GLOMERULAR FILTRATION RATE ESTIMATION BY A NOVEL BINNING-LESS ISOTONIC STATISTICAL BIVARIATE REGRESSION METHOD

SEBASTIAN GILES, SIMONE FIORI*

Abstract. Statistical bivariate regression is a method for finding a relationship between unpaired sets of data based on statistic distribution matching. In the present paper, an efficient numerical algorithm is proposed to perform bivariate regression. The algorithm is then applied to correlate glomerular filtration rate to serum creatinine concentration. Glomerular filtration rate is an important indicator of kidney function. As direct measurement is highly impractical, there is considerable interest in developing numerical algorithms to estimate glomerular filtration rate from parameters which are easier to obtain, such as demographic and 'bedside' assays data.

1. Introduction. Several real-world complex phenomena in virtually any branch of science lack accurate mathematical descriptions (see, e.g., [7, 19] for examples related to geophysics and geology). In these cases, it is not possible to predict the value of a variable by an analytically derived expression, therefore, empirical methods must be employed to develop a model of the observed process. The most common set of tools used to infer a functional relationship between variables is regression analysis (see, e.g., [12, 17]).

The problem that triggered the present research is the estimation of glomerular filtration rate (GFR), which is used in clinical nephrology as an indicator of kidney function and is relevant for assessing progression of renal disease. Glomerular filtration rate is also frequently required for evaluating optimal dosage for medications [8, 18]. However, determination of true GFR is time-consuming, costly and difficult to perform [4, 28]. Thus, there is considerable interest in developing models to estimate GFR using simpler parameters such as a patient's age, weight, height, gender and values which can be more conveniently measured as part of a standard blood test.

Traditional forms of regression are based on some parametrised function whose graph is made to lay reasonably close to the points making up the experimental dataset. Values for the parameters are typically calculated using the least squares method [10].

Isotonic regression allows greater freedom for the regression curve to fit data by constructing a piecewise linear function, described by a lookup table (LUT). Overfitting is avoided by requiring the function to be isotonic, which is reasonable whenever the underlying physical phenomenon is inherently monotonic (such as the relationship between percentage body fat and waist circumference [11]). Various algorithms can be used to find the LUT values-entries that satisfy a least squares condition [1, 16].

Statistical bivariate regression (SBR) provides an improvement over isotonic regression. Its advantages derive from the fact that it relies on finding the relationship between the statistical distributions of the variables. SBR is not based on a least squares method, therefore, it does not require data to be associated in ordered pairs. This makes it ideal for correlating two quantities that cannot be both measured on the same individual. The algorithm presented in [5] independently estimates the probability density functions (PDF) of the two variables by dividing the dataset ranges into bins and populating look-up tables for the relative frequencies. The PDFs are then integrated numerically to obtain the cumulative distribution functions (CDF) which are, or can be licitly adjusted to be, bijective and allow for the regression model to be obtained as the map between values with equal probabilities.

This report describes an alternative algorithm developed to make statistical bivariate regression more versatile and faster over large datasets by entirely avoiding the binning (previously required to estimated PDFs) and numerical integration operations (previously required to estimated a CDF from a PDF that

*S. Giles is with the School of Information and Automation Engineering, Università Politecnica delle Marche, Via Brecce Bianche, I-60131 Ancona (Italy).

S. Fiori is with Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Via Brecce Bianche, I-60131 Ancona (Italy).

This draft is dated November 9, 2017.

it is associated to).

The present paper is organized as follows. Section 2 recalls the notion of bivariate statistical regression and explains the main idea and the details about the proposed binning-less bivariate statistical regression algorithm. Section 3 explains an analysis of the glomerular-filtration-rate (GFR) estimation based on creatinine levels and illustrates such analysis by means of numerical tests performed on a dataset drawn from a study on pediatric patients in mainland China. Section 4 concludes the paper.

2. Binning-less statistical bivariate regression algorithm. Given random variables X and Y, for which we expect the existence of a monotonic function f such that $Y = f(X)$, let $D_X \in \mathbb{R}^n$ and $D_Y \in \mathbb{R}^m$ be arrays whose components are realizations of X and Y respectively.

The developed regression algorithm is encapsulated in a function that takes the D_X and D_Y dataset arrays along with an x for which the estimated value of $f(x)$ is returned.

Denoting by $P_X(x)$ and by $P_Y(y)$ the respective CDFs of X and Y, we recall from our previous works [5, 6] that

$$f(x) = P_Y^{-1}(P_X(x)) \text{ if } f \text{ is monotonically increasing,} \tag{2.1}$$

$$f(x) = P_Y^{-1}(1 - P_X(x)) \text{ if } f \text{ is monotonically decreasing.} \tag{2.2}$$

The proposed SBR regression procedure can be separated into two parts: the evaluation of a cumulative distribution function and the evaluation of an inverse cumulative distribution function. The former is handled by the CDF function defined in Algorithm 2, while the latter by INVCDF defined in Algorithm 3. Besides the argument for the CDF (or inverse CDF), both procedures require a dataset from which to infer the actual distributions. Algorithm 1 provides pseudocode for joining the two parts together in the case that a monotonically increasing regression model is sought. If the regression model is expected to be monotonically decreasing, then the value P on Line 3 should be replaced by its complement $1 - P$.

Algorithm 1 Statistical Bivariate Regression (for monotonically increasing models)

```
1: function STATISTICALREGRESSION($D_X$,$D_Y$, $x_q$)
2:     $P \leftarrow$ CDF($D_X, x_q$)          ▷ Evaluate CDF for value $x_q$ in dataset $D_X$
3:     $y_q \leftarrow$ INVCDF($D_Y, P$)       ▷ Evaluate inverse CDF for probability $P$ on dataset $D_Y$
4:     return $y_q$
5: end function
```

We developed a binning-less procedure to estimate the value of the CDF $P(q)$ of a generic random variable Q for which n realizations are stored as the components of an array D. The main idea to avoid binning is to estimate the cumulative distribution function of a dataset without resorting to an estimate of the probability density function first. This can be done by embracing the definition of CDF itself, which leads us to counting the number of realizations whose values are smaller than (or equal to) q and dividing this count by n. The solution shown in Algorithm 2 expands this idea to allow for a continuous, strictly monotonic interpolation for values that are not included in the original dataset. (Array indexing is 1-based and symbol $\wedge$ denotes logical conjunction.)

Here are a few comments about Algorithm 2:

- **Line 2**: The algorithm is notably simplified by sorting the entries of D in ascending order.
- **Lines 4–13**: This part is essentially a binary search for q in array D. The loop starts with indexes l and r as the endpoints of D and ends with l as the index of the last value less than q, and r as that of the first value greater than q. The only exception that may occur is handled in lines 14–16.
- **Line 8**: Making sure that $D[m] \neq D[l]$ is needed to prevent l and r from converging to 1 and 2 respectively, in the case that q is smaller than all elements in D.

Algorithm 2 Cumulative distribution function estimation

```
 1: function CDF($D$, $q$)
 2:     $D \leftarrow$ SORT($D$)                          ▷ Dataset is sorted in ascending order
 3:     $n \leftarrow$ LENGTH($D$)                        ▷ Gets the cardinality of the dataset $D$
 4:     $l \leftarrow 1$
 5:     $r \leftarrow n$
 6:     while $r - l > 1$ do
 7:         $m \leftarrow \lfloor (l + r)/2 \rfloor$
 8:         if $D[m] > q \wedge D[m] \neq D[l]$ then
 9:             $r \leftarrow m$
10:         else
11:             $l \leftarrow m$
12:         end if
13:     end while
14:     while $D[r] = D[l]$ do
15:         $l \leftarrow l - 1$
16:     end while
17:     while $r < n \wedge D[r] = D[r + 1]$ do
18:         $r \leftarrow r + 1$
19:     end while
20:     $d \leftarrow (q - D[l])/(D[r] - D[l])$
21:     $P \leftarrow (l + d \cdot (r - l))/n$
22:     if $P < 0$ then                                    ▷ Trimming out of range interpolations
23:         $P \leftarrow 0$
24:     else if $P > 1$ then
25:         $P \leftarrow 1$
26:     end if
27:     return $P$
28: end function
```

- **Lines 14–16**: This loop is needed to fix l in the case that it converges to $n - 1$ as a consequence of q being greater than all values in D.
- **Lines 17–19**: The previous operations already guarantee that l is the last index for the value $D[l]$, which means that there are l elements in D which are less than or equal to $D[l]$. This loop finds r, the count of elements whose values are smaller than (or equal to) the value $D[r]$.
- **Lines 20–21**: The probability $P(D[l])$ is estimated by the ratio l/n, whilst the probability $P(D[r])$ is estimated by the ratio r/n. The sought probability $P(q)$ is obtained via linear interpolation of the two.
- **Lines 22–26**: These checks fix the CDF for values of the query point q that lay outside the range of D.

The evaluation of the inverse CDF is based on the same principle applied in a reverse fashion: the input argument is a probability, it is multiplied by the cardinality n of the dataset D and rounded to an integer r. If the dataset D is sorted, then there will be r values less than or equal to $D[r]$. The Algorithm 3 again allows for a continuous, strictly monotonic interpolation to yield values that are not included in the original dataset.

Here are a few comments about Algorithm 3:

- **Line 2**: The algorithm is notably simplified by sorting D in ascending order.
- **Line 4**: The input probability is denormalized into the range $[0, n]$.

Algorithm 3 Inverse cumulative distribution function estimation

```
1:  function INVCDF($D$,$P_q$)
2:      $D \leftarrow$ SORT($D$)
3:      $n \leftarrow$ LENGTH($D$)
4:      $p \leftarrow P_q \cdot n$
5:      $r \leftarrow \lceil p \rceil$
6:      if $r = 0$ then
7:          $r \leftarrow 1$
8:      end if
9:      $l \leftarrow r$
10:     while $r < n \land D[r] = D[r+1]$ do
11:         $r \leftarrow r + 1$
12:     end while
13:     while $l > 1 \land D[l-1] = D[l]$ do
14:         $l \leftarrow l - 1$
15:     end while
16:     if $l = 1$ then
17:         $l \leftarrow r$
18:         $r \leftarrow r + 1$
19:         while $r < n \land D[r] = D[r+1]$ do
20:             $r \leftarrow r + 1$
21:         end while
22:     else
23:         $l \leftarrow l - 1$
24:     end if
25:     $d \leftarrow (p - l)/(r - l)$
26:     $q \leftarrow D[l] + d \cdot (D[r] - D[l])$
27:     return $q$
28: end function
```

- **Lines 5–9**: The indexes r and l take the value of p rounded to the next integer to be used as an index, which also requires r and l to be non-zero.
- **Lines 10–12**: The index r is made to point to the last occurrence of the smallest value whose CDF is greater than the query probability.
- **Lines 13–15, 20**: l is made to point to the first occurrence of the same value.
- **Line 23**: Normally l is then made to point to the previous element which is the last occurrence of the greatest value whose CDF is less than the query probability.
- **Lines 16–21**: If l is already pointing to the first value of the array then l takes the value of r and r is made to point to the last occurrence of the following value.
- **Lines 25–26**: The inverse cumulative distribution function value $P^{-1}(l/n)$ is estimated by $D[l]$, whilst the inverse cumulative distribution function value $P^{-1}(r/n)$ is estimated by $D[r]$. Nearby values are obtained via linear interpolation.

MATLAB codes to implement the three functions described in the present section are reported in the Appendix A, along with some comments on their computational burden.

Ideally, it should hold that INVCDF(CDF(q)) $= q$. To verify this identity, we ran a numerical test on the Algorithms 2 and 3 using the real-world dataset analyzed in Section 3. For each element q in the array, we calculated the relative deviation from identity

$$\delta = \left| \frac{\textsc{invCdf}(\textsc{cdf}(q)) - q}{q} \right|. \tag{2.3}$$

The largest value of δ was found to be in the order of 10^{-15}, which is definitely acceptable.

To conclude this section of the paper, let us consider a minimal working numerical example. Let $D_X = [2, 3, 5, 5, 6, 6, 7, 9]$ and $D_Y = [6, 7, 10, 10, 11, 11, 11, 12, 15, 20]$, i.e., $n = 8$, $m = 10$. The actual underlying model is $f(x) = 2x + 2$. Note that the data are not paired. Both arrays are already sorted for simplicity. Let the query point be $x_q = 4$: we expect the result of statistical regression (Algorithm 1) to be $y_q \approx 10$. The proposed procedure prescribes first to estimate the probability $P_X(4)$: the Algorithm 2 will find $l = 2$ and $r = 4$. The Algorithm 2 then will compute the linear interpolation:

$$d = \tfrac{x_q - X[l]}{X[r] - X[l]} = \tfrac{4-3}{5-3} = 0.5 \quad \Rightarrow \quad P_X(4) = \tfrac{l + d\cdot(r-l)}{n} = \tfrac{2+0.5\cdot(4-2)}{8} = 0.375.$$

The proposed procedure prescribes next to estimate the inverse probability $P_Y^{-1}(0.375)$: the Algorithm 3 will find $p = 0.375 \cdot m = 3.75$, $r = 4$ and $l = 2$. By linear interpolation, the Algorithm 3 then finds

$$d = \tfrac{p-l}{r-l} = \tfrac{3.75-2}{4-2} = 0.875 \Rightarrow P_Y^{-1}(0.375) = D[l] + d \cdot (D[r] - D[l]) = 7 + 0.875 \cdot (10 - 7) = 9.625.$$

3. Application to glomerular filtration rate estimation.

Chronic kidney disease is a recognized public health problem. Chronic kidney disease is classified into stages according to the level of glomerular filtration rate, and stage-specific action plans facilitate the evaluation and the management of chronic kidney disease. Glomerular filtration rate can be estimated by means of empirical formulas that incorporate blood serum creatinine concentration, blood serum cystatin-C concentration as well as demographic and clinical variables such as age, gender, race, and body size. Glomerular filtration rate estimating formulas provide a more accurate assessment of the level of kidney function than biomarkers concentrations alone.

Measuring the glomerular filtration rate is crucial for determining appropriate drug dosing, monitoring the effects of therapeutic interventions, and following the progression of chronic kidney disease. For instance, in pediatric autologous hematopoietic stem cell transplantation treatment protocols, chemotherapy dosing is commonly based on renal function, as patients with a reduced GFR levels receive reduced dosages, which can affect toxicity profiles and therapeutic benefit [13].

3.1. Acronyms, formulas and references.

Commonly used formulas to estimate the GFR are based on blood serum creatinine concentration levels and on demographic/clinical data. The level of serum creatinine concentration may be measured by two different tecniques: the Jaffe method [27] and the isotope dilution mass spectrometry (IDMS) enzymatic method [29]. The most frequently used formulas to estimate GFR are

- **MDRD Study formula**: Modification of Diet in Renal Disease. It is used only for chronic kidney disease, as it was found to be inaccurate for acute renal failure. MDRD may underestimate the actual glomerular filtration rate in healthy patients [14, 22]. The performance of the Modification of Diet in Renal Disease Study formula varies substantially among populations, because of differences among studies in the range of GFR, methods for GFR measurement, and methods for creatinine assays in blood plasma. The MDRD 4-variable formula reads

$$GFR = 186 \cdot sCr^{-1.154} \cdot age^{-0.203} \cdot [1.2010 \text{ if Black}] \cdot [0.742 \text{ if Female}]. \tag{3.1}$$

- **CKD-EPI**: Chronic Kidney Disease Epidemiology Collaboration. The CKD-EPI formula is based on the same four variables as the MDRD Study formula, but it resulted from a different technique to model the relationship between estimated GFR and blood serum creatinine concentration, age, gender and race. This formula was reported to perform better and with less bias than the MDRD Study formula, especially in patients with higher GFR. This results in reduced misclassification of chronic kidney disease [15]. The CKD-EPI formula reads

$$GFR = 141 \cdot \min\left(\tfrac{sCr}{k}, 1\right)^{a} \cdot \max\left(\tfrac{sCr}{k}, 1\right)^{-1.209} \cdot 0.993^{age} \cdot [1.159 \text{ if Black}] \cdot [1.018 \text{ if Female}], \tag{3.2}$$

where the constant k equals 0.7 for females and 0.9 for males, while the constant a equals 0.329 for females and 0.411 for males.

- **Mayo Quadratic formula**: The Mayo Clinic Quadratic equation attempts to estimate GFR from variables including serum creatinine concentration, age and gender. This formula appears to have better performance characteristics when used in patients with preserved renal function [21, 22]. The Mayo Quadratic formula reads

$$GFR = \exp\left(1.911 + \frac{5.249}{sCr} - \frac{2.114}{sCr^2} - 0.00686 \cdot age - [0.205 \text{ if Female}]\right). \tag{3.3}$$

  If the sCr level is less than 0.8 mg/dL, it is recommended to use the value 0.8 mg/dL for sCr.
- **Schwartz2009**: Updated Schwartz formula, also referred to as bedside Schwartz formula. It is one of several formulas to estimate GFR in pediatric patients, like the Counahan-Barratt formula based on blood serum creatinine concentration [3], and the Grubb formula based on blood serum cystatin-C concentration [26]. In most cases, the bedside Schwartz formula allows rapid and reasonably accurate estimation of GFR for clinical use in children with chronic kidney disease. The updated Schwartz formula reads

$$GFR = 41.3 \cdot \frac{height}{sCr}, \tag{3.4}$$

  where the serum creatinine concentration refers specifically to the values measured by the IDMS enzymatic method. The updated Schwarz formula is a standardized version of the original Schwartz formula $GFR = \sigma \cdot \frac{height}{sCr}$, where the serum creatinine concentration refers specifically to the values measured by the Jaffe method, and the constant σ depends on muscle mass, which varies with a child's age, and ranges in $33 - 55$.

In the above formulas and throughout this paper, measurement units of the GFR and the sCr values are mL/min/1.73m^2 and mg/dL respectively, while patients *height* is expressed in meters and patients *age* is expressed in years.

When measurement and calibration is more broadly available, glomerular filtration rate estimates using cystatin C may also exhibit broad clinical utility. Commonly used formulas based on blood serum cystatin C concentration levels and demographic/clinical variables are:

- **CKiD**: Chronic Kidney Disease in Children. A primary goal of the CKiD study was to develop a formula to estimate GFR using demographic variables and endogenous biochemical markers of renal function. The CKiD formula combines values of blood serum concentration of cystatin C (cysC), blood serum creatinine concentration and blood urea nitrogen concentration (BUN). The formula reads

$$GFR = 39.8 \cdot \left(\frac{heigth}{sCr}\right)^{0.456} \cdot \left(\frac{1.8}{cysC}\right)^{0.418} \cdot \left(\frac{30}{BUN}\right)^{0.079} \cdot \left(\frac{heigth}{1.4}\right)^{0.179} \cdot [1.076 \text{ if Male}]. \tag{3.5}$$

  It may be useful as a confirmatory test in specific circumstances when estimation of GFR based on serum creatinine is less accurate, or when the clinical scenario calls for a second test[9, 24].
- **Filler formula**: The empirical Filler formula to estimate the glomerular filtration rate reads

$$GFR = 91.62 \cdot (cysC)^{-1.123}. \tag{3.6}$$

It is one among several look-alike formulas as the Zappitelli formula $GFR = 75.94 \cdot (cysC)^{-1.17}$, the Larsson formula $GFR = 77.24 \cdot (cysC)^{-1.2623}$, the Hoek formula $GFR = 80.35 \cdot (cysC)^{-1} - 4.32$, the Rule formula $GFR = 66.8 \cdot (cysC)^{-1.30}$ and the Le Bricon formula $GFR = 78 \cdot (cysC)^{-1} + 4$ [13].

In the above formulas, the measurement unit of the cysC values is mg/L, while the measurement unit of the BUN values is mg/dL. The level of cystatin C is measured through a particle-enhanced nephelometric assay.

3.2. Experimental results on statistical regression. Existing multivariate formulas for GFR estimation have been compared and validated in [30] over a dataset of 87 Chinese children and adolescents aged 1 through 18. The authors of the research have included their dataset with the publication. For each patient, the available data comprise age, gender, physical parameters (such as height and weight), GFR (measured using double-sample plasma clearance [25]), two values for serum creatinine concentration as well as cystatine-C and blood urea nitrogen concentration. The two values of serum creatinine concentration correspond to two different measurement techniques, namely, the Jaffe method and the IDMS enzymatic method.

The study [30] compared four different formulas, namely, the original Schwartz formula, the updated Schwartz formula, the Filler formula and the CKiD formula. The study found the most effective estimation formula to be the updated Schwartz one.

Over said data, we also computed estimations using the other three widely employed formulas, namely MDRD, CKD-EPI and Mayo Quadratic, and compared them with the results of the updated Schwartz estimation formula. From the Figure 3.1, it is clear that the updated Schwartz formula outperforms all of the other functions.

In order to apply the statistical bivariate regression algorithm, we first assessed the existence of a single dominant variable. This was clearly found to be the serum creatinine concentration. Other variables used to estimate the glomerular filtration rate are age and height, whose effect however is marginal, as the scatter plots in Figure 3.2 reveal no strong statistical features. Quantitatively, this analysis is confirmed by the sample correlation coefficient [20], which, for a generic paired dataset (z_i, y_i) for $i = 1, 2, \ldots, n$ is calculated as

$$\rho_{y,z} = \frac{\sum_{i=1}^{n} (z_i - \bar{z})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (z_i - \bar{z})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}, \tag{3.7}$$

where $\bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$. The coefficient of correlation ρ takes values in the interval $[-1, 1]$. If variables are directly correlated, then we expect the coefficient to approach $+1$ while, if variables are inversely correlated, we expect a value of the coefficient of correlation close to -1. Unrelated variables yield a value of ρ close to 0. Table 3.1 shows correlation coefficients between the GFR and each of age, height and sCr for the whole population and the gender-defined subsets. The results illustrated in the table confirm the weak statistical correlation between glomerular filtration rate and age, as well as between glomerular filtration rate and height, especially for female children.

Gender	$\rho_{GFR,sCr}$	$\rho_{GFR,age}$	$\rho_{GFR,height}$
Males	-0.7051	-0.1375	-0.0910
Females	-0.7249	$+0.0801$	$+0.0548$
Both	-0.6744	-0.0565	-0.0425

Table 3.1. Sample correlation coefficients between glomerular filtration rate and age, height and creatinine concentration levels. Serum creatinine concentration refers to the IMDS-traceable values.

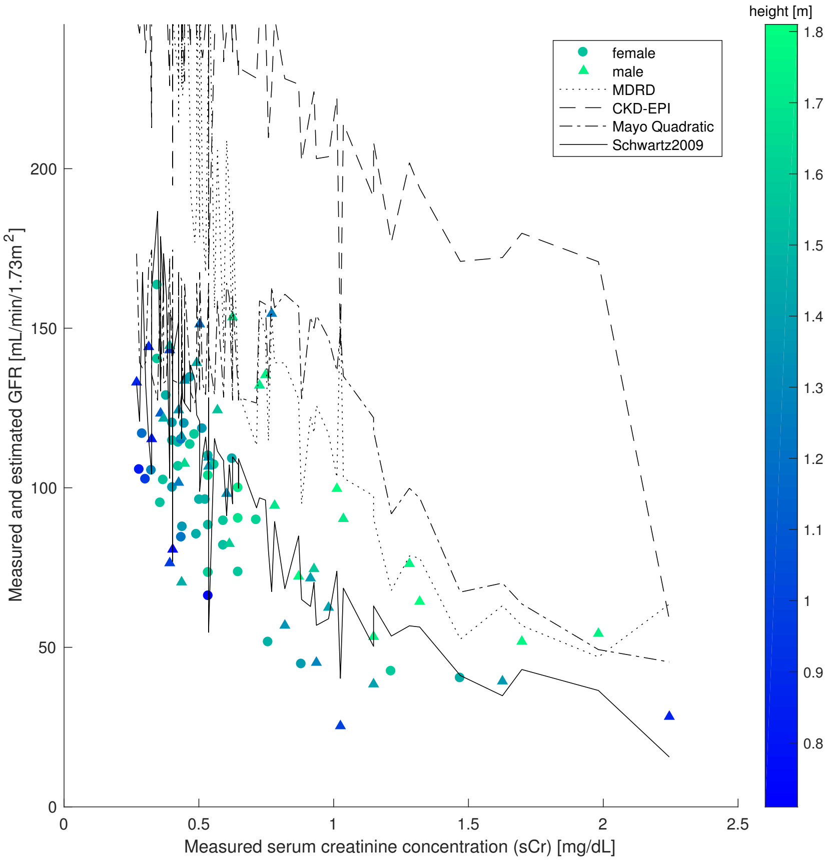

Fig. 3.1: Comparison of predictions made using different equations (MDRD, CKD-EPI, Mayo Quadratic and updated Schwartz). The serum creatinine concentration refers to the IMDS-traceable values.

From the Figure 3.1, it is readily appreciated that the GFR–sCr relationship presents a monotonically decreasing trend, which enables us to apply the SBR regression algorithm presented in the Section 2. According to the observations drawn about the performances of the closed-form models, we did not compare the SBR regression algorithm with the MDRD, the CKD-EPI and the Mayo Quadratic formulas.

As SBR generates a bivariate regression curve, for the sake of the comparison, a simplified version of the updated Schwartz formula was introduced to be independent of height. This was done by replacing the variable height with a constant equal to the mean height of all individuals in the dataset. This model is illustrated in the Figure 3.3, along with the datapoints, the regression curve obtained by SBR using the numerical-algebraic neural system (NANS) method explained in [5], the original updated Schwartz formula and the regression curve obtained using the binning-less method described in Section 2.

The input-output nature of bivariate regression grants the use of functional notation (i.e., given a value for the independent variable x, the prediction made for the value of the dependent variable y can be expressed as $y = f(x)$). The notation is commonly used in reference to closed-form models and will be adopted in this paper to also indicate predictions made using the Algorithm 1 and by interpolating

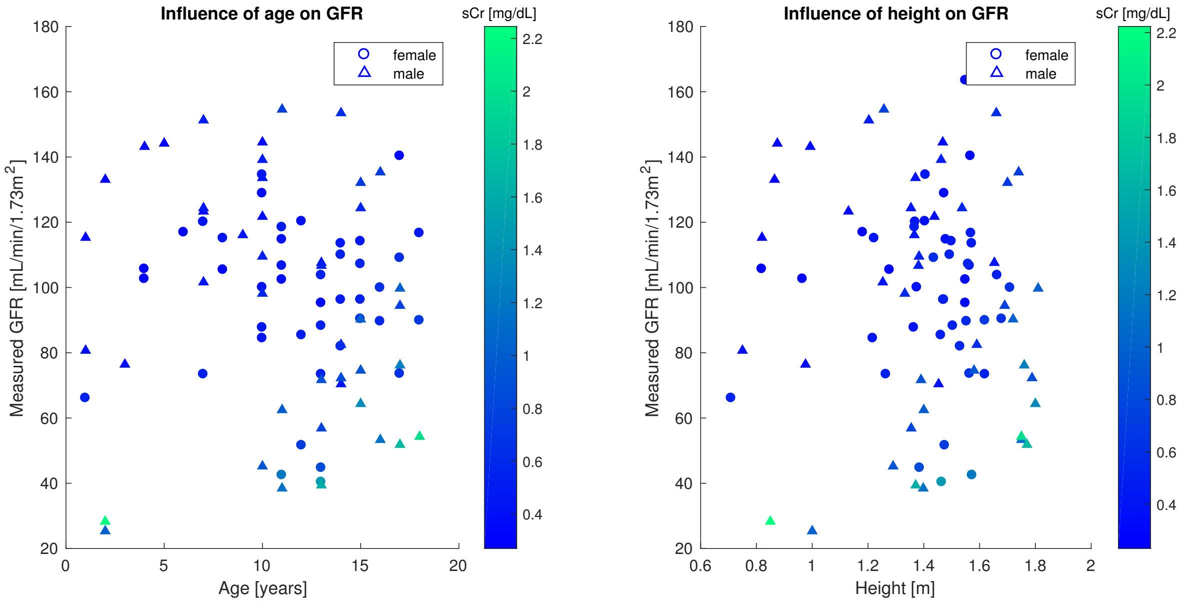

Fig. 3.2: Scatter plots of the glomerular filtration rate versus patients age or height suggest weak correlation. Serum creatinine concentration refers to the IMDS-traceable values.

the curve obtained using the NANS method.

The two closed-form models and the two numerical regression algorithms displayed in the Figure 3.3 were compared on prediction performance using three indexes: mean squared error (MSE), mean absolute error (MAE) and coefficient of determination (R^2). For a generic prediction-making system, represented by the function $f(x)$, being validated over the set of datapoints $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, these indexes are defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2, \tag{3.8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i|, \tag{3.9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{\sum_{i=1}^{n}(f(x_i) - \bar{y})^2}, \text{ where } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{3.10}$$

The reference values for a well-performing algorithm are $MSE \geq 0$ and $MAE \geq 0$ as close as possible to zero and R^2 as close as possible to 1[†]. In the present context, each x_i represents an instance of serum creatinine concentration sCr, each y_i represents an instance of glomerular filtration rate GFR, and $n = 87$. Comparisons were also made to evaluate the generalization ability of the closed-form model (simplified

[†] For a linear model, the coefficient R^2 ranges in [0, 1]: when $R^2 = 1$, it is said that a model explains the data perfectly. For a non-linear model, the coefficient of determination may take negative values. A negative R^2 means that the sample average explains the data better than the model.

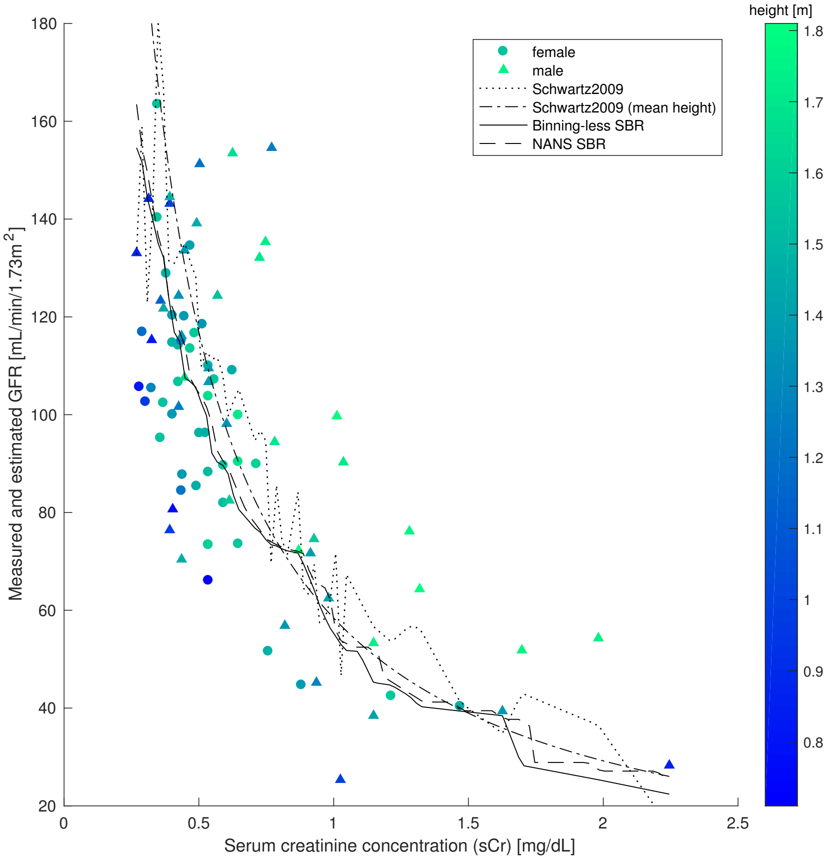

Fig. 3.3: Data set with overlaid regression and estimation curves. Comparison of the updated Schwartz equation, the simplified Schwartz equation, the Numerical-Algebraic Neural System (NANS) method introduced in [5] and the proposed method. Serum creatinine concentration refers to the IDMS-traceable values.

Schwartz) as well as of the considered numerical regression algorithms (SBR and NANS-SBR). This was achieved by measuring the “roughness” of the the regression curves through the index G defined by

$$G = \sum_{i=3}^{N} \frac{(f_i - 2f_{i-1} + f_{i-2})^2}{N-2}, \tag{3.11}$$

on the basis of the second-order differences of a sequence f_i. By definition, the index G increases with sharp changes in slope. The reference value for a well-performing algorithm is $G \geq 0$ as close as possible to zero. To be useful, the f_i values have to be sorted in some significant manner: in the present context, for each model $f(x)$ to be evaluated, f_i assumes the predictions at $N = 100$ equally spaced, increasing, values of serum creatinine concentration, namely:

$$f_i = f\left(x_{\min} + (i-1)\frac{x_{\max} - x_{\min}}{N-1}\right), \text{ for } i = 1, \ \ldots, \ N, \tag{3.12}$$

where $x_{\min}$ and $x_{\max}$ are respectively the smallest and largest measured creatinine concentration levels. The same index cannot be applied to multivariate functions, therefore the updated Schwartz equation was not tested with this criterion. An index similar to G was discussed in [2] to prevent overfitting of a neural-network model. The value of G is expected to be large for irregular curves and indeed it is close to zero for the simplified Schwartz model (independent of height), which is essentially a hyperbola, graph of a smooth function.

Models and formulas	G	MSE	MAE	R^2
Updated Schwartz formula	—	863.92	23.19	+0.1491
Simplified Schwartz formula	0.36	1341.80	27.84	−0.3216
NANS SBR model	1.83	696.62	20.32	+0.3139
Binning-less SBR model	1.65	674.90	20.19	+0.3353

Table 3.2. Generalization/roughness index (G), mean squared error (MSE), mean absolute error (MAE) and coefficient of determination (R^2) for the four considered estimation models (updated Schwartz formula, simplified Schwartz formula independent of height, numerical-algebraic neural-system based statistical bivariate regression algorithm, and proposed statistical bivariate regression algorithm).

The results of comparison are summarized in the table 3.2. Among the four methods considered, the binning-less statistical bivariate regression algorithm exhibits the lowest MSE and MAE values, that shows how SBR is very effective at fitting data. Taking the G-value of the regression line provided by the simplified Schwartz formula as a baseline, the binning-less algorithm exhibits a closer value to such baseline than the NANS SBR: this result shows that the novel SBR algorithm returns a smoother regression line compared to the NANS SBR one.

4. Conclusions. The aim of the present paper is to discuss the statistical bivariate regression method and to provide an improved algorithm which does not rely on binning for the steps that require estimation of the cumulative distribution functions.

The proposed algorithm was compared to the original statistical bivariate regression algorithm based on numerical-algebraic neural systems in the application to a real-world dataset. The application that motivated the present research is the estimation of an index of kidney function, the glomerular filtration rate, on the basis of regression by the creatinine concentration level in blood plasma. The comparison proved an all-round improvement in the new method, which, aside from being more efficient, yields a closer fit and a smoother regression curve.

REFERENCES

[1] M.J. Best and N. Chakravarti, "Active set algorithms for isotonic regression; a unifying framework", *Mathematical Programming*, Vol. 47, pp 425 – 439, 1990

[2] C.M. Bishop, "Training with noise is equivalent to Tikhonov regularization", *Neural Computation*, Vol. 7, No. 1, pp. 108 – 116, 1995

[3] R. Counahan, C. Chantler, S. Ghazali, B. Kirkwood, F. Rose and T.M. Barratt, "Estimation of glomerular filtration rate from plasma creatinine concentration in children", *Archives of Disease in Childhood*, Vol. 51, pp. 875 – 878, 1976

[4] Z.H. Endre, J.W. Pickering and R.J. Walker, "Clearance and beyond: the complementary roles of GFR measurement and injury biomarkers in acute kidney injury (AKI)", *American Journal of Physiology – Renal Physiology*, Vol. 301, No. 4, pp. F697 – F707, 2011

[5] S. Fiori, “Fast statistical regression in presence of a dominant independent variable”, *Neural Computing and Applications*, Vol. 22, No. 7, pp. 1367 – 1378, 2013

[6] S. Fiori, T. Gong and H.K. Lee, “Bivariate nonisotonic statistical regression by a lookup table neural system”, *Cognitive Computation*, Vol. 7, No. 6, pp. 715 – 730, 2015

[7] D. Gao, “Texture model regression for effective feature discrimination: Application to seismic facies visualization and interpretation”, *Geophysics*, Vol. 69, No. 4, pp. 958 – 967, 2004

[8] J. Gill, R. Malyuk, O. Djurdjev and A. Levin, “Use of GFR equations to adjust drug doses in an elderly multi-ethnic group — a cautionary tale”, *Nephrology Dialysis Transplantation*, Vol. 22, No. 10, pp. 2894 – 2899, 2007

[9] A. Grubb, S. Blirup-Jensen, V. Lindstrom, C. Schmidt, H. Althau and I. Zegers, “First certified reference material for cystatin C in human serum ERM-DA471/IFCC”, *Clinical Chemistry and Laboratory Medicine*, Vol. 48, No. 11, pp. 1619 – 1621, 2010

[10] F.E. Harrell, *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*, Springer Series in Statistics, 2015

[11] I. Janssen, P. T Katzmarzyk and R. Ross, “Waist circumference and not body mass index explains obesity-related health risk”, *American Society for Clinical Nutrition*, Vol. 79, pp. 379 – 384, 2004

[12] R.E. Kopp and R.J. Orford, “Linear regression applied to system identification for adaptive control systems”, *The American Institute of Aeronautics and Astronautics Journal*, Vol. 1, No. 10, pp. 2300 – 2306, 1963

[13] B.L. Laskin, E. Nehus, J. Goebel, J.C. Khoury, S.M. Davies and S. Jodele, “Cystatin C-estimated glomerular filtration rate in pediatric autologous hematopoietic stem cell transplantation”, *Biology of Blood and Marrow Transplantation*, Vol. 18, No. 11, pp. 1745 – 1752, 2012

[14] A.S. Levey, J.P. Bosch, J.B. Lewis, T. Greene, N. Rogers and D. Roth, “A more accurate method to estimate glomerular filtration rate from serum creatinine: a new prediction equation. Modification of Diet in Renal Disease Study Group”, *Annals of Internal Medicine*, Vol. 130, pp. 461 – 470, 1999

[15] A.S Levey, L.A. Stevens, C.H. Schmid, Y.L. Zhang, A.F. Castro III, H.I. Feldman, J.W. Kusek, P. Eggers, F. van Lente, T. Greene and J. Coresh, “A new equation to estimate glomerular filtration rate”, *Annals of Internal Medicine*, Vol. 150, No. 9, pp. 604 – 612, 2009

[16] P. Mair, K. Hornik and J. de Leeuw, “Isotone optimization in R: pool-adjacent-violatorsaAlgorithm (PAVA) and active set methods“, *Journal of Statistical Software*, Vol. 32, No. 5, pp. 1 – 24, 2009

[17] E. Masry, “Multivariate local polynomial regression for time series: uniform strong consistency and rates”, *Journal of Time Series Analysis*, Vol. 17, pp. 571 – 599, 1996

[18] K. Matsushita, M. van der Velde, B.C. Astor, M. Woodward, A.S. Levey, P.E. de Jong, J. Coresh and R.T. Gansevoort, “Association of estimated glomerular filtration rate and albuminuria with all-cause and cardiovascular mortality in general population cohorts: a collaborative meta-analysis”, *The Lancet*, Vol. 375, No. 9731, pp. 2073 – 2081, 2010

[19] J.M. McArthur, R.J. Howarth and T.R. Bailey, “Strontium isotope stratigraphy: LOWESS version 3: Best fit to the marine Sr-isotope curve for 0-509 Ma and accompanying look-up table for deriving numerical age”, *The Journal of Geology*, Vol. 109, No. 2, pp. 155 – 170, 2001

[20] M.M. Mukaka, “A guide to appropriate use of correlation coefficient in medical research”, *Malawi Medical Journal*, Vol. 24, No. 3, pp. 69 – 71, 2012

[21] V. Rigalleau, C. Lasseur, C. Raffaitin, C. Perlemoine, N. Barthe, P. Chauveau, C. Combe and H. Gin, “The Mayo clinic quadratic equation improves the prediction of glomerular filtration rate in diabetic subjects”, *Nephrology Dialysis Transplantation*, Vol. 22, No. 3, pp. 813 – 818, 2007

[22] A.D. Rule, T.S. Larson, E.J. Bergstralh, J.M. Slezak, S.J. Jacobsen and F.G. Cosio, “Using serum creatinine to estimate glomerular filtration rate: accuracy in good health and in chronic kidney disease”, *Annals of Internal Medicine*, Vol. 141, No. 12, pp. 929 – 937, 2004

[23] G.J. Schwartz, A. Muñoz, M.F. Schneider, R.H. Mak, F. Kaskel, B.A. Warady and S.L. Furth, “New equations to estimate GFR in Children with CKD”, *Journal of the American Society of Nephrology*, Vol. 20, No. 3, pp. 629 – 637, 2009

[24] G.J. Schwartz, M.F. Schneider, P.S. Maier, M. Moxey-Mims, V.R. Dharnidharka, B.A. Warady, S.L. Furth and A. Muñoz, “Improved equations estimating GFR in children with chronic kidney disease using an immunonephelometric determination of cystatin C”, *Kidney International*, Vol. 82, No. 4, pp. 445 – 453, 2012

[25] M.A. Serdar, I. Kurt, F. Ozcelik, M. Urhan, S. Ilgan, M. Yenicesu, T. Kutluay, “A practical approach to glomerular filtration rate measurements: creatinine clearance estimation using cimetidine”, *Annals of Clinical & Laboratory Science*, Vol. 31, No. 3, pp. 265 – 273, 2001

[26] O. Simonsen A. Grubb and H. Thysell, “The blood serum concentration of cystatin C (gamma-trace) as a measure of the glomerular filtration rate”, *Scandinavian Journal of Clinical and Laboratory Investigation*, Vol. 45, No. 2, pp. 97 – 101, 1985

[27] C. Slot, “Plasma creatinine determination a new and specific Jaffe reaction method”, *Scandinavian Journal of Clinical and Laboratory Investigation*, Vol. 17, No. 4, pp. 381 – 387, 1965

[28] I. Soveri, U.B. Berg, J. Björk, C.G. Elinder, A. Grubb, I. Mejare, G. Sterner and S.E. Bäck, “Measuring GFR: a systematic review”, *American Journal of Kidney Diseases*, Vol. 64, No. 3, pp. 411 – 424, 2014

[29] M.J. Welch, A. Cohen, H.S. Hertz, K.J. Ng, R. Schaffer, P. van der Lijn, and E. White, “Determination of serum

creatinine by isotope dilution mass spectrometry as a candidate definitive method", *Analytical Chemistry*, Vol. 58, No. 8, pp. 1681 – 1685, 1986

[30] K. Zheng, M. Gong, Y. Qin, H. Song, X. Shi, Y. Wu, F. Li and X. Li, "Validation of glomerular filtration rate-estimating equations in Chinese children", *PLoS ONE*, Vol. 12, No. 7, pp. e0180565 (doi:10.1371/journal.pone.0180565), 2017

Appendix A. MATLAB codes to implement the estimating functions. The MATLAB codes used to implement the functions explained in Section 2 in the numerical application to GFR estimation are reported in the present Appendix for the convenience of the reader.

The Figure A.1 shows the MATLAB code used to estimate the cumulative distribution function in a given query-value.

```
% evaluate cumulative distribution function inferred from D for values q
function P = cdf(D,q)
  D = sort(D);
  nd = length(D);
  nq = length(q);
  L = zeros(nq, 1);
  R = zeros(nq, 1);

  % for each query
  for k = 1:nq
    % binary search for query's "neighbours"
    l = 1;
    r = nd;
    while r - l > 1
      m = floor((l+r)/2);
      if D(m) > q(k) && D(m) ≠ D(l)
        r = m;
      else
        l = m;
      end
    end

    while D(r) == D(l)
      l = l - 1;
    end

    % reach last occurrence of right neighbour
    while r < nd && D(r) == D(r+1)
      r = r + 1;
    end
    L(k) = l;
    R(k) = r;
  end
  % vectorized interpolation
  d = (q-D(L))./(D(R)-D(L));
  P = (L + (R - L).*d)/nd;

  % trim out of range interpolations
  P = P .* (P > 0); % if p<0 then p=0
  P = P - (P - 1).*(P > 1); %if p>1 then p=1

end
```

Fig. A.1: MATLAB code used to implement the CDF function.

The Figure A.2 shows the MATLAB code used to estimate the inverse cumulative distribution function in a given query-value.

```matlab
% evaluate inverse cumulative distribution fuction inferred from D for
% probabilities P. (D must contain at least two different values.)
function x = invcdf(D, P)
  nd = length(D);
  D = sort(D);
  np = length(P);
  P = P*nd;       % float in [0,nd]
  R = ceil(P);    % int in [0, nd]
  R = R + (R==0); % int in [1, nd], points in [0,1) are interpolated externally
  L = zeros(np,1);

  % for each query

  for k = 1:np
    r = R(k);
    l = r;

    % find last occurrence of pointed element
    while r < nd && D(r) == D(r+1)
      r = r + 1;
    end

    % find first occurrence of pointed element
    while l > 1 && D(l - 1) == D(l)
      l = l - 1;
    end

    if l == 1
      % find last occurrence of following element
      l = r;
      r = r + 1;
      while r < nd && D(r) == D(r+1)
        r = r + 1;
      end
    else
      % find last occurrence of previous element
      l = l - 1;
    end

    R(k) = r;
    L(k) = l;
  end

  % vectorized interpolation
  d = (P-L)./(R-L);
  x = (D(L) + (D(R) - D(L)).*d);

end
```

Fig. A.2: MATLAB code used to implement the invCdf function.

The Figure A.3 shows the MATLAB code used to implement the statistical bivariate regression algorithm.

A clear advantage of the current version of the codes over the previous structure developed in [5] is that the current version does not make extensive use of high-level (pre-built) functions and can therefore

```
% all arguments must be column vectors
% decresing model hypothesis
function [Qy] = sbr(Dx,Dy,Qx)

  P = 1 - cdf(Dx,Qx); % = pmf(Dx,Qx) % for increasing model
  Qy = invcdf(Dy, P);

end
```

Fig. A.3: MATLAB code used to implement the proposed SBR algorithm.

be easily re-coded into a lower-level programming language such as C/C++ for efficiency and portability.

The MATLAB implementation of the proposed SBR algorithm is already efficient in computation. To get a glimpse of the computational burden of the proposed SBR algorithm in comparison with the NANS SBR algorithm presented in [5], we tested the algorithms on the creatinine/GFR dataset discussed in the Section 3 on a number N of query points variable from 1 to 100 (we recall that the dataset contains 87 data-pairs). The Table A.1 shows runtimes (in ms) to complete 1000 independent trials on a 2.4 GHz dual-core machine running the Matlab R2017a environment.

Algorithms	$N = 1$	$N = 10$	$N = 50$	$N = 100$
NANS SBR model	0.376978	0.365465	0.371726	0.383950
Binning-less SBR model	0.123865	0.148599	0.231372	0.379054

Table A.1. Runtimes (in ms) of the algorithms tested on the creatinine/GFR dataset discussed in the Section 3 on a variable number N of query points, averaged over 1000 independent trials to get rid of statistical machine-level fluctuations.

The proposed algorithm appears suitably light in computation complexity to get paired to bed-side testing.