos, por lo que el orden de incremento en la complejidad es incluso superior al caso de kDB. La implicación, sin embargo, en términos del análisis en términos de sesgo y varianza es diferente: en kDB aumentar k permite reducir el sesgo a cambio de incrementar la varianza, mientras que en AkDB un aumento en k reduce la varianza pero aumenta mucho la complejidad del modelo resultante. Además, en AkDE la influencia de incrementar k se traslada en un aumento de las necesidades computacionales, sobre todo espaciales, que hacen obligatorio realizar una selección de modelos [6], [7], [11] para permitir su uso con bases de datos

Este artículo ha sido parcialmente financiado por fondos FEDER y la Agencia Estatal de Investigación (AEI/MINECO) mediante los proyectos TIN2016-77902-C3-1-P y TIN2016-82013-REDT. Jacinto Arias también está financiado por el MECD mediante la beca FPU13/00202.

de tamaño medio-grande. En este trabajo también estudiamos de forma crítica algunos de los principios en los que se basan estos procesos de selección de modelos.

Aunque el análisis basado en sesgo y varianza es habitual en otro tipo de ensembles, no es común en los ensembles de BNCs, por ello, la novedad de este trabajo reside en que proponemos evaluar los ensembles (y sus modelos constituyentes) de BNCs usando el enfoque de sesgo y varianza, estudiando su comportamiento en dominios de diferente tamaño. Los resultados que obtenemos indican que el algoritmo AkDE exhibe un comportamiento completamente diferente en muestras grandes que al considerar los benchmark habituales en aprendizaje automático, tradicionalmente formados por conjuntos de pequeño tamaño. Por otra parte, el estudio de los modelos constituyentes del ensemble, nos muestra discrepancias importantes con respecto al funcionamiento interno (y agregado) de otros ensembles típicos como RF. Estas observaciones podrían hacer a los usuarios replantearse la forma que se entiende el clasificador AkDE, especialmente en el caso de grandes conjuntos de datos. A partir del estudio, proponemos un nuevo ensemble de BNCs que si posee las propiedades esperadas en clasificadores clásicos tipo ensemble como bagging y RF.

El resto del trabajo contiene una breve descripción de los BNCs usados en el estudio, seguido de la descripción del análisis de sesgo y varianza seguido en el artículo y su aplicación al estudio del algoritmo AkDE desde la perspectiva *ensemble*. A continuación realizamos una experimentación usando tanto datos masivos como benchmarks clásicos en aprendizaje automático para evaluar los clasificadores objeto del estudio. Por último, a la luz de los resultados, proponemos un ensemble de BNCs cuyo comportamiento se asemeja más a los ensembles clásicos (bagging y RF).

II. Clasificadores basados en Redes Bayesianas

El problema de clasificación supervisada consiste en predecir la *etiqueta* $y \in \Omega_Y = \{y_1, \ldots, y_c\}$ para la variable de respuesta (o clase) Y, para un ejemplo $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_d})$ definido sobre d atributos $\{X_1, \ldots, X_d\}$. Para resolverlo, el objetivo es inducir un modelo (o clasificador) a partir de un conjunto de datos formado por m ejemplos previamente *etiquetados* $D = \{(x^{(1)}, y^1), \ldots, (x^{(m)}, y^{(m)})\}$.

El formalismo de las Redes Bayesianas (RBs) [13] usa un grafo dirigido acíclico (DAG) para representar de forma eficiente la distribución de probabilidad conjunta (DPC) definida sobre el conjunto de variables. En particular una RB factoriza la DPC mediante el producto $p(y, \mathbf{x}) = \prod_{\mathbf{i=1}}^{\mathbf{d}} \mathbf{p}(\mathbf{x_i} \mid \pi_{\mathbf{x_i}})$, donde π_{x_i} denota el conjunto de padres del atributo X_i en el DAG. Desde un punto de vista probabilístico, la etiqueta a asignar a un ejemplo $\mathbf{x}$ es la que maximiza la probabilidad a posteriori, es decir, $\arg\max_{y \in \Omega_Y} p(y \mid \mathbf{x})$.

En la práctica, los algoritmos que han demostrado un mejor rendimiento en esta tarea son los denominados BNCs [3], redes Bayesianas cuya estructura otorga un papel distinguido a la variable clase y que limitan de distinta forma el número de dependencias permitido. De ellos, el más popular por su simplicidad es Naive Bayes (NB), que asume que todas las variables predictoras son independientes dada la clase. Sin embargo, la suposición realizada resta capacidad discriminativa a NB lo que lo convierte en un clasificador con mucho sesgo. Otros modelos de BNCs se basan en imponer la estructura base de NB pero permitiendo algunas dependencias adicionales entre las variables predictoras.

A. k-dependence BN Classifier (kDB)

El algoritmo kDB [14] se basa en el uso de estimadores k-dependientes, es decir, cada variable puede depender de otros k atributos y de la clase. Requiere aprendizaje estructural y el modelo producido es una única RB.

El algoritmo propuesto en [14] usa los conceptos de información mutua (MI) e información mutua condicional (CMI) para guiar el proceso de aprendizaje estructural. Primero se ordenan las variables en función de $I(X_i; C)$. Después, a la variable i-ésima de dicho orden se le ponen como padres en el grafo la variable clase C y las k variables X_j con mayor $I(X_j, X_i|C)$, t.q. sólo se consideran las que preceden a X_i en el orden. La posterior estimación de parámetros requiere una pasada adicional por los datos cuando $k \geq 2$.

Fijando adecuadamente el valor de k se mejora el ajuste del modelo a los datos, sin embargo, aumentar k implica también obtener modelos más complejos, habitualmente con menor sesgo pero mayor varianza debido al sobreajuste. En [11] se estudia que en bases de datos grandes el riesgo de sobreajuste (e incremento de la varianza) disminuye considerablemente. El algoritmo kDB *selectivo* (SkDB) combate la complejidad del modelo eligiendo el k adecuado para cada variable [11], lo que requiere una pasada adicional por los datos.

B. Averaged k Dependence Estimators

El clasificador AkDE puede modelarse como un ensemble formado por un conjunto de BNCs $H = \{h_i(\pi_i), i = 1, \ldots, K\}$, donde el número de modelos K se fija restringiendo el espacio de los posibles modelos a una familia de clasificadores k-dependientes. En particular, cada modelo es un NB aumentado en el que además de depender de la clase, todos los atributos dependen de un conjunto fijo (el mismo para todos) de k padres, denotado por π_i. El ensemble consiste entonces en la familia *completa* de este tipo de clasificadores, por lo que $K = \binom{d}{k}$. La etapa de inferencia consiste en promediar la distribución de probabilidad de la clase a posteriori en todos los clasificadores del ensemble. Como caso particular, el clasificador A1DE (AODE) asume que todos los atributos dependen de la clase y otro atributo, conocido como *super padre* (SP). En AODE los modelos constituyentes se conocen como SPODE y en general como SPkDE. Al fijarse la estructura, no se requiere aprendizaje estructural, siendo un clasificador *out of core* real.

Al igual que en kDB, en AkDE podemos ajustar el balance sesgo-varianza variando el parámetro k, lo que permite representar desde clasificadores con mucho sesgo y poca varianza como NB (A0DE), hasta clasificadores con poco sesgo pero alta varianza, obtenidos al incrementar k y poder usar distribuciones de probabilidad de alta dimensionalidad

[18]. Sin embargo, en el caso de AkDE la influencia de incrementar k en la complejidad del modelo resultante se traslada al incremento del tamaño de las tablas de probabilidad y, sobre todo, al incremento en el número de modelos del ensemble. Por ejemplo, con 100 variables predictoras A1DE contendrá 100 modelos individuales pero A2DE contendrá 5000. Indudablemente esto representa un problema muy importante de cara a la escalabilidad del algoritmo en problemas de media-alta dimensionalidad (número de variables) por lo que se han propuesto diferentes soluciones basadas en realizar selección de modelos [6], [7], [11]. Es importante remarcar que si en kDB la selección de modelos tenía como objetivo un mejor ajuste de los datos, en AkDE la selección de modelos es obligatoria para convertirlo en un modelo usable en la práctica.

Estas técnicas de selección de modelos se basan en una aproximación híbrida entre el enfoque filter y wrapper, guiados por el uso de conceptos de Teoría de la Información, mayormente Información Mutua (MI). En particular, los algoritmos *Sample Attribute Selective AkDE (SASAkDE)* [6] y **Selective AkDE (SAkDE)** [7] han mostrado que es posible reducir la complejidad espacial, pero también (sorprendentemente) obtener mejor predicción (accuracy) que el ensemble formado por la familia completa de SPkDEs. Sin embargo, también hay algunos puntos débiles que deben ser estudiados: (1) este proceso añade pasadas adicionales por los datos, lo que supone un inconveniente en el caso de grandes muestras (e.g. big data); (2) la idea subyacente es asumir que la información mutua condicional del conjunto de super-padres dada la clase es un buen indicador del rendimiento en términos de clasificación del sub-modelo resultante. En este trabajo analizamos empíricamente esta hipótesis; y (3) volviendo a nuestra discusión inicial y las ventajas de tener hiperparámetros interpretables, indicar que este proceso de selección de modelos también añade cierta confusión al proceso, puesto que a igual k la complejidad del modelo resultante puede diferir mucho de un problema a otro.

III. Sesgo y Varianza en clasificación

Distintos estudios [2], [4] han analizado la capacidad predictiva de los clasificadores ensemble mediante la descomposición del error en términos de *sesgo* y *varianza*. Intuitivamente, un algoritmo de aprendizaje *sesgado* muestra un error persistente (similar) al entrenar con distintos conjuntos de datos, mientras que un algoritmo con alta varianza muestra un error que fluctúa entre los distintos conjuntos de datos.

Breiman [4] presenta una taxonomía de clasificadores estables (poca varianza a riesgo de tener mucho sesgo) e inestables (poco sesgo y alta varianza). Los clasificadores inestables, *en media* tienen un buen rendimiento, lo que les convierte en el marco ideal para los ensembles basados en voto por la mayoría, puesto que su uso consigue reducir la varianza [4]. Ejemplos son RF o bagging con árboles de decisión, que son algoritmos estado-del-arte en clasificación supervisada. Por otra parte, los modelos inestables son buenos candidatos para su aplicación en problemas grandes [11], ya que el impacto de la varianza se suaviza al disponer de más datos.

Sin embargo, la descomposición del error en sesgo y varianza proviene de la regresión numérica, y aunque su interpretación es también intuitiva en el caso de la clasificación, su aplicación no lo es. De hecho, mientras que en regresión promediar funciones independientes reduce la varianza sin modificar el sesgo, en clasificación promediar los modelos podría incrementar el error de clasificación [15].

Existen distintas formulaciones de la descomposición en sesgo/varianza [15], [16]. En este trabajo hemos optado por la implementación realizada en [16][1] de la descrita en [2], [4]. Las métricas utilizadas se obtienen a partir de la estabilidad de un clasificador $\mathcal{L}$ cuando se entrena y testea repetidamente sobre un número de conjuntos de datos $\mathcal{T}$. Definimos la tendencia central $C^{\circ}_{\mathcal{LT}}(\mathbf{x})$ como la clase con máxima probabilidad de ser seleccionada para un ejemplo determinado $\mathbf{x}$ por parte de todos los clasificadores aprendidos a partir de $\mathcal{T}$: $C^{\circ}_{\mathcal{LT}}(\mathbf{x}) = \arg\max_y P_{\mathcal{T}}(\mathcal{L}(\mathbf{x}) = y)$

El sesgo puede entonces medirse como el error introducido por la tendencia central del algoritmo, es decir, el error de la clasificación más frecuente, mientras que la varianza es el error introducido por la desviación de dicha tendencia central. Habitualmente se habla de estos valores en términos de *contribución* del sesgo y la varianza al error de clasificación, ya que éste puede expresarse como la suma de ambos. Para calcularlos, primero se obtiene una estimación de la tendencia central a partir de nuestra muestra $\mathcal{D}$, para lo que se ejecuta una 10x3 validación cruzada, induciendo así 30 modelos $\mathcal{L}(T^i_k)$ y el correspondiente conjunto de test f^i_k para cada una de las $i \in \{1, \ldots, 10\}$ repeticiones y $k \in \{1, 2, 3\}$ conjuntos (folds) de entrenamiento. Por tanto, se obtienen 10 predicciones independientes para cada instancia $\mathbf{x}$ y fijamos la tendencia central $C^{\circ}_{\mathcal{LT}}(\mathbf{x})$ para dicho ejemplo como la media. Formalmente, la tendencia central se obtiene como:

$$C^{\circ}_{\mathcal{LT}}(\mathbf{x}) = \arg\max_y P\left(\sum_{i=1}^{10}\sum_{k=1}^{3} 1\left[\mathbf{x} \in f^i_k \wedge \mathcal{L}(T^i_k)(\mathbf{x}) = y\right]\right)$$

La contribución del sesgo y la varianza al error se calculan para cada instancia y se agregan para todo el conjunto de datos:

$$\begin{aligned} sesgo &= P_{(\mathbf{x},y),\mathcal{T}}(\mathcal{L}(\mathcal{T})(\mathbf{x}) \neq y \wedge \mathcal{L}(\mathcal{T})(\mathbf{x}) = C^{\circ}_{\mathcal{LT}}(\mathbf{x}) \\ varianza &= P_{(\mathbf{x},y),\mathcal{T}}(\mathcal{L}(\mathcal{T})(\mathbf{x}) \neq y \wedge \mathcal{L}(\mathcal{T})(\mathbf{x}) \neq C^{\circ}_{\mathcal{LT}}(\mathbf{x}) \end{aligned}$$

IV. AkDE Bajo la Perspectiva de un Ensemble

Entre los modelos ensemble más populares encontramos el clasificador RF [5] basado en agregar el voto de árboles de decisión aplicando Bagging: Aprender cada modelo a partir de una muestra obtenida realizando bootstraping, y Random Subspaces: Seleccionar un conjunto de nodos subóptimo a la hora de estimar la partición de cada nodo del árbol. Esto incrementa la diversidad de los modelos de un modo predecible, especialmente en árboles completamente desarrollados. El estudio original demuestra que, si se incluyen modelos

[1]Esta metodología se ha usado p.e. para estudiar la propuesta inicial del clasificador AkDE [17], [18]

suficientes el sesgo del ensemble converge asintóticamente a niveles relativos a un árbol de decisión individual.

En el caso de AkDE se aprende un número finito de modelos en función de k, imponiendo una cota para la reducción de varianza y siendo imposible estabilizar el sesgo asintóticamente. Un número fijo de modelos reducirá al mismo tiempo la diversidad, deteriorando los resultados de la agregación. Además, los modelos del ensemble no se aprenden mediante una estrategia aleatoria subóptima sino que su estructura se fija heurísticamente conforme a cada super-padre. En general un clasificador k-dependiente H reduce el sesgo ajustando su estructura de la manera más fiel posible a un modelo óptimo sin restringir H', mientras que en el caso de AkDE no hay garantía de que las distribuciones de probabilidad modeladas aproximen a dicho clasificador, por lo que podemos asumir que muchos parámetros impactarán negativamente en el sesgo.

La literatura muestra que AkDE obtiene baja varianza en la práctica, sin embargo, no existe evidencia de que esto sea un resultado directo de la agregación o es inherente a los modelos individuales. Para profundizar realizaremos un experimento, para nuestro conocimiento inédito, evaluando la contribución particular al error de sesgo y varianza para una colección de ensembles y los modelos que los componen.

A. Sesgo y la Varianza en Problemas de Gran Tamaño

Un BNC con bajo sesgo es idóneo para grandes volúmenes de datos ya que obtendrá distribuciones de probabilidad mejor calibradas. Para determinar el sesgo de un modelo podemos medir su estabilidad para muestras de tamaño incremental. Para ello utilizaremos la base de datos sintética *pokerhand* [8] como benchmark, evaluando la descomposición en sesgo y varianza del error sobre 20 muestreos desde 50k instancias hasta 1M [2].

En la Figura 1 podemos ver que RF obtiene menor varianza que los modelos individuales, mientras que es máxima en el caso de árbol totalmente desarrollado. Al contrario, un árbol sin aleatorizar, es un modelo menos sesgado que el ensemble por estar compuesto por modelos que no se ajustan perfectamente a los datos. No obstante, la suma confirma que el ensemble es el de menor error y mayor estabilidad, especialmente para dominios pequeños donde la varianza es más dificil de reducir. En los modelos BNC confirmamos que los modelos con mayor sesgo como naive Bayes o kDB con $k = 1$ no mejoran conforme el tamaño de la muestra aumenta, mientras que al aumentar k la mejora es clara y constante, en presencia de suficientes datos que permitan calibrar los parámetros correctamente. Respecto a la varianza, confirmamos que los modelos sencillos son más estables, mientras que los clasificadores complejos requieren de más datos para estabilizarse.

Aunque los resultados para A1DE deberían ser superiores a los de kDB con $k = 1$ o $k = 2$, vemos que empeoran al aumentar la muestra. Si utilizamos el voto por la mayoría en lugar de la agregación numérica de probabilidades, Figura 1 bajo el nombre *a1de-majority*, vemos que la agregación de probabilidades suaviza el error cometido por los modelos de peor calidad, mientras que el voto por la mayoría los imita, implicando que si hay más modelos sesgados que acertados el ensemble lo estará también. Un tamaño de muestra mayor calibra las probabilidades de dichos modelos hacia valores extremos, convirtiendo al ensemble en un modelo equivalente al voto por la mayoría para muestras grandes.

B. Diseño Alternativo de Ensembles Basados en BNCs

Recientemente se han propuesto ensembles de BNCs alternativos, dado que en la práctica AkDE requiere aplicar selección de modelos y con ello una fase de aprendizaje adicional, podemos utilizar otros modelos que ya la realizan como kDB. Cabe destacar el clasificador kDF (k-dependent forest) [10] basado en construir un ensemble compuesto por n modelos kDB, uno para cada atributo X donde se aplica un proceso de ordenación de los atributos más sofisticado que introduce diversidad. Aunque esta propuesta supera en rendimiento a A1DE sufre de los mismos problemas descritos anteriormente ya que solo considera un número finito de modelos obtenidos de un modo no aleatorio.

Un modelo ensemble basado en agregación o voto por mayoría debería capturar ambas propiedades, para lo que proponemos una alternativa básica, el clasificador k-dependiente aleatorio (RkDB). Este ensemble tendá $h \in [1, \inf)$ modelos independientes aprendidos por una versión alterada de kDB, considerando solo un subconjunto en una proporción $\alpha \in [0, 1]$ de los padres disponibles para cada nodo en cada modelo, tomando como inspiración el particionado subóptimo que realiza RF. Así, añadimos diversidad mediante aleatoriedad controlada, preservando el sesgo del clasificador original.

C. Sesgo y Varianza en los Modelos Individuales

En nuestro segundo experimento evaluaremos el comportamiento de la descomposición del error para un benchmark clásico (véase la Tabla I) obtenido a partir del repositorio UCI [8]. Este conjunto de problemas es el utilizado en la mayoría de propuestas basadas en el clasificador AkDE. Estos modelos han obtenido siempre un buen rendimiento en este contexto, como hemos podido reproducir según muestran los resultados de la Tabla II. Comprobamos que no existe diferencia significativa entre el rendimiento de A1DE y RF y la nueva propuesta RkDB cuando k=1, no obstante para sucesivos valores de k los resultados empeoran, lo que achacamos al reducido tamaño de muestra de los problemas que conforman el benchmark. Si se observa la Figura 1 podemos comprobar que ocurre lo mismo en muestras de poco tamaño.

A la vista de que los ensembles mejoran a los modelos individuales, estudiaremos el efecto de la agregación en la descomposición en sesgo y varianza. La Figura 2 muestra la distancia entre el ensemble y los modelos individuales, podemos ver como RF reduce la varianza como esperabamos seguido de RkDB, mientras que en A1DE esta reducción es

[2]Experimentos realizados en un cluster Apache Spark de 7 nodos con procesadores hexacore Intel Xeon E5-2609v3 1.90GHz y 64GB de RAM. El software está basado en [1] y el código está disponible en http://github.com/jacintoArias/pgm2018.

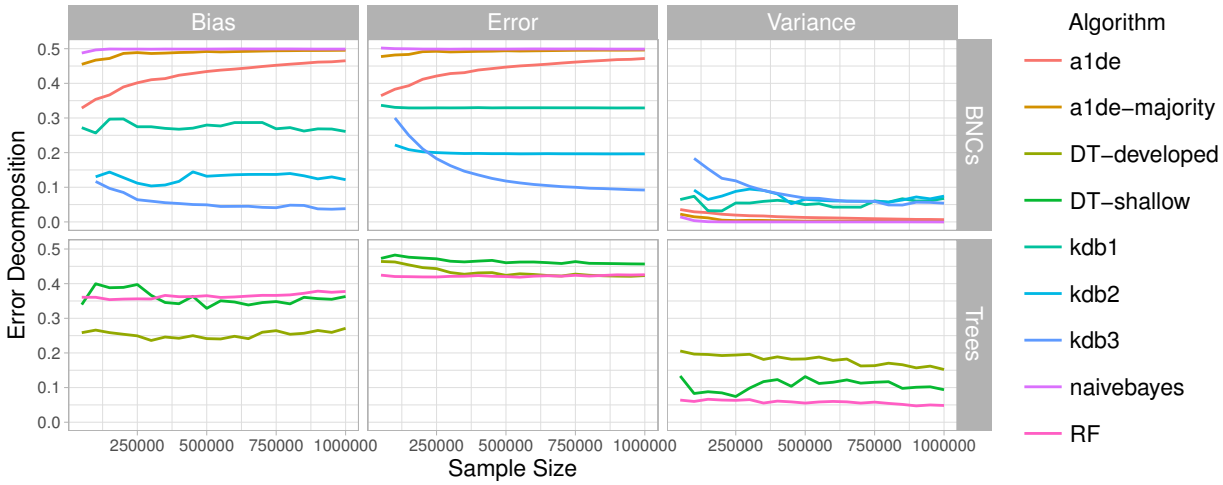

Figura 1. Evolución de sesgo y varianza para experimentos repetidos sobre muestras de mayor tamaño incrementalmente. La gráfica muestra en el eje y los valores para varianza, sesgo y error de izquierda a derecha, para la familia de modelos BNC en la parte superior y los basados en árboles en la inferior.

Problema	Casos	Atr.	Etiq.
Pokerhand	1000000	10	8
Adult	48842	15	2
Chess	28056	6	18
Letter	20000	17	26
Nursery	12960	9	5
PenDigits	10992	17	10
CensusIncome	10419	14	2
Mushrooms	8124	23	2
Musk	6598	168	2
OpticalDigits	5620	49	10
PageBlocks	5473	11	5
Spambase	4601	58	2
Hypothyroid	3772	30	4
Kr.vs.kp	3196	37	2
Splice	3190	62	3
Segment	2310	20	7
Mfeat	2000	6	2

Problema	Casos	Atr.	Etiq.
Car	1728	8	4
Contraceptive-mc	1473	10	3
German	1000	21	2
Vowel	990	14	11
Tic-Tac-Toe	958	10	2
Anneal	898	39	6
Vehicle	846	19	4
PimaIndiansDiabetes	768	9	2
BreastCancer-w	699	10	2
BalanceScale	625	5	3
CreditApproval	690	15	2
Cylinder-bands	512	39	2
Haberman	306	3	2
HouseVotes84	435	17	2
HorseColic	368	22	2
Ionosphere	351	35	2
PrimaryTumor	339	18	22

Problema	Casos	Atr.	Etiq.
Soybean	307	35	15
Haberman	306	3	2
HeartDisease-c	303	14	2
Audiology	226	70	24
New-Thyroid	215	6	3
Glass-id	214	10	3
Sonar	208	61	2
Autos	205	26	7
Wine	178	14	3
Hepatitis	155	20	2
TeachingAssistant	151	6	3
Iris	150	5	3
Promoters	106	58	2
Zoo	101	17	7
Post-operative	90	9	3
LaborNegotiations	57	17	2
LungCancer	32	57	3

Tabla I
Propiedades de los problemas utilizados en los experimentos. Los atributos continuos han sido discretizados en cuatro intervalos por igual frecuencia para los modelos BNC.

casi imperceptible, incluso la mejora parece venir de una reducción en sesgo. Observando una muestra de estas diferencias a nivel individual, Figura 3, podemos ver dos distribuciones muy diferentes, muy consistente en el caso de RF y RkDB pero casi aleatoria en A1DE. Este experimento saca a luz que A1DE al no seguir los mismos principios de diseño que otros ensembles tampoco se comporta de la manera esperada, lo que explica porque la selección de subconjuntos puede mejorar el rendimiento respecto al conjunto entero, una propiedad nada deseable y difícil de controlar en un ensemble [15].

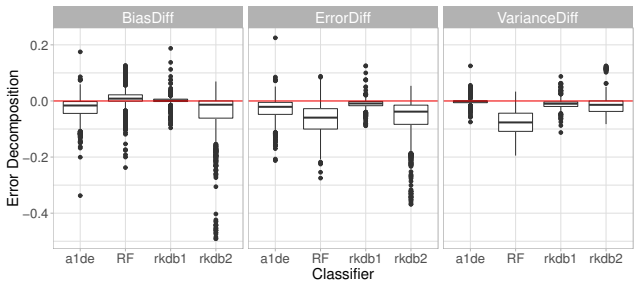

Figura 2. Distribución de las diferencias en la descomposición del error entre el modelo agregado y los diferentes modelos individuales. La línea roja en y=0 ayuda a distinguir entre diferencias positivas y negativas.

D. Sobre la Efectividad de la Selección de Modelos

Hemos observado baja consistencia en los modelos que componen un ensemble AkDE, donde unos modelos muestran propiedades deseables como bajo sesgo mientras que otros parecen ser propensos a error y tienen poco margen de varianza que reducir. Los algoritmos de selección de modelos introducidos aprovechan esta circunstancia para reducir el espacio de modelos y al mismo tiempo mejorar su rendimiento utilizando heurísticas basadas en la hipótesis de que la Información Mutua (MI) entre los atributos predictores que guían los modelos y la clase es un buen indicador de su rendimiento.

Cabe realizar entonces la pregunta de si estos métodos seleccionan entonces aquellos clasificadores propensos a mejorar en la agregación, poco sesgo y mayor varianza, o solo aquellos que minimizan el error independientemente. Para responder empíricamente hemos evaluado diversos criterios de selección de modelos de manera incremental, es decir, añadiendo cada vez un modelo adicional. Se han evaluado tres criterios wrapper: mínimo error, bias y varianza; un criterio filter basado en MI y un criterio aleatorio para establecer una base de comparación. La Tabla III muestra la suma del

Figura 3. Distribución bidimensional de sesgo (x) y varianza (y) para los modelos individuales (cruces azules) y el ensemble (punto rojo).

error sobre todos los incrementos para cada criterio y base de datos del benchmark anterior. Los resultados muestran que los mejores criterios son aquellos que minimizan el error o su componente del sesgo, mientras que la varianza, la IM y el resultado aleatorio son equivalentes. Esta evidencia reafirma nuestra hipótesis y revela que la los criterios filter basados en Información Mutua pueden no ser los más acertados.

V. Conclusiones y Trabajo Futuro

Hemos visto que los buenos resultados de AkDE en la práctica son inconsistente con la definición de un ensemble, y que la mejora en rendimiento al seleccionar modelos solo es relevante en la aplicación de técnicas wrapper que son muy costosas de computar en un escenario real. En su lugar hemos propuesto y comparado favorablemente un clasificador sencillo de tipo ensemble basado en BNCs. Los resultados muestran que es comparable a AkDE y que proporciona un recorrido de mejora que exploraremos en trabajos posteriores, especialmente en problemas de gran tamaño, donde hemos visto una clara superioridad del clasificador básico kDB en calibración de probabilidades y reducción del sesgo.

Algorithm	rank	pvalue	win	tie	loss
A1DE	2.48	-	-	-	-
RF	2.49	**9.7929e-01**	26	0	27
RkDB1	2.76	**8.7224e-01**	30	2	21
kDB1	3.58	7.1644e-03	37	0	16
RkDB2	4.75	1.8647e-09	46	0	7
kDB2	4.93	7.4299e-11	46	1	6

Tabla II

Las columnas gana, empata y pierde comparan el error obtenido por el mejor modelo (A1DE) contra los demás (p.e, A1DE gana 24 veces contra RF y pierde 17). El rango y el p-valor corresponden al test de Friedman y post-hoc (Holm) con $\alpha = 0.05$. En negrita las hipótesis rechazadas.

Criterio	Rango	p-valor	Gana	Empata	Pierde
error	1.52	-	-	-	-
bias	1.83	**3.6273e-01**	24	3	17
mi	3.75	7.8353e-11	43	0	1
variance	3.81	3.7139e-11	41	1	2
random	4.09	1.0269e-13	43	0	1

Tabla III

Comparativa de la suma del error para distintos criterios de selección en A1DE. Métricas descritas en la Tabla II.

References

[1] Jacinto Arias, Jose A. Gamez, and Jose M. Puerta. Learning distributed discrete Bayesian Network Classifiers under MapReduce with Apache Spark. *Knowledge-Based Systems*, 117:16 – 26, 2017.

[2] Eric Bauer, Ron Kohavi, Philip Chan, Salvatore Stolfo, and David Wolpert. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36:105–139, 1999.

[3] Concha Bielza and Pedro Larrañaga. Discrete Bayesian Network Classifiers: A Survey. *ACM Comput. Surv.*, 47(1):5:1–5:43, jul 2014.

[4] Leo Breiman. Arcing classifiers. *Annals of Statistics*, 26(3):801–849, 1998.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] Shenglei Chen, Ana M. Martinez, Geoffrey I. Webb, and Limin Wang. Sample-Based Attribute Selective AnDE for Large Data. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):172–185, 2017.

[7] Shenglei Chen, Ana M. Martínez, Geoffrey I. Webb, and Limin Wang. Selective AnDE for large data learning: a low-bias memory constrained approach. *Knowledge and Information Systems*, 50(2):475–503, 2017.

[8] Dua Dheeru and Efi Karra. UCI Machine Learning Repository, 2017.

[9] Thomas G. Dietterich. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees. *Machine Learning*, 40:139–157, 2000.

[10] Zhiyi Duan and Limin Wang. K-Dependence Bayesian Classifier Ensemble. *Entropy*, 19(12), 2017.

[11] A M Martínez, G I Webb, S Chen, and N A Zaidi. Scalable learning of Bayesian network classifiers. *Journal of Machine Learning Research*, 17:1–30, 2016.

[12] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, 17(34):1–7, 2016.

[13] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 2014.

[14] Mehran Sahami. Learning Limited Dependence Bayesian Classifiers. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, KDD'96, pages 335–338. AAAI Press, 1996.

[15] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

[16] Geoffrey I. Webb. MultiBoosting: a technique for combining boosting and wagging. *Machine Learning*, 40(2):159–196, 2000.

[17] Geoffrey I. Webb, Janice R. Boughton, and Zhihai Wang. Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

[18] Geoffrey I. Webb, Janice R. Boughton, Fei Zheng, Kai Ming Ting, and Houssam Salem. Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification. *Machine Learning*, 86(2):233–272, oct 2012.