# shinytests: Una herramienta gráfica para la comparación estadística en minería de datos

Jacinto Carrasco*, Salvador García* and Francisco Herrera*
* Dpto. de Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada
Granada, España
Email: {jacintocc, salvagl, herrera}@decsai.ugr.es

***Resumen*—Los test estadísticos constituyen el procedimiento más fiable para la validación de los resultados obtenidos en múltiples escenarios. En particular, debido a su robustez y aplicabilidad, los test no paramétrico son una herramienta habitual y útil en el proceso de diseño y evaluación de los algoritmos de aprendizaje automático para ámbitos tanto de clasificación como de optimización. Las nuevas tendencias como el uso de test bayesianos y la observación de la distribución del parámetro de interés representan un enfoque a tener en cuenta.**

**En esta contribución se presenta la aplicación shiny de R *shinytests*, la cual integra test bayesianos y no paramétricos para facilitar la realización de test estadísticos en la comparación de algoritmos de aprendizaje automático y optimización.**

***Index Terms*—Test estadísticos, test bayesianos, software, shinyapp, R**

## I. Introducción

En el desarrollo de algoritmos de aprendizaje automático y de optimización existe una necesidad creciente de validar y examinar la incertidumbre presente en estos procesos. Los test estadísticos son la herramienta recomendada para asegurar que las conclusiones obtenidas de los correspondientes experimentos no están sesgadas por la intención del investigador o se han obtenido por una cuestión de azar [1].

Existen numerosos test que pueden usarse para este propósito, los cuales pueden ser clasificado en dos grandes categorías: Los test frecuentistas, principalmente los test de hipótesis nula [2] o NHST (*Null Hypothesis Statistical Tests*), y los test bayesianos [3]. El primer grupo está subdividido en test paramétricos, que no se tendrán en cuenta en este artículo por estar suficientemente extendidos, y los test no paramétricos [4], los cuales requieren unas condiciones de aplicabilidad menos estrictas que los test paramétricos aunque esto se traduzca en ocasiones en una menos habilidad para encontrar diferencias existentes entre los resultados de los algoritmos [5]. Estos prerrequisitos son habitualmente la normalidad de la muestra o la homocedasticidad, condiciones que pueden ser comprobadas mediante test no paramétricos como el test de Kolmogorov-Smirnov, así como otros test sobre ciertas propiedades de la muestra, como la aleatoriedad de una muestra o el ajuste a una distribución. Para la comparación del desempeño de algoritmos se usan habitualmente los test de Wilcoxon de rangos con signo o el test de Friedman para la comparación de múltiples algoritmos. Además de un resumen descriptivo de los test incluidos, se incluye un caso del uso de la aplicación *shiny* de `R` para la aplicación de los test bayesianos y la obtención de las gráficas correspondientes de la distribución del parámetro de interés, lo que ayuda a comprender estos test y sintetiza la información dada por éstos.

Esta contribución está organizada de la siguiente manera. En la Sección II se introducen los conceptos estadísticos necesarios y se describen los diferentes test estadísticos. En la Sección III se describen los principales métodos incluidos en la aplicación y se muestran varios ejemplos de su uso. En la Sección IV se concluye la contribución.

Este trabajo se ha sustentado por el proyecto de investigación TIN2017-89517-P. J. Carrasco disfruta de una beca FPU del Ministerio de Educación de España.

## II. Antecedentes

En la Subsección II-A se introducen conceptos básicos de estadística que den soporte al resto del artículo. A continuación, en la Subsección II-B se describe el uso de los test frecuentistas clásicos para la comparación de algoritmos, con especial interés en los test no paramétricos. Los test bayesianos para la comparación de la eficacia de algoritmos se incluye en la Subsección II-C.

### *II-A. Conceptos preliminares*

En la inferencia estadística estamos interesados en obtener una predicción fiable a partir de los datos, por lo que debemos evitar llegar a conclusiones erróneas producidas por efectos aleatorios. Los principales conceptos a tener en cuenta son [2]:

- Los resultados de los algoritmos implicados en la comparación constituyen una **muestra**. Esta representa el desempeño del algoritmo sobre uno o varios problemas, ya sea la medida de ajuste sobre un problema de optimización o bien el acierto sobre un conjunto de datos en un problema de clasificación. Desde el punto de vista estadístico, esta muestra proviene de una distribución de probabilidad desconocida y será usada para inferir información relevante.
- Al hablar del **parámetro** de interés, o de la distribución de un cierto parámetro, nos referimos a la medida usada para evaluar la diferencia entre los resultados de los algoritmos, o bien el ajuste de una muestra con respecto a una distribución.

- Un enfoque frecuentista para inferir información relevante consiste en el cálculo de un estadístico, es decir, un estimador de una característica de la distribución.
- La **distribución** de la cual obtenemos la muestra es desconocida, por lo que los estadísticos se usarán para estimar el parámetro de interés.

### *II-B. Test frecuentistas*

Los test frecuentistas son la herramienta más común en la comparación del desempeño de algoritmos hasta ahora [6]. En ellos, se establece una hipótesis nula ($\mathcal{H}_0$) y una hipótesis alternativa ($\mathcal{H}_1$). Entonces, haciendo uso de una muestra se calcula la probabilidad de obtener una muestra tan alejada de la hipótesis nula como la que disponemos asumiendo que $\mathcal{H}_0$ es cierta. Esta probabilidad se conoce como $p$-valor [2]. Entonces, si la probabilidad obtenida es menor que un valor fijo $\alpha$ (normalmente 0,05), se rechaza $\mathcal{H}_0$, mientras que de otra manera no hay suficiente evidencias como para rechazar la hipótesis nula.

- Además, como estamos interesados en la comparación estadística, debemos prestar atención a las propiedades de los test estadísticos. Se define el **error de tipo I** como la probabilidad de rechazar la hipótesis nula $\mathbf{H}_0$ cuando es cierta y el **error de tipo II** cuando $\mathbf{H}_0$ no se rechaza y es falsa.
- La principal medida de para comparar la calidad de un test es la **potencia**, es decir, la probabilidad de rechazar $\mathcal{H}_0$. Estaremos interesados en obtener una mayor potencia manteniendo el error de tipo I, que se representa con el parámetro $\alpha$.

*II-B1. Test paramétricos:* Estos test parten de la suposición de que la muestra proviene de una familia conocida de distribuciones, habitualmente la distribución normal. Cuando se cumple la hipótesis de normalidad se obtiene un test más potente. Los principales test que se corresponden con esta categoría son el $t$-test para la comparación de dos muestras pareadas y el test ANOVA para la comparación de múltiples algoritmos. En ambos test la hipótesis nula consiste en la equivalencia de la media del desempeño de los algoritmos involucrados.

*II-B2. Test no paramétricos:* Los test no paramétricos no asumen que la muestra provenga de una distribución de una familia conocida [7], lo que se traduce en que se tengan condiciones menos restrictivas sobre la muestra, como la simetría o la continuidad [8]. En consecuencia, los test no paramétricos son más robustos que los paramétricos, puesto que normalmente no se dan las condiciones necesarias para su uso.

Para asegurarnos de que estamos usando correctamente el test ANOVA o el $t$-test debemos comprobar la normalidad de la muestra, para lo que pueden usarse test sobre la bondad del ajuste para, al menos, no rechazar esta hipótesis. Sirven para ello test de bondad del ajuste como los test de Kolmogorov-Smirnov, Shapiro-Wilk y D'Agostino-Pearson [9].

El test no paramétrico recomendado para la comparación de algoritmos depende del número de algoritmos a comparar y distintas situaciones implicadas:

- **Test de signo y Test de Rangos con signo de Wilcoxon**: El test de signo es un análogo del $t$-test simple y el test de Wilcoxon es la versión análoga del $t$-test pareado.
- **Test de Friedman**: Este test cumple con la función análoga al test paramétrico ANOVA. Se realiza una comparación de $k$ algoritmos en $n$ problemas (conjuntos de datos o funciones *benchmark*). El estadístico se calcula en base al orden de los algoritmos para cada problema. El test de Iman-Davenport constituye una propuesta más potente basada en el test de Friedman.
- **Test de Friedman de rangos alineados**: Esta mejora del test de Friedman usa el orden de los resultados en todos los problemas, lo que se traduce en que es tenida en cuenta la dificultad de cada problema.

En el caso de que existan diferencias significativas en la realización de test para múltiples algoritmos y se rechace la hipótesis nula $\mathcal{H}_0$, nuestro propósito será discernir dónde se encuentran estas diferencias. Para este paso es necesario un ajuste en el $p$-valor obtenido para mantener el control sobre el *Family-wise Error Rate* (FWER). Algunos ejemplos de test post-hoc son los de Bonferroni-Dunn, Holm, Holland, Hochberg o Li [6], [10], [11].

### *II-C. Test bayesainos*

Un enfoque distinto es el propuesto por Benavoli *et al.* [12]. La principal diferencia es que no se establece una hipótesis nula sobre el parámetro de interés para realizar un test de hipótesis nula, sino que se obtiene una distribución de probabilidad sobre el parámetro de interés.

*II-C1. Comparación con los test frecuentistas:* Según Benavoli [13], las principales diferencias que se podrían identificar son:

- En los test frecuentistas, las decisiones sobre la significatividad de un test son dicotómicas, basadas en el $p$-valor y el nivel $\alpha$ de significatividad. En la estadística bayesiana, no existe un umbral fijo para el rechazo de la hipótesis nula sino la distribución del parámetro, de donde obtenemos la probabilidad de que la hipótesis nula sea cierta.
- En la aplicación de los NHST existe una confusión habitual, y es que el $p$-valor no representa la probabilidad de que se dé la hipótesis nula, sino, asumiendo que la hipótesis nula es cierta, obtener una muestra tan alejada de $\mathcal{H}_0$ como la que disponemos. Normalmente querremos responder la primera pregunta, la cual obtenemos usando los test bayesianos.
- Una crítica común a los NHST es que el tamaño del efecto y el tamaño de la muestra no son distinguibles. Esto significa que un efecto tan pequeño como sea necesario puede ser considerado como significante si se añaden suficientes instancias a la muestra. Como el tamaño de la muestra depende del investigador, se

podría variar el número de observaciones hasta obtener el resultado esperado.
- Los NHST no ofrecen información cuando la hipótesis nula no se rechaza. En esta situación, no podríamos decir que no hay diferencia entre las muestras, sino que no disponemos suficientes evidencias para rechazar la hipótesis nula. En cambio, en los test bayesianos la distribución del parámetro es informativa aunque no indique una suficiente diferencia entre los algoritmos.
- El proceso para realizar un test bayesianos consiste en establecer un modelo probabilístico *a priori* (basándonos en la información que disponemos o con una distribución *a priori* poco informativa), calcular en interpretar la distribución *a posteriori* basándonos en los datos disponibles, y evaluar el modelo.

*II-C2. t-test bayesiano correlado:* Esta versión bayesiana del $t$-test se usa para comparar los resultados de dos algoritmos de clasificación en un escenario de validación cruzada con $k$ *folds* partition [14]. Este test tiene en consideración la correlación entre los distintos folds y parte de la hipótesis de que los datos vienen de una distribución gaussiana multivariante cuya matriz de covarianza depende de la correlación $\rho$ entre los *folds*. Debido a que $\rho$ no puede estimarse a partir de los datos, se utiliza la heurística sugerida por Nadeau y Bengio [15] y $\rho = \frac{n_{test}}{n_{tot}}$, esto es, el número de instancias en la partición de evaluación partido por le número total de instancias. Se parte de una distribución Normal-Gamma como la distribución *a priori* de la diferencia entre los algoritmos, por lo que se obtiene *a posteriori* una distribución de Student sobre la diferencia entre las medias $\mu$. Debemos además considerar la posibilidad de que no haya una diferencia significativa entre el desempeño de ambos algoritmos, por lo que se debe definir una región de equivalencia (a la que llamaremos *rope*, por *region of practical equivalence*), $[r_{min}, r_{max}]$, definida para $\mu$, y las relaciones entre los algoritmos se considerarán en términos de la *rope*. Por ejemplo, para $a_1, a_2$ algoritmos involucrados en la comparación, $P(a_1 \gg a_2) = P(\mu > r_{max})$ o $P(a_1 = a_2) = P(\mu \in \text{rope})$, donde la relación entre los algoritmos se refiere a la comparación del desempeño de ambos algoritmos. La *rope* por tanto nos permite realizar decisiones automáticas, aunque volviendo de esta manera a la pérdida de información y las decisiones dicotómicas. Sin embargo, en esta ocasión la interpretación de las probabilidades son directas y los límites para las decisiones pueden variar en función del contexto.

*II-C3. Test bayesiano de signo:* La versión bayesiana del test no paramétrico de signo hace uso del Proceso de Dirchlet (DP, *Dirichlet Process*) [16]. Podemos entender este proceso como una distribución de probabilidad sobre una familia de distribuciones de probabilidad, de manera que la inferencia se realiza en dos pasos.

- En primer lugar se obtiene la función de densidad de la distribución *a posteriori* como una combinación lineal de deltas de Dirac centradas en las observaciones, cuyos pesos provienen de una distribución de Dirichlet.
- Entonces, aproximamos la anterior función de probabilidad *a posteriori* como una probabilidad *a posteriori* de la que podemos calcular la probabilidad del parámetro de pertenecer a cada región de interés.

*II-C4. Test bayesiano de rangos con signo:* La versión bayesiana de rangos con signo tiene el mismo *background* estadístico que el test bayesiano de signo. También hace uso del DP como el método para realizar la inferencia a partir del datos. La diferencia radica en el hecho de que el test de rangos con signo usa dos muestras y la comparación entre ellas en el cómputo de las probabilidades de las posibles relaciones entre algoritmos. En este test no obtenemos una fórmula para la distribución *a posteriori*, pero podemos obtenerla muestreando los pesos de la distribución de Dirichlet.

*II-C5. Test bayesiano de Friedman:* El test bayesiano de Friedman [17] realiza un procedimiento similar a los descritos previamente para comprobar si es factible que el parámetro con la media del orden de la clasificación de cada algoritmo se quede en una región cercana al punto medio que constituye la hipótesis nula en un test frecuentista ($[(m+1)/2, \ldots, (m+1)/2]$). Si el parámetro de interés $\mu$ no se encuentra en esta región, existirá una diferencia significativa.

## III. Aplicación *shiny*

Esta sección contiene en la Subsección III-A una descripción de la aplicación *shiny* desarrollada y su base, el paquete `rNPBST` [18]. La Subsección III-B contiene una descripción de la utilización de la aplicación para la comparación de algoritmos, principalmente sobre el uso de los métodos bayesianos, debido a que éstos son menos conocidos y su uso no está extendido.

### III-A. Paquete rNPBST

El paquete `rNPBST` ha sido desarrollado inicialmente como un wrapper de la biblioteca `JavaNPST` desarrollada por Derrac *et al.* [19]. Es una biblioteca en Java que integra un extensivo conjunto de test no paramétricos de diferentes familias y con diferentes propósitos.

En la biblioteca original en Java sólo se incluyen test no paramétricos aunque se han añadido en el paquete de `R` varios test bayesianos y métodos asociados de visualización a través de `ggplot` y `ggtern` [20]. Los test se clasifican 11 categorías atendiendo al propósito de los test o el tipo de dato usando Tabla I.

El paquete `rNPBST` está disponible en un repositorio de Github[1] y se puede instalar usando el paquete `devtools` y ejecutando en `R`:

```
devtools::install_github("JacintoCC/rNPBST")
```

### III-B. Ejemplo de uso

Para la ejecución de la aplicación *shiny* será necesario ejecutar en `R` la siguiente función del paquete `shiny`:

```
shiny::runGitHub(repo = "shinytests",
                 username = "JacintoCC")
```

[1] http://www.github.com/JacintoCC/rNPBST

Tabla I: Test incluidos en la versión actual de `rNPBST`

| Family | Test |
|---|---|
| Test de aleatoriedad | Número de rachas<br>Rachas crecientes y decrecientes<br>Rachas crecientes y decrecientes (Mediana)<br>Von Neumann |
| Test de bondad del ajuste | Chi-Squared<br>Kolmogorov-Smirnov<br>Lilliefors<br>Anderson-Darling |
| Una muestra y muestras pareadas | Cuantil de confianza<br>Cuantil de la población<br>Test de signo<br>Test de Wilcoxon de rangos con signo |
| Procedimientos general de dos muestras | Wald-Wolfowitz<br>Test de medias<br>Control Median<br>Kolmogorov-Smirnov |
| Problema de escala | David-Barton<br>Freund-Ansari-Bradley<br>Mood<br>Klotz<br>Siegel-Tukey<br>Sukhatme |
| Problema de posición | Wilcoxon Rank-Sum<br>van der Waerden |
| Independencia de muestras | Extended Median test<br>Kruskal-Wallis<br>Jonckheere-Terpstra<br>Charkraborti-Desu |
| Muestras bivariadas | Kendall<br>Daniel Trend |
| Múltiples clasificadores | Friedman<br>Iman-Davenport<br>Rangos alineados de Friedman<br>Page<br>Coeficiente de concordancia<br>Concordanza incompleta<br>Correlación parcial |
| Conteo de datos | Coeficiente de contingencia<br>Test exacto de Fisher<br>McNemar<br>Test de igualdad multinomial<br>Ordered Equality test |
| Bayesianos | t-test bayesiano correlado<br>Test bayesian de signo<br>Test bayesiano de rangos con signo<br>Test bayesiano de Friedman |

Entonces se descargan automáticamente los paquetes necesarios y se abre en el navegador la aplicación. Se incluye la posibilidad de subir un fichero `.csv` para realizar los test estadísticos, así como seleccionar distintos test. A medida que vamos realizando cambios en el conjunto de datos introducido, o seleccionando el test a realizar en el menú lateral, se actualizarán automáticamente los resultados. También podremos seleccionar la opción de incluir un gráfico en algunos de los test bayesianos, el cual podremos descargar.

Para ejemplificar el uso de algunos test, presentamos un un estudio comparativo entre cinco algoritmos clásicos para problemas de clasificación. Los algoritmos incluidos en la comparación están descritos en la Tabla II. Los resultados de cada algoritmo en los distintos conjuntos de datos se incluyen en el paquete `rNPBST` y como conjunto por defecto en la aplicación `shinytests` para poder ejemplificar su uso. La medida usada ha sido la *accuracy*. Se incluye para cada algoritmo descrito en la Tabla II una tabla con los resultados en las particiones 5-dob-cv [21] de algunos de los conjuntos de datos disponibles[2] para clasificación en el repositorio KEEL

[2]abalone, australian, automobile, balance, breast, bupa, car, cleveland, crx, dermatology, german, glass, hayes-roth, heart, ionosphere, led7digit, letter, lymphography, mushroom, optdigits, satimage, spambase, splice, tic-tac-toe, vehicle, vowel, wine, yeast and zoo

Tabla II: Algoritmos comparados en el conjunto de datos de ejemplo

| Algoritmo | Descripción | Conjunto de datos |
|---|---|---|
| `multinom` | Regresión logística, del paquete `nnet`. | `results.lr` |
| `knn` | Biblioteca `class`. Parám. $k = 1$, $l = 0$. | `results.knn` |
| `randomForest` | Biblioteca `randomForest`. Parám. `mtray` $= \sqrt{p}$. | `results.rf` |
| `nnet` | Biblioteca `nnet` library. | `resuts.nnet` |
| `naiveBayes` | Clasificador Naïve Bayes del paquete `e1071`. | `results.nb` |

Tabla III: Wilcoxon Rank Sum test

| Wilcoxon Rank Sum test | | |
|---|---|---|
| data.name | | results[, 1:2] |
| statistic | | 665.00 |
| p.value | Asymptotic Left Tail | 0.001565 |
| | Asymptotic Right Tail | 0.998512 |
| | Asymptotic Double Tail | 0.003129 |

[22] y en las particiones creadas para ello en este repositorio. Los resultados de las diferentes particiones se resumen usando el promedio en el conjunto de datos `results`. Se han mantenido por separado los conjuntos para todas las particiones para usarlos en el $t$-test correlado bayesiano.

*III-B1. Análisis de muestras pareadas:* Para una comparación paramétrica entre dos algoritmos podemos usar el test Wilcoxon Rank-Sum. El resultado de aplicar dicho test al conjunto de datos por defecto, seleccionando las dos primeras columnas (correspondientes a la comparación de la regresión logística y el KNN) se incluye en la Tabla III, puesto que en la aplicación *shiny* se incluye tanto una tabla HTML mostrando los resultados, como el código TEXque produce dicha tabla.

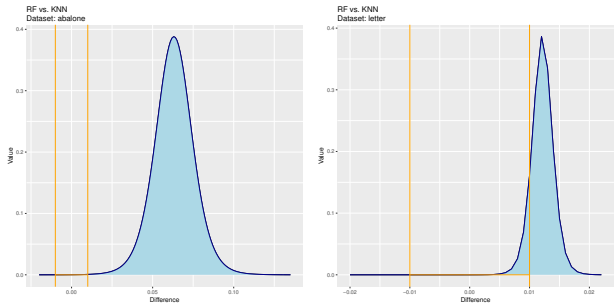
Vemos en la tabla Tabla III que la hipótesis nula $\mathcal{H}_0 : \mu_{LR} = \mu_{KNN}$ puede rechazarse debido a que el $p$-valor asintótico es menor que $0{,}05$, de manera que este test identifica una diferencia significativa entre estos dos algoritmos. Para determinar cuál obtiene mejores resultados, podemos mirar el $p$-valor para las hipótesis alternativas direccionales y concluimos quela regresión logística obtiene mejores resultados debido a que no podemos rechazar $\mathcal{H}_\prime$ cuando la hipótesis alternativa es $\mathcal{H}_1 : \mu_{LR} > \mu_{KNN}$.

*III-B2. Test para comparaciones múltiples:* Como se ha descrito en la Sección II, la hipótesis nula del test de Friedman es la equivalencia de las medianas de los diferentes algoritmos, por lo que un $p$-valor menor que un test significa que la hipótesis nula puede ser rechazada y existe una diferencia entre los algoritmos comparados. Se incluye en la Tabla IV los resultados del test de Friedman.

*III-B3. $t$-test bayesiano correlado:* En este test comparamos los resultados obtenidos por `random forest` y `knn` para un único dataset. Con el resultado de este test podemos obtener la Fig. 1 con la diferencia entre estos algoritmos.

Tabla IV: Friedman test

| Friedman test | | |
|---|---|---|
| data.name | | df |
| statistic | s | 2812.00 |
| | q | 39.06 |
| p.value | | 6.789e-08 |




(a) Abalone (b) Letter

Figura 1: Distribución de RF vs KNN para dos conjuntos de datos

Tabla V: Bayesian correlated t-test

| Bayesian correlated t-test | | |
|---|---|---|
| probabilities for abalone dataset | left | 4.962e-05 |
| | rope | 4.407e-04 |
| | right | 9.995e-01 |
| probabilities for letter dataset | left | 1.378e-07 |
| | rope | 1.105e-01 |
| | right | 8.895e-01 |
| rope | | -0.01 |
| | | 0.01 |

La distribución *a posteriori* del parámetro de interés muestra cómo con un 99,9 % `random forest` tiene un mejor resultado que `knn` en este conjunto de datos. La distribución de la diferencia se muestra en Fig. 1 para los conjuntos de datos `abalone` y `letter`. En este segundo conjunto de datos, aunque `random forest` también obtiene un mejor resultado que `knn`, hay una mayor probabilidad de que ambos algoritmos obtengan el mismo resultado que en el primer conjunto de datos. Para los test bayesianos también obtenemos la Tabla V con los resultados, en este caso es la probabilidad de pertenencia a cada región de interés.

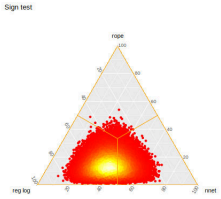
*III-B4. Test bayesiano de signo:* Para este test usamos los resultados promediados de dos algoritmos en todos los conjuntos de datos.

Hay una mayor probabilidad para la hipótesis de que la regresión logística obtenga un resultado mejor que la red neuronal, aunque podemos comprobar que las diferencias son pequeñas en la Tabla VI. En la Fig. 2 se observa una muestra de la distribución *a posteriori* y podemos comprobar cómo hay una mayor concentración de puntos en la región izquierda, que corresponde con la situación en la que la regresión logística obtiene un mejor resultado que la red neuronal. En la Fig. 2b la comparación se realiza entre `neural network` and `random forest`. Hay incluso una concentración mayor en la región izquierda, lo que nos dice que hay incluso una mayor probabilidad de que `random forest` obtenga un mejor resultado que `neural network`.

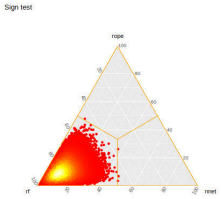
*III-B5. Test bayesiano de rangos con signo:* Repetimos la experimentación usando el test bayesiano de rangos con signo, mostrando la gráfica asociada en la Fig. 3 y los resultados numéricos en la Tabla VII.

Tabla VI: Bayesian Sign-test

| test | | |
|---|---|---|
| probabilities | left | 0.4780 |
| | rope | 0.1444 |
| | right | 0.3777 |




(a) Red neuronal vs regresión logística




(b) Red neuronal vs random forest

Figura 2: Muestra de la distribución *a posteriori* del test bayesiano de signo

La probabilidad *a posteriori* para la región izquierda es ligeramente mayor que la probabilidad para el test bayesiano de signo, por lo que tenemos una mayor certeza para esta comparación usando el test bayesiano de rangos con signo. Como se puede ver en la Fig. 3, la distribución está desplazada hacia la izquierda, por lo que se espera una mayor potencia de este test con respecto al test bayesiano de signo.

*III-B6. Test bayesiano de Friedman:* En la Tabla VIII se incluye los resultados del test bayesiano de Friedman. En esta tabla se incluyen el orden medio de clasificación para cada algoritmo y la hipótesis seleccionada $h$, que en este caso $h = 1$, lo que significa que se rechaza que el parámetro pertenezca a la región de igual *ranking* para todos los algoritmos.

## IV. Conclusiones

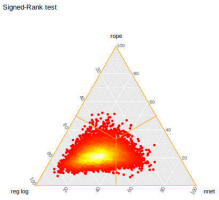
La experimentación inherente a la naturaleza del aprendizaje automático y el rápido crecimiento del número de algoritmos propuestos conlleva la necesidad de establecer un método claro de comparación del desempeño de estos algoritmos y

Tabla VII: Bayesian Signed-Rank test

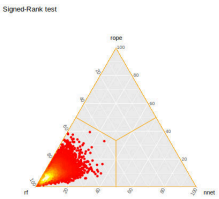
| test | | |
|---|---|---|
| probabilities | left | 0.4921 |
| | rope | 0.2220 |
| | right | 0.2859 |

Tabla VIII: Bayesian Friedman test

| Bayesian Friedman test | | | | | |
|---|---|---|---|---|---|
| h | 1 | | | | |
| meanranks | 3.1 | 2.433 | 4.467 | 2.233 | 2.767 |




(a) Red neuronal vs regresión logística




(b) Red neuronal vs random forest

Figura 3: Muestra de la distribución *a posteriori* del test bayesiano de rangos con signo.

una herramienta software que facilite este procedimiento. En esta contribución presentamos la aplicación *shiny* de `R`, cuyo principal objetivo es proporcionar una herramienta gráfica para los principales test no paramétricos y bayesianos existentes en el paquete `rNPBST`, de manera que se disponga de un software para investigadores interesados en comparar nuevo algoritmos. Como tareas futuras se trabajará en añadir nuevos test a esta aplicación.

## Referencias


[1] N. Japkowicz and M. Shah, eds., *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[2] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. crc Press, 2003.

[3] J. M. Bernardo and A. F. Smith, *Bayesian Theory*. IOP Publishing, 2001.

[4] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[5] J. Luengo, S. García, and F. Herrera, "A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7798–7808, 2009.

[6] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[7] F. Pesarin and L. Salmaso, *Permutation Tests for Complex Data: Theory, Applications and Software*. John Wiley & Sons, 2010.

[8] E. Kasuya, "Wilcoxon signed-ranks test: Symmetry should be confirmed before the test," *Animal Behaviour*, vol. 79, pp. 765–767, Mar. 2010.

[9] J. Pizarro, E. Guerrero, and P. L. Galindo, "Multiple comparison procedures applied to model selection," *Neurocomputing*, vol. 48, no. 1, pp. 155–173, 2002.

[10] S. Garcia and F. Herrera, "An Extension on"Statistical Comparisons of Classifiers over Multiple Data Sets"for all Pairwise Comparisons," *Journal of Machine Learning Research*, vol. 9, no. Dec, pp. 2677–2694, 2008.

[11] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3 – 18, 2011.

[12] A. Benavoli and C. P. de Campos, "Statistical Tests for Joint Analysis of Performance Measures," in *Advanced Methodologies for Bayesian Networks - Second International Workshop, AMBN 2015, Yokohama, Japan, November 16-18, 2015. Proceedings*, pp. 76–92, 2015.

[13] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a Change: A Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis," *Journal of Machine Learning Research*, vol. 18, no. 77, pp. 1–36, 2017.

[14] G. Corani and A. Benavoli, "A Bayesian approach for comparing cross-validated algorithms on multiple data sets," *Machine Learning*, vol. 100, no. 2-3, pp. 285–304, 2015.

[15] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, vol. 52, no. 3, pp. 239–281, 2003.

[16] A. Benavoli, G. Corani, F. Mangili, M. Zaffalon, and F. Ruggeri, "A Bayesian Wilcoxon signed-rank test based on the Dirichlet process," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1026–1034, 2014.

[17] A. Benavoli, G. Corani, F. Mangili, and M. Zaffalon, "A Bayesian nonparametric procedure for comparing algorithms," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1264–1272, 2015.

[18] J. Carrasco, S. García, M. del Mar Rueda, and F. Herrera, "rNPBST: An R Package Covering Non-parametric and Bayesian Statistical Tests," in *Hybrid Artificial Intelligent Systems: 12th International Conference, HAIS 2017, La Rioja, Spain, June 21-23, 2017, Proceedings* (F. J. Martínez de Pisón, R. Urraca, H. Quintián, and E. Corchado, eds.), pp. 281–292, Cham: Springer International Publishing, 2017.

[19] J. Derrac, S. García, and F. Herrera, "JavaNPST: Nonparametric Statistical Tests in Java," *ArXiv e-prints*, Jan. 2015.

[20] N. Hamilton, *Ggtern: An Extension to 'Ggplot2', for the Creation of Ternary Diagrams*. 2018. R package version 2.2.2.

[21] E. Alpaydin, "Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, vol. 11, pp. 1885–1892, Nov. 1999.

[22] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2010.