II CONVENCIÓN CIENTÍFICA INTERNACIONAL
"II CCI UCLV 2019"

DEL 23 AL 30 DE JUNIO DEL 2019
CAYOS DE VILLA CLARA, CUBA

II CONFERENCIA INTERNACIONAL DE PROCESAMIENTO DE LA INFORMACIÓN "CIPI 2019"

Aprendizaje de funciones de distancia para problemas de predicción con salidas múltiples mediante el descenso del gradiente estocástico

Distance metric learning in multi-output learning through stochastic gradient descent

Hector Gonzalez[1*], Carlos Morell[2], Francesc J. Ferri[3]

[1]Universidad de las Ciencias Informáticas (UCI), La Habana, Cuba. hglez@uci.cu
[2]Universidad Central Marta Abreu (UCLV), Villa Clara, Cuba. cmorellp@uclv.edu.cu
[3]Dept. Informática, Universitat de Valéncia. Spain. francesc.ferri@uv.es

Resumen
Se propone un método eficiente y robusto para problemas de regresión con salidas múltiples usando distancias y vecinos más próximos. Para ello se aprende una función de distancia en el espacio de entrada reformulándolo como un problema de optimización con margen máximo. En concreto, se aplica el método de descenso por gradiente estocástico al problema primal y se proponen diversas heurísticas para preservar una cierta relación de orden entre los valores de entrada y los de salida en entornos locales de los elementos del conjunto de entrenamiento. La experimentación sugiere que el método resultante es robusto y eficiente y podría ser extendido para ser aplicado sobre volúmenes de datos a gran escala en el contexto de BigData.

Abstract
An efficient and robust method is proposed for multitarget regression problem based on distance metric learning and nearest neighbors. This formulation, learn a distance function in the input space thorugth an optimization problem with large margin. Specifically, the stochastic gradient descent method is applied in the primal problem and severals heuristic are proposed to preserve the order relationship between the input and output varibles in local environments of the training set. Experimentation suggests that the resulting method is robust and efficient and could be extended to be applied over large scale data volumes in the BigData context.

Palabras claves: *Regresión con Salidas Múltiples; Gradiente Descendente Estocástico; Aprendizaje de Funciones de Distancia; Regresión basada en la Regla de los Vecinos más Cercanos.*

Keywords: *Multi-output regression; Stochastic gradient descent; Distance-based learning; Nearest-neighbor regression.*

1. Introducción

En los últimos años ha habido un crecimiento considerable de las investigaciones en aprendizaje de funciones de distancia dentro del área del Aprendizaje Automático (*Machine Learning*). El objetivo que se plantea es aprender una función, $d_W(.,.)$, que permita medir de manera adecuada disimilitudes entre pares de objetos. Esta función es comúnmente una distancia de Mahalanobis, lo que transforma el problema en la estimación de una determinada matriz, W (semidefinida positiva), a partir de la información subyacente en los datos. La función de distancia obtenida una vez resuelto el problema, puede ser empleada por métodos de clasificación basados en instancias como los algoritmos *k-Nearest Neighbors (k-NN)* Cover and Hart (1967) o en algoritmos de agrupamiento como *k-Means* MacQueen et al. (1967). En el contexto del aprendizaje de funciones de distancia se han publicado trabajos de revisión como los propuestos por Kulis Kulis (2012) y Bellet Bellet et al. (2014, 2015), los cuales centran su atención en los problemas de clasificación. También se hace una revisión exhaustiva del estado del arte en el contexto del aprendizaje de funciones de distancia para regresión en Nguyen et al. (2016). Por otro lado, en la reciente revisión Li and Tian (2018), se establece una taxonomía de esta disciplina y se desarrolla una evaluación empírica de los algoritmos de aprendizaje de distancias más representativos dentro de la taxonomía propuesta.

Para modelar el problema de aprender una función de distancia, se emplea normalmente un enfoque supervisado donde se tiene en cuenta la información de la variable objetivo dentro del conjunto de datos de entrenamiento. En este sentido, se espera que objetos de una misma clase (según la variable objetivo) se encuentren cercanos según el espacio métrico asociado a la función de distancia. Para problemas de clasificación, resulta natural imponer restricciones, en forma de pares o ternas relativas, de manera que se puedan discernir pares similares y disimilares o quién está más cerca de quién en una determinada terna Kulis (2012); Bellet et al. (2014); Weinberger and Saul (2009); Davis et al. (2007); Tarlow et al. (2013); Nguyen and Guo (2008); Nguyen et al. (2016); Koestinger et al. (2012). Por otra parte, en los problemas de regresión no resulta tan natural aplicar directamente este tipo de heurísticas para encontrar una función de distancia adecuada. Por ello, existen muy pocas contribuciones en el campo del aprendizaje de funciones de distancias para regresión, destacándose el algoritmo MLKR y sus extensiones, basado en el estimador de Nadayara-Watson y la estimación directa de la función de distancia por medio de la regresión multivariada Weinberger and Tesauro (2007); Gonzalez et al. (2018).

Un modelo reciente en el que se aplican restricciones basadas en ternas en el contexto de regresión múltiple es el *Distance Metric Learning for Multitarget Prediction*, DMLMTP Gonzalez et al. (2016). Este algoritmo se basa en el LQRF de Schultz y Joachim Schultz and Joachims (2004) y en otros trabajos directamente relacionados como Perez-Suay et al. (2013); Wang et al. (2015) pero extendido al caso de múltiples variables de salida. Las ideas principales que caracterizan el algoritmo DMLMTP son las siguientes:

- La predicción se realiza mediante el promedio ponderado en la vecindad de cada punto en función de la distancia aprendida.
- Para el aprendizaje se consideran las tripletas de objetos dentro de vecindades o entornos locales así como sus distancias relativas según la función que se pretende aprender.
- Se sigue una heurística para conservar las (una especie de) relaciones de orden entre los mismos objetos en los espacios de entrada y de salida.
- Se restringe la matriz de Mahalanobis al caso diagonal y se formula el problema como un problema de maximización de un margen blando de manera análoga a las ν-SVM.
- Se resuelve el problema de optimización cuadrático mediante una adaptación del algoritmo SMO Platt (1998); Platt et al. (1999) que garantiza la condición de positividad de la matriz diagonal.

No obstante, algunos aspectos de este algoritmo son claramente mejorables.

- Por un lado, el hecho de considerar todas las ternas posibles en la vecindad de cada punto, eleva el número total de restricciones a $|\mathcal{T}| = Nk(k-1)/2$, donde N es el número de datos de entrenamiento y k es el tamaño de la vecindad. Además de representar un problema en cuanto a carga computacional, en estas restricciones pueden aparecer redundancias que pueden influir en posibles sobreajustes de los modelos de aprendizaje.
- La solución del problema cuadrático, que se obtiene sobre el correspondiente dual, resulta computacionalmente costosa puesto que se debe manejar en cada iteración una matriz de Gram de tamaño $|\mathcal{T}|$.

El presente trabajo tiene como objetivo desarrollar una formulación alternativa eficiente de la idea del algoritmo DMLMTP. Para ello, se considerará el problema primal junto con métodos eficientes basados en el descenso por gradiente estocástico. Formulaciones similares han dado muy buenos resultados recientemente en problemas de optimización similares Nguyen et al. (2018); Wang et al. (2018); Xie and Li (2018). Además de la formulación y el método de optimización, se revisará la heurística para la selección de restricciones en forma de tripletas a partir de los datos de entrenamiento con la idea de obtener ventajas computacionales. Con estas mejoras, se obtiene un algoritmo con un funcionamiento similar al DMLMTP pero bastante más eficiente y con potencial para ser extendido a casos más generales.

2. Materiales y métodos

2.1. Definiciones y Notaciones

Para desarrollar nuestra propuesta se hace necesario introducir algunas notaciones y definiciones. Las matrices serán denotadas en mayúscula, mientras que los vectores se indican en minúscula con una flecha en la parte superior. La matriz diagonal $W \succeq 0$ expresa que es semi-definida positiva PSD. Sean, $\vec{x} \in \mathcal{X} \subseteq \mathbb{R}^p$ y $\vec{y} \in \mathcal{Y} \subseteq \mathbb{R}^q$ dos vectores cualesquiera en los espacios de entrada y salida, respectivamente. Para el conjunto de datos con N instancias de entrenamiento, $\mathcal{D} = \{(\vec{x}_1, \vec{y}_1), \ldots, (\vec{x}_N, \vec{y}_N)\}$, se puede aprender un predictor $h : \mathcal{X} \to \mathcal{Y}$ que estime, de manera simultanea, el conjunto de variables de salidas. Este tipo de problema es conocido como predicción con salidas múltiples. En nuestro caso la predicción, $h(\vec{x})$, se obtendrá como el promedio ponderado por la distancia a $\vec{x}$ de los valores de salida para los k vecinos más próximos. Es decir,

$$h(\vec{x}) = \frac{1}{S} \sum_{\vec{x}_i \in \mathcal{N}_k(\vec{x})} d_W(\vec{x}, \vec{x}_i)\vec{y}_i$$

donde $S = \sum_{\vec{x}_i \in \mathcal{N}_K(\vec{x})} d_W(\vec{x}, \vec{x}_i)$, y $\mathcal{N}_k()$ es el conjunto de los k vecinos más próximos según $d_W(.,.)$.

Para introducir las restricciones se considerarán relaciones entre determinadas ternas de elementos del conjunto de entrenamiento que designaremos como $\mathcal{T} \subseteq \{(i,j,\ell) \ : \ 1 \leq i,j,\ell \leq N\}$. Para cada elemento de $\mathcal{T}$ escribiremos $X_{ij\ell} = (\vec{x}_i, \vec{x}_j, \vec{x}_\ell)$ y $Y_{ij\ell} = (\vec{y}_i, \vec{y}_j, \vec{y}_\ell)$ para

referirnos a la terna de valores de entrada y salida, respectivamente. Las ternas en $\mathcal{T}$ deberán ser elegidas de manera que sean representativas y estén formadas por elementos relativamente cercanos en el espacio de entrada. El objetivo a exigir es que se cumpla la siguiente relación: si los elementos i y j están cerca en el espacio de entrada (comparados con i y ℓ), esto mismo debe pasar con sus valores de salida. De manera más formal,

$$d_W(\vec{x}_i, \vec{x}_j) < d_W(\vec{x}_i, \vec{x}_\ell) \Leftrightarrow d_\mathcal{Y}(\vec{y}_i, \vec{y}_j) < d_\mathcal{Y}(\vec{y}_i, \vec{y}_\ell), \tag{1}$$

donde $d_\mathcal{Y}(.,.)$ es una medida de distancia en el espacio de salida. En el presente trabajo se explorarán diferentes heurísticas para seleccionar conjuntos de ternas que darán lugar a comportamientos diferentes tanto en desempeño como en coste computacional. Por conveniencia, en este trabajo definiremos las cantidades $\Delta^\mathcal{Y}_{ij\ell} = d^2_\mathcal{Y}(\vec{y}_i, \vec{y}_\ell) - d^2_\mathcal{Y}(\vec{y}_i, \vec{y}_j)$ y $\Delta^W_{ij\ell} = d^2_W(\vec{x}_i, \vec{x}_\ell) - d^2_W(\vec{x}_i, \vec{x}_j)$, con lo que la condición anterior se expesará en base a la positividad de éstas, $\Delta^W_{ij\ell} > 0$ y $\Delta^\mathcal{Y}_{ij\ell} > 0$.

2.2. Formulación del problema

La nueva propuesta, denominada MLMOL (*Metric Learning for Multi-Output Learning*), presenta tres modificaciones esenciales con respecto al método DMLMTP Gonzalez et al. (2016). Por un lado se formula el problema al estilo de las C-SVM Vapnik (2013) y se resuelve el problema primal. Además, se utiliza el descenso por gradiente estocástico como método de optimización. Y por último, se utilizan diferentes heurísticas mejoradas para la selección de las ternas. De este modo el problema primal que define nuestra propuesta será:

$$\begin{aligned} \min_W \quad & \frac{1}{2}\|W\|^2_F + C \sum_{ij\ell \in \mathcal{T}} \ell(W, X_{ij\ell}, Y_{ij\ell}) \\ s.a. \quad & W \succeq 0 \end{aligned} \tag{2}$$

donde la función de pérdida para cada terna de $\mathcal{T}$ mide hasta que punto se incumple la condición de orden en (1) con la adición de un margen y que se define como

$$\ell(W, X_{ij\ell}, Y_{ij\ell}) = \text{máx}[0, 1 - \delta^y_{ij\ell}\Delta^W_{ij\ell})] \tag{3}$$

donde $\delta^y_{ij\ell} = \text{sgn}(\Delta^{\mathcal{Y}}_{ij\ell})$ y sgn() indica la función signo.

El problema de optimización, descrito en la ecuación (2), se puede escribir usando únicamente vectores de la siguiente manera

$$\begin{aligned} \min_{\vec{w}} \quad & \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{ij\ell\in\mathcal{T}} \text{máx}[0, 1 - \delta^y_{ij\ell}\vec{w}^T\vec{\phi}_{ij\ell}] \\ s.t. \quad & \vec{w} \geq \vec{0} \end{aligned} \tag{4}$$

donde se ha expresado el problema en términos de vector diagonal de W, $\vec{w} \in \mathbb{R}^p$, cuyas coordenadas deben ser positivas, $\vec{w} \geq \vec{0}$. Por otra parte, $\vec{\phi}_{ijk} \in \mathbb{R}^p$ determina un vector en el espacio de entrada definido como como $\vec{\phi}_{ij\ell} = (\vec{x}_i - \vec{x}_\ell) \circ (\vec{x}_i - \vec{x}_\ell) - (\vec{x}_i - \vec{x}_j) \circ (\vec{x}_i - \vec{x}_j)$, donde el operador $\circ$ representa el producto elemento a elemento o de Hadamard.

2.3. Solución computacional mediante gradiente estocástico.

El gradiente de la función criterio expresada en (4) se puede escribir como:

$$\nabla\mathcal{L} = \vec{w} + C\sum_{ij\ell\in\mathcal{T}} \text{máx}[0, 1 - \delta^y_{ij\ell}\vec{\phi}_{ij\ell}]$$

o si lo particularizamos para una sola terna como:

$$\nabla\mathcal{L}_{ij\ell} = \vec{w} + C\,\text{máx}[0, 1 - \delta^y_{ij\ell}\vec{\phi}_{ij\ell}] = \begin{cases} \vec{w} - C\vec{\phi}_{ij\ell} & \text{si} \quad \delta^y_{ij\ell} = -1 \quad \text{y} \quad \vec{w}^T\vec{\phi}_{ij\ell} > -1 \\ \vec{w} + C\vec{\phi}_{ij\ell} & \text{si} \quad \delta^y_{ij\ell} = 1 \quad \text{y} \quad \vec{w}^T\vec{\phi}_{ij\ell} < 1 \\ 0 & \text{otro caso} \end{cases}$$

donde se ve explícitamente que no todas las ternas de $\mathcal{T}$ dan lugar a contribuciones no nulas al gradiente. A partir de esto y por motivos de eficiencia se establece un criterio formado por pequeños subconjuntos aleatorios o *minibatches*, $\mathcal{B} \subseteq \mathcal{T}$, formados por ternas activas de manera que en cada iteración del algoritmo se actualiza la solución con el promedio de las contribuciones de cada terna de $\mathcal{B}$. La actualización de la solución en cada iteración una vez añadida la restricción de positividad se escribiría como

$$\vec{w}_{t+1} = \text{máx}[\vec{0}, \vec{w}_t - \alpha_t \frac{1}{|\mathcal{B}|} \sum_{ij\ell \in \mathcal{B}}^{\mathcal{B}} \nabla \mathcal{L}_{ij\ell}] \tag{5}$$

donde α_t es la tasa de aprendizaje y el operador máx[.,.] actúa coordenada a coordenada sobre los vectores a los que se aplica.

2.4. Heurística para definir el conjunto de ternas.

En el algoritmo DMLMTP propuesto en Gonzalez et al. (2016) se define una heurística local, H_0, para la selección de las ternas en $\mathcal{T}$ que consiste, para todo $\vec{x}_i$, en considerar todos los pares dentro de su k-vecindad, $\mathcal{N}_k(\vec{x}_i)$. En particular,

$$\mathcal{T}_{H_0} = \left\{(i,j,\ell) \ : \ \vec{x}_j, \vec{x}_\ell \in \mathcal{N}_K(\vec{x}_i) \quad \text{y} \quad \Delta^{\mathcal{Y}}_{ij\ell} \geq 0\right\} \tag{6}$$

Para reducir el número de restricciones consideradas a lo largo del proceso, y al mismo tiempo, reducir la posible redundancia de información que se introduce, se propone modificar la heurística anterior por otras de manera que el conjunto de ternas se defina con el objeto $\vec{x}_\ell$ como el primer elemento de la vecindad de $\vec{x}_i$, a la que nos referiremos como H_1, o bien, que el objeto $\vec{x}_j$ sea el último de la vecindad y que designaremos como H_2. Más concretamente,

$$\mathcal{T}_{H_1} = \left\{ (i,j,\ell) \ : \ \vec{x}_\ell = \underset{\vec{x}\in\mathcal{N}_K(\vec{x}_i)}{\operatorname{argmin}} (d(\vec{x},\vec{x}_i)), \quad \vec{x}_j \in \mathcal{N}_K(\vec{x}_i) - \{\vec{x}_\ell\} \quad \text{y} \quad \Delta^{\mathcal{Y}}_{ij\ell} \geq 0 \right\} \tag{7}$$

para la variante del primer vecino (H_1) y

$$\mathcal{T}_{H_2} = \left\{ (i,j,\ell) \ : \ \vec{x}_j = \underset{\vec{x}\in\mathcal{N}_K(\vec{x}_i)}{\operatorname{argmax}} (d(\vec{x},\vec{x}_i)), \quad \vec{x}_\ell \in \mathcal{N}_K(\vec{x}_i) - \{\vec{x}_j\} \quad \text{y} \quad \Delta^{\mathcal{Y}}_{ij\ell} \geq 0 \right\} \tag{8}$$

para el último de los vecinos (H_2).

3. Resultados y discusión

En primer lugar, se diseñó un experimento para estudiar el comportamiento de las heurísticas H_1 y H_2 respecto a la empleada en el algoritmo DMLMTP, H_0. Recordemos que en el caso de H_0 se emplean todos los pares en la vecindad, $Nk(k-1)/2$, mientras que en las restantes dos heurísticas se emplean solo $(K-1)$. Adicionalmente a la heurística, se decidió estudiar la influencia de varios métodos de optimización estocásticos en la biblioteca de código SGDLibrary Kasai (2017). Se ejecutaron 4 variantes de algoritmos de optimización estocásticos. El primero de ellos, el clásico gradiente estocástico SGD con *minibatches*, el segundo y el cuarto son dos variantes de LBFGS uno de los cuales con memoria limitada y el tercer algoritmo es una variante Quasi-Newton. En la experimentación se simularon 6 conjuntos de datos sintéticos en el contexto de la predicción con salidas múltiples. Las bases de datos fueron generadas con 20 variables de entrada independientes distribuidas aleatoriamente y 5 variables de salida generadas a partir de una matriz de regresión para la cual controlamos los niveles de dependencia. Como medida de error para la experimentación se empleó el aRRMSE similar a los trabajos Spyromitros-Xioufis et al. (2016); Gonzalez et al. (2016). El número de casos en la base de datos fue variando en el rango $N = 200, 500, 1000$ mientras que para el análisis local del algoritmo se tomaron como valores de la vecindad $k = 5, 10$. La tabla 1 muestra los resultados de los diferentes métodos de optimización ejecutados sobre los conjuntos de datos

sintéticos.

Tabla 1. Resultados sobre conjuntos de datos sintéticos.

	Heurística H_0				**Heurística H_1**				**Heurística H_2**			
Data	**SGD**	**SVRG**	**SQN**	**oBFGS**	**SGD**	**SVRG**	**SQN**	**oBFGS**	**SGD**	**SVRG**	**SQN**	**oBFGS**
Data1 $N = 200$, $k = 5$	1.026	1.084	1.070	1.079	1.033	1.060	1.076	1.061	**1.011**	1.052	1.046	1.047
Data2 $N = 500$, $k = 5$	1.033	1.059	1.085	1.052	**1.025**	1.071	1.094	1.069	**1.025**	1.073	1.063	1.062
Data3 $N = 1000$, $k = 5$	1.038	1.056	1.076	1.042	1.036	1.086	1.079	1.084	**1.021**	1.048	1.057	1.059
Data4 $N = 200$, $k = 10$	**1.001**	1.063	1.033	1.103	1.007	1.074	1.073	1.060	1.002	1.009	1.005	1.013
Data5 $N = 500$, $k = 10$	1.028	1.064	1.054	1.075	1.020	1.073	1.077	1.067	1.031	1.030	1.035	**1.024**
Data6 $N = 1000$, $k = 10$	1.037	1.074	1.062	1.065	**1.034**	1.042	1.039	1.041	1.037	1.045	1.046	1.045

Si analizamos los resultados desde el punto de vista de las heurísticas estudiadas, apreciamos que la variante H_2 muestra los mejores resultados lo que nos indica que es más representativo tomar como pivote el elemento más lejano de la vecindad. De igual manera, el hecho de que H_2 muestre ventajas sobre la variante H_0 está relacionada con la redundancia de información que introduce esta heurística lo cual puede afectar al ajuste del modelo. En tal sentido, consideramos que la reducción de datos para el proceso de entrenamiento tiene consecuencias similares a las de la función de regularización. Con relación a los métodos de optimización, se observa un funcionamiento estable para el algoritmo SGD con *minibatches* la cual es una variante sencilla y fácilmente escalable a ambientes distribuidos dentro de los métodos basados en el gradiente estocástico. Adicionalmente, la aplicación de la prueba de Friedman con la corrección de Shaffer para el análisis a nivel de heurística, nos corrobora los resultados estables de H_2 con diferencias significativas respecto a las restantes heurísticas, como se puede apreciar en el gráfico del rayo numérico de los ranking promedio en la figura 1. De igual manera, la figura 2 esboza como se comportaron los diferentes métodos de optimización evaluados luego de aplicar la prueba no paramétrica de Friedman con la corrección de Shaffer. En esta gráfica se aprecian los resultados superiores del SGD, con diferencias estadísticamente significativas respecto al resto de los métodos, para los conjuntos de datos estudiados.

Luego de estudiar los métodos de optimización y las heurísticas sobre conjuntos de datos generados sintéticamente, desarrollamos una segunda evaluación sobre 12 conjuntos de datos reales disponibles para problemas de predicción con salidas múltiples. Para ello comparamos el algoritmo DMLMTP con las variantes MLMOL (empleando la heurística H_0) y MLMOL* (con la heurística H_2). Se ejecutó un procedimiento de validación cruzada en la experimentación con 10 *folds*. En el caso del algoritmo SGD empleado en la experimentación se definió un

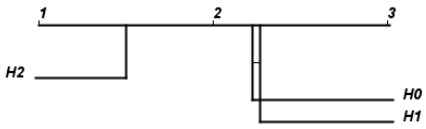

Figura 1. Ranking promedios para las diferentes heurísticas estudiadas tomando en cuenta todos los algoritmos.

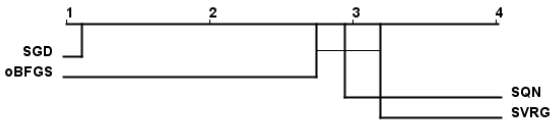

Figura 2. Evaluación estadística de los métodos de optimización para todas las heurísticas.

tamaño fijo de *minibatch* con 20 instancias y 5 repeticiones o épocas. La medida de error, al ejecutar cada algoritmo sobre los conjuntos de datos reales se sintetizan en la tabla 2, resaltando en negrita los mejores resultados.

Tabla 2. Medida de error sobre bases de datos reales.

dataset	DMLMTP	MLMOL	MLMOL*
edm	0.765	0.753	**0.729**
sf1	0.985	0.985	**0.983**
sf2	0.972	0.969	**0.968**
wq	**0.933**	0.935	0.937
oes10	**0.5**	0.504	0.504
oes97	0.602	**0.591**	0.597
atp1d	0.449	0.442	**0.432**
atp7d	**0.559**	0.574	0.592
jura	0.622	**0.621**	0.625
slump	0.696	**0.689**	0.718
enb	0.137	**0.134**	**0.134**
andro	0.555	0.564	**0.514**

Al ejecutar la prueba de Friedman sobre los resultados alcanzados, para las bases de datos reales, se obtiene que las variantes estudiadas, basadas en el gradiente estocástico y resueltos en el primal, muestran mayor estabilidad que el algoritmo DMLMTP, a pesar que estas diferencias no son estadísticamente significativas. En tal sentido, se logran mejores resultados en 9 de las bases de datos estudiadas para problemas de predicción con salidas múltiples.

Lo más significativo de los resultados obtenidos, es la mejora en eficiencia del método del descenso por gradiente estocástico respecto a la variante de SMO desarrollada en Gonzalez et al. (2016). Al mismo tiempo, las heurísticas tuvieron también un efecto en la eficiencia pero en menor medida. La figura 4 muestra los tiempos de cómputo, como promedio para los

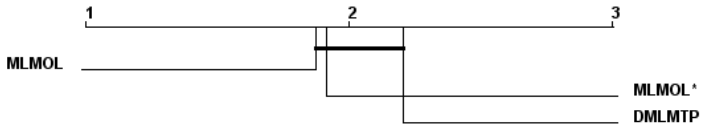

Figura 3. Prueba de Friedman sobre conjuntos de datos reales.

10 *folds*, durante la ejecución de cada algoritmo. Nótese que en la medida que cada modelo crece en el tamaño de la vecindad, el coste del algoritmo DMLMTP crece muy rápidamente mientras que las variantes del MLMOL se mantienen suficientemente estables. Este resultado nos sugiere la posibilidad de emplear el algoritmo MLMOL sobre volúmenes de datos a gran escala en el contexto de BigData. Por otra parte, los métodos de gradiente estocástico pueden ser implementados y ejecutados en ambientes distribuidos extendiendo la propuesta desarrollada en la presente investigación.

4. Conclusiones

En el presente trabajo se ha propuesto una variante más robusta y eficiente directamente relacionada con propuestas anteriores en el contexto de la regresión para salidas múltiples basada en distancias y vecinos. En particular, se ha reformulado el problema de optimización para obtener la métrica y con ello arribar a una solución computacional con un coste relativamente reducido y se han propuesto nuevas heurísticas para la selección de restricciones a partir de un conjunto de datos de entrenamiento que ofrece muy buenas perspectivas después de la experimentación realizada. Como trabajo futuro se propone la extensión del método para matrices semidefinidas positivas cualesquiera así como nuevas heurísticas que den lugar a una mayor aplicabilidad y robustez del método.

Referencias

Bellet, A., Habrard, A., and Sebban, M. (2014). A survey on metric learning for feature vectors and structured data. Technical report, Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA.

Bellet, A., Habrard, A., and Sebban, M. (2015). Metric learning. *Synthesis Lectures on*

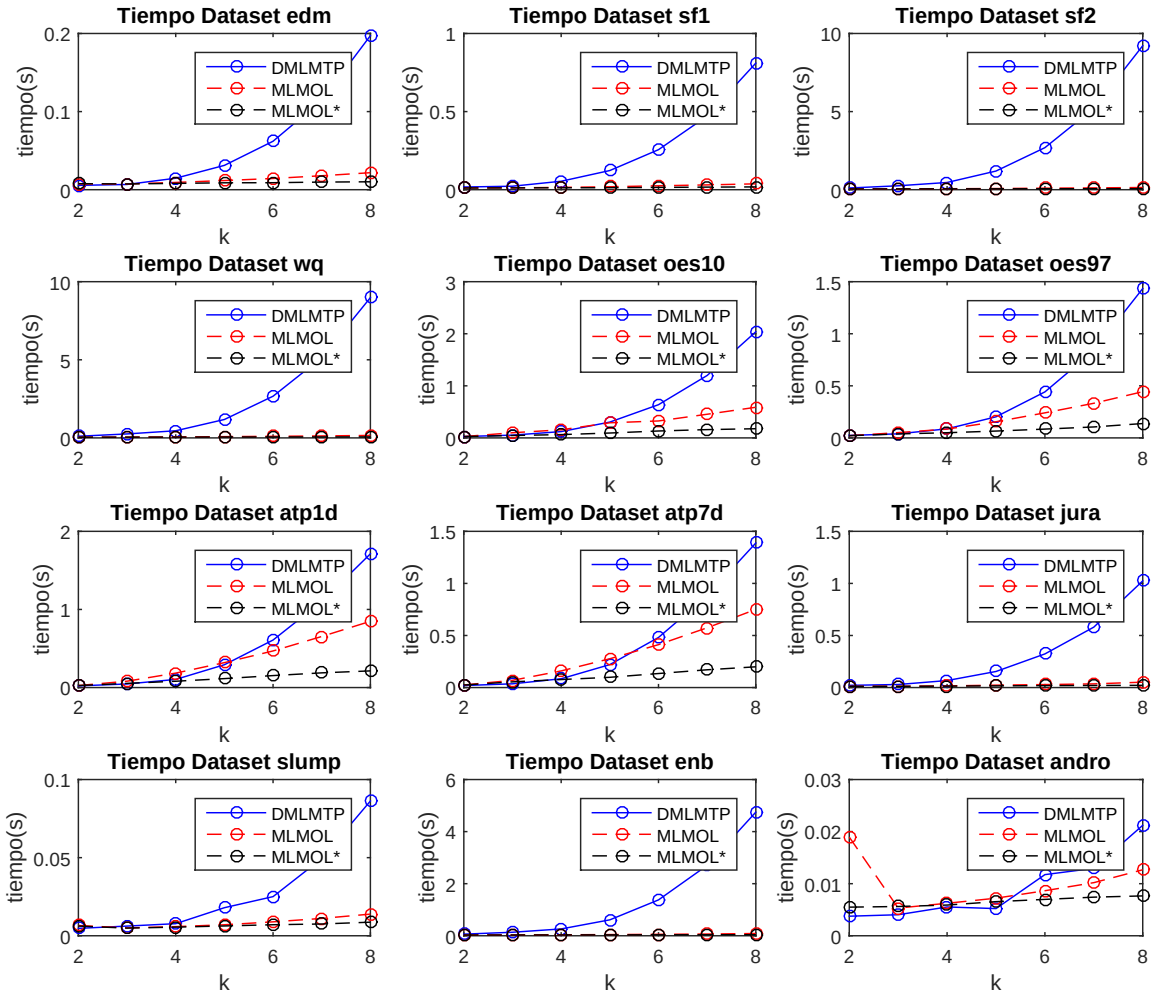

Figura 4. Tiempos de ejecución de los algoritmos evaluados sobre las bases de datos reales.

Artificial Intelligence and Machine Learning, 9(1):1–151.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM.

Gonzalez, H., Morell, C., and Ferri, F. J. (2016). Improving nearest neighbor based multi-target prediction through metric learning. In *Iberoamerican Congress on Pattern Recognition*, pages 368–376. Springer.

Gonzalez, H., Morell, C., and Ferri, F. J. (2018). Accelerated proximal gradient descent in metric learning for kernel regression. In *International Workshop on Artificial Intelligence and Pattern Recognition*, pages 219–227. Springer.

Kasai, H. (2017). Sgdlibrary: A matlab library for stochastic gradient descent algorithms. *arXiv preprint arXiv:1710.10951*.

Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2012). Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE.

Kulis, B. (2012). Metric learning: A survey. *Foundations & Trends in Machine Learning*, 5(4):287–364.

Li, D. and Tian, Y. (2018). Survey and experimental study on metric learning methods. *Neural Networks*.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Nguyen, B., Morell, C., and De Baets, B. (2016). Large-scale distance metric learning for k-nearest neighbors regression. *Neurocomputing*, 214:805–814.

Nguyen, B., Morell, C., and De Baets, B. (2018). Scalable large-margin distance metric learning using stochastic gradient descent. *IEEE transactions on cybernetics*.

Nguyen, N. and Guo, Y. (2008). Metric learning: A support vector approach. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases-Part II*, pages 125–136. Springer-Verlag.

Perez-Suay, A., Ferri, F. J., Arevalillo-Herráez, M., and Albert, J. V. (2013). Comparative evaluation of batch and online distance metric learning approaches based on margin

maximization. In *IEEE International Conference on Systems, Man, and Cybernetics, Manchester, SMC 2013, United Kingdom, October 13-16, 2013*, pages 3511–3515.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

Platt, J. et al. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods support vector learning*, 3.

Schultz, M. and Joachims, T. (2004). Learning a distance metric from relative comparisons. *Advances in neural information processing systems (NIPS)*, page 41.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. (2016). Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98.

Tarlow, D., Swersky, K., Charlin, L., Sutskever, I., and Zemel, R. (2013). Stochastic k-neighborhood selection for supervised and unsupervised learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 199–207.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Wang, F., Zuo, W., Zhang, L., Meng, D., and Zhang, D. (2015). A kernel classification framework for metric learning. *IEEE Trans. Neural Netw. Learning Syst.*, 26(9):1950–1962.

Wang, Z., Shao, Y.-H., Bai, L., Li, C.-N., Liu, L.-M., and Deng, N.-Y. (2018). Insensitive stochastic gradient twin support vector machines for large scale problems. *Information Sciences*, 462:114–131.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244.

Weinberger, K. Q. and Tesauro, G. (2007). Metric learning for kernel regression. In *Artificial Intelligence and Statistics*, pages 612–619.

Xie, Z. and Li, Y. (2018). Large-scale support vector regression with budgeted stochastic gradient descent. *International Journal of Machine Learning and Cybernetics*, pages 1–13.