1

Bedside Linear Regression Equations to Estimate Equilibrated Blood Urea

Elmer A. Fernández[1,2], Mónica Balzarini[2,3] and Rodolfo Valtuille[4]
[1]*Faculty of Engineering, Catholic University of Córdoba*
[2]*National Council of Scientific and Technological Research (CONICET)*
[3]*Biometry Department, National University of Córdoba*
[4]*Fresenius Medical Care*
Argentina

1. Introduction

Three decades ago Sargent and Gotch established the clinical applicability of Kt/V, a dimensionless ratio which includes clearance of dialyzer (K),duration of treatment(t) and volume of total water of the patient (V), as an index of Hemodialysis (HD) adequacy (Gotch & Keen, 2005). This parameter, derived from single-pool(sp) urea(U) kinetic modelling, has become the gold standard for HD dose monitoring and it is widely used as a predictor of outcome in HD populations (Locatelli et al., 1999; Eknoyan et al., 2002; Locatelli, 2003). However, this spKt/V overestimates the HD dose because it does not take into account the concept of U rebound (UR). UR begins immediately at the end of HD session and it is completed 30-60 minutes after. UR is related to disequilibriums in blood/cell compartments as well as the flow between organs desequilibriums, both produced during HD treatment.
Therefore, equilibrated (Eq) Kt/V is the true HD dose and it requires the measurement of a true eqU when UR is completed. A blood sample to obtain an eqU concentration has several drawbacks that make this option impractical (Gotch and Keen,2005). For this reason in the last decade several formulas were developed to predict the eqU and also (Eq) Kt/V eliminating the need of waiting for a equilibrated urea mesurement. For instance, the "rate formula" (Daurgidas et al., 1995) is the most popular and validated equation. It is based in the prediction of (Eq)Kt/V as a linear function of (sp)Kt/V and the rate of dialysis(K/V). Another approach has been proposed by Tattersall, a robust formula based on double–pool analysis (Smye et al.1999). However, spite this eqU prediction approach is conceptually rigorous, it is not accurate (Gotch, 1990; Guh et al., 1999; Fernandez et al., 2001). Consequently, the availability of a model to predict subject-specific equilibrated concentration will be very helpful.
Although the behaviour of urea is non-linear since its extraction from blood follows some exponential family model as a function of time, we found that prediction of its equilibrated concentration after the end of the treatment session by means of linear models is accurate. In this study, we have shown how to build linear models to predict equilibrated urea based on two statistical procedures and a machine learning method that can be implemented in hemodialysis centres. The fitted model can be used for daily treatment monitoring and is

easily implemented in common available spreadsheets. A linear model is based on linear combinations of unknown parameters which must be estimated from data. The first step in looking for an appropriate model relies on prior knowledge or basic assumptions about the problem at hand that should be expressed in a hypothesized mathematical structure. The model can be expressed as $E(\mathbf{Y})=f(\mathbf{X},\boldsymbol{\beta})$, where $E(\mathbf{Y})$ is the expected value of the output vector, "f " is a linear function, i.e. $E(y_i) = f(\mathbf{x}_i,\boldsymbol{\beta}) = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + ... + x_{ip}\beta_p$, $\mathbf{X}$ is a matrix of input variables and $\boldsymbol{\beta}$ is a vector of parameters that needs to be estimated. In this way a set of potential mappings has been defined. The second step implies the estimation of the components of the vector $\boldsymbol{\beta}$. This step includes the selection of a specific mapping (a 'proper' $\boldsymbol{\beta}$) from the set of possible ones, choosing the parameter vector $\boldsymbol{\beta}$ that performs best according to some optimization criteria. There are several techniques to find a proper $\hat{\boldsymbol{\beta}}$ when using a linear model, being $\hat{\boldsymbol{\beta}}$ an estimation of $\boldsymbol{\beta}$ vector. Each of them has its own assumptions and requirements. Here we explore three different approaches for the estimation of the parameters of the $\boldsymbol{\beta}$ vector. They are: the Ordinary Least Square (OLS) procedure, based on the minimization of the sum of squared residuals $\sum_{i=1}^{N}\left(y_i - f\left(\mathbf{x}_i,\hat{\boldsymbol{\beta}}\right)\right)$ which assume independence on the $\mathbf{X}$ matrix columns. The Partial Least Square (PLS) method based on decomposition schema maximizing the estimated covariance between the input and its outputs, and which is able to handle co-linearity or lack of independence among the $\mathbf{X}$ matrix columns. Finally, we use the Support Vector Machine algorithm (SVM) which is based on the minimization of the empirical risk over ε-sensitive loss functions. In this study, the three regression procedures were used to estimate the $\boldsymbol{\beta}$ coefficients in order to predict the equilibrated urea concentration at the end of the dialysis session. The input variables were the intradialysis urea concentrations (U_0, U_{120}, U_{240}), the predialysis body weight and ultrafiltration patient data. Data analysis and modeling requires performing several tasks. In this work we use the Knowledge Discovery in Data Base (KDD) strategy as an ordered analysis framework. In this sense several steps involving different KDD stages such as problem/data understanding, collection, cleaning, pre-processing, analysis-modeling and results interpretation were implemented.

2. Material and methods

2.1 Data collection

2.1.1 Patients

One hundred and nine stable patients were selected from two dialysis units as follows: sixty one from Unit1 (mean age 56±3.5 years and mean time on dialysis (MTD) 32±12.3 months) and 48 from Unit2 (mean age 58±18.0 and MTD of 42±23.5). All patients were from Buenos Aires, Argentina, and were subjected to chronic HD treatment for at least 3 months. The selection criteria to include patients in the study were: (1) patients without infection or hospitalization in the previous 30 days; (2) patients with an A-V fistula (70% autologous fistula and 30% prosthetic fistula) with a blood flow rate (QB) of ≥ 300 ml/min, and (3) patients having consented to participate in the study. The study protocol complied with the Helsinki Declaration and was approved by the Ethical Committee of the Catholic University of Córdoba. All patients received HD three times a week with current hemodialysis machines using variable bicarbonate and sodium. Hollow-fiber polysulfone and cellulose diacetate dialyzers were used (see Fernandez et al, 2001 for more details). For the purpose of

this study, all patients were dialyzed over 240 min and the flows of blood (QB) and dialysate (QD) were fixed at 300 and 500 ml/min, respectively. It is known that hemodialysis dose is influenced by several factors including dialysis time, hemodialysis schedule and blood and dialysate flow (Daugirdas et al. 1997). In order to decrease the complexity, such variables were handled externally, fixing their values to control their effects on the equilibrated urea prediction model.

2.1.2 The input and output variables

Blood samples were obtained at the mid-week HD session. They were taken from the arterial line at different times to obtain urea determinations: 1) predialysis urea (U_0), at the beginning of the procedure; 2) intradialysis urea (U_{120}), in the middle of the HD session (at 120 min from the beginning); 3) postdialysis urea (U_{240}), at the end of the HD session.

For the intradialysis urea (U_{120}) and postdialysis urea (U_{240}), QB was slowed to 50 ml/min and blood was sampled 15 seconds later. At this point, access recirculation ceased and the dialyzer inlet blood reflected the arterial urea concentration. Regarding the protocols for intradialysis samples, it is worth noting that originally Smye et al. 1997 proposed taking them within 60 min from the beginning of the session and at 20 min before its finalization. We, however, decided to take the intradialysis sample 120 min after the beginning of the HD session (U_{120}), which allowed us to compare our results with those reported by Guh et al. 1999.

Urea (U) determinations were performed in triplicate on each blood sample using autoanalyzers (see Fernandez et al, 2001 for more details). The urea averages were calculated and recorded with an accuracy of 1% for both machines. For information about the pre- and post-treatment status of the patient, we used the pre- and post-dialysis body weights (BW_0, BW_{240}). Both variables are commonly used in clinical practice to decide the treatment schedule as well as to calculate the treatment dose. These variables were recorded in the same dialysis session when the blood samples were taken.

The output variable was the equilibrated urea. For the purpose of this study, the patients were retained one hour in the dialysis center and the equilibrated urea levels (U_{eq}) were extracted 60 min after the end of HD. The summary statistics for the input and output variables are shown in Table 1

	U_0	U_{120}	U_{240}	Bw	UF	U_{eq}
Min	59	31	21	45.3	0.0	23
1st Quantile	127	64	40	59.4	2.0	50
Median	149	77	49	71.0	2.7	59
Mean	149	80	53	72.0	2.7	62
3rd Quantile	169	96	62	83.8	3.3	76
Max	221	144	98	119.0	5.5	112

Table 1. Summary statistics of the patient data distribution.

2.2 Ordinary least squares

The Ordinary Least Square approach estimates the $\boldsymbol{\beta}$ coefficient vector by minimizing the sum of squared residuals from the data

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i - f\left(\tilde{x}_i, \boldsymbol{\beta}\right)\right)^2 \tag{1}$$

where $\tilde{x}_i = \left(1, x_i\right)$ with x_i the "*i-th*" row of the input matrix **X**. The algorithm looks for the **β** that minimize (1). This is achieved taking derivatives of equation 1 and setting them to zero, yielding the following closed solution:

$$\hat{\boldsymbol{\beta}}_{OLS} = \left(\tilde{\mathbf{X}}^t \cdot \tilde{\mathbf{X}}\right)^{-1} \cdot \tilde{\mathbf{X}}^t \cdot \mathbf{Y} \tag{2}$$

where "*t*" means "transpose" and $\left(\tilde{\mathbf{X}}^t \cdot \tilde{\mathbf{X}}\right)$ is a singular matrix with $\tilde{\mathbf{X}}$ the extended input matrix holding $\tilde{x}_i = \left(1, x_i\right)$ in each row.

2.3 Partial least squares

Partial Least Squares not only generalizes but also combines features from regression and Principal Component Analysis, to deal with correlated explanatory variables in linear models (abdi, 2003, Shawe-Taylor & Cristianini, 2005). It is particularly useful when one or several dependent variables (outputs) must be predicted from a large and potentially highly correlated set of independent variables (inputs). In the PLS algorithm (Wood et al., 2001), **X** and **Y** are expressed as:

$$\mathbf{X}^{N\times p} = \mathbf{T}^{N\times A}\left(\mathbf{P}^{p\times A}\right)^t + \mathbf{H}^{N\times p} \tag{3}$$

$$\mathbf{Y}^{N\times p} = \mathbf{U}^{N\times A}\left(\mathbf{C}^{1\times A}\right)^t + \mathbf{R}^{N\times p} \tag{4}$$

where A is the number of PLS factors (A ≤ p) and **H** and **R** are error matrices. The columns of **T** and **U** ("score" matrices) provide a new representation of the **X** and **Y** variables in an orthogonal space. The matrices **P** and **C** are the projections ("loadings") of the **X** and **Y** columns into the new set of variables in **T** and **U**. The **T** matrix is calculated as $\mathbf{T=X\cdot W}$ where $\mathbf{W=U(P'U)^{-1}}$. In the PLS algorithm, **U** and **P** are built iteratively (Wood et al.,2001) by means of matrix products between consecutive deflations of the original matrices **X** and **Y**. Thus, the **T** matrix is also a good estimator of **Y**, so

$$\mathbf{Y}^{N\times p} = \mathbf{T}^{N\times A}\left(\mathbf{C}^{1\times A}\right)^t + \mathbf{E}^{N\times p} \tag{5}$$

where $\mathbf{C}^{1\times A}$ is the "loadings" matrix of **Y** that projects it over the new space represented by **T**. The error term in **E** represents the deviations between the observed and predicted responses. Replacing **T** in the above equation yields:

$$\mathbf{Y} = \mathbf{X}\cdot\mathbf{W}\cdot\mathbf{C}^t + \mathbf{E} = \hat{\boldsymbol{\beta}}_{PLS}\cdot\mathbf{X} + \mathbf{E} = \hat{\mathbf{Y}} + \mathbf{E} \tag{6}$$

where $\hat{\mathbf{Y}}$ is the predicted output.

The number of factors chosen impacts the estimation of the regression coefficients. In a model with "A" factors, the **β** coefficients are calculated as follows:

$$\hat{\boldsymbol{\beta}}_{PLS}^{p\times 1} = \mathbf{W}^{p\times A}\left[\mathbf{C}^{1\times A}\right]^{t} \tag{7}$$

In the PLS algorithm the input and output data are centered prior to calculate the different matrices. In addition the input training matrix **X** could be scaled dividing each column by its standard deviation. Thus, regression coefficients estimated by means of equation (7) lives in the scaled **X** domain. The values of the **β** coefficients in the raw data domain are calculated as follows:

$$\hat{\mathbf{Y}} = \mathbf{V}^{-1}\hat{\boldsymbol{\beta}}_{PLS}\mathbf{X}_{raw} + \mathbf{V}^{-1}\hat{\boldsymbol{\beta}}_{PLS}\overline{\mathbf{X}}_{raw} + \overline{Y} = \hat{\beta}_0 + \hat{\boldsymbol{\beta}}_{raw}\mathbf{X}_{raw} \tag{8}$$

where $\hat{\mathbf{Y}}$ is the estimated Ueq, $\mathbf{V}$ is a diagonal matrix of standard deviations for each column of **X** and $\overline{\mathbf{X}}$ is the vector of columns means from **X**. $\overline{Y}$ is the mean of the response variable from the training data set, and $\hat{\beta}_0 = \mathbf{V}^{-1}\hat{\boldsymbol{\beta}}_{PLS}\overline{\mathbf{X}}_{raw} + \overline{Y}$ is the intercept.

2.4 Support vector machine

In previous cases, the sum of squared deviation of the data can be viewed as a loss function measuring the amount of loss associated with the particular estimation of **β**. In the Support Vector Machine framework (Vapnik, 2000), the loss function only provides information on those data points from which the loss is beyond a threshold ε yielding to

$$L_{\varepsilon}^{p}(\mathbf{x}, y, f) = \left|y - f(\mathbf{x})\right|_{\varepsilon}^{p} = \max\left(0, \left|y - f(\mathbf{x})\right|^{p} - \varepsilon\right) \tag{9}$$

with *p=1* or *2*. Then the algorithm try to minimize an empirical risk defined as

$$R_{emp}(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N}\left(\left|y_i - f(x_i, \boldsymbol{\beta})\right|_{\varepsilon}^{p}\right) \tag{10}$$

constrained to $\|\boldsymbol{\beta}\|^2 \le C$ where *C* is a user defined constant, playing a role of regularization constant, a trade-off between complexity and losses.
The optimization problem, in primal form, can be defined as follows

$$\text{minimize } \|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\left(\xi_i^p + \xi'^p_i\right)$$

$$\text{subject to } \begin{cases} \left(\langle \boldsymbol{\beta}\cdot x_i\rangle + \beta_0\right) - y_i \le \varepsilon + \xi_i \;; i = 1...N \\ y_i - \left(\langle \boldsymbol{\beta}\cdot x_i\rangle + \beta_0\right) \le \varepsilon + \xi'_i \;; i = 1...N \\ \xi_i, \xi'_i \ge 0 \;; i = 1...N \end{cases} \tag{11}$$

The ξ and ξ' symbols represent slack variables for those points above or below the target in more than ε and $\xi_i\xi'_i = 0$. This minimization problem can be rewritten in terms of Lagrange multipliers (dual form) as

$$\text{Maximize } \sum_{i=1}^{N} y_i\left(\alpha'_i-\alpha_i\right)-\varepsilon\sum_{i=1}^{N}\left(\alpha'_i-\alpha_i\right)-\frac{1}{2}\sum_{i,j=1}^{N}\left(\alpha'_i-\alpha_i\right)\left(\alpha'_i-\alpha_i\right)\left(\left\langle x_i\cdot x_j\right\rangle+\frac{\delta_{ij}}{C}\right)$$

$$\text{subject to } \begin{cases}\sum_{i=1}^{N}\left(\alpha'_i-\alpha_i\right)=0\\ \alpha'_i;\alpha_i\geq 0,\ i=1..N\end{cases} \tag{12}$$

where α,α' are Lagrangean multipliers satisfying $\alpha_i\alpha'_i=0$ and $\frac{\delta_{ij}}{C}=0$ for p=*1*. The following Karush-Kuhn-Tucker conditions should also be satisfied

$$\alpha_i\left(\left\langle\boldsymbol{\beta}\cdot x_i\right\rangle+\beta_0-y_i-\varepsilon-\xi_i\right)=0,\ i=1...N$$
$$\alpha_i\left(y_i-\left\langle\boldsymbol{\beta}\cdot x_i\right\rangle-\beta_0-\varepsilon-\xi'_i\right)=0,\ i=1...N$$

Then the link between the dual and primal representation is given by

$$\hat{\boldsymbol{\beta}}_{SVM}=\sum_{i=1}^{N}\left(\alpha_i-\alpha'_i\right)x_i$$

where $\alpha_i,\alpha'_i\geq 0$ (Cristianini and Shawe-Taylor,2000).

In our application, for the SVM case, both input and output training data where centered and scaled to have zero means and unity standard deviation. The values of the **β** coefficients in the raw data domain were calculated as follows:

$$\hat{\mathbf{Y}}=sd_{\mathbf{y}}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}_{SVM}\mathbf{X}_{raw}+sd_{\mathbf{y}}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}_{SVM}\overline{\mathbf{X}}_{raw}+\overline{Y}=\hat{\beta}_0+\hat{\boldsymbol{\beta}}_{SVM}\mathbf{X}_{raw} \tag{13}$$

where $\hat{\mathbf{Y}}$ is the estimated Ueq, $\mathbf{V}$ is a diagonal matrix of standard deviations for each column of $\mathbf{X}$ and $\overline{\mathbf{X}}$ is the vector of columns means from **X.** The mean and standard deviation of Ueq from training data set are $\overline{Y}$ and $sd_{\mathbf{y}}$, respectively. The intercept is expressed as $\beta_0=sd_{\mathbf{y}}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}_{SVM}\overline{\mathbf{X}}_{raw}+\overline{Y}$.

2.5 Statistical modeling of equilibrated urea

The three estimation procedures (OLS, PLS, and SVM) to obtain the regression coefficients **β** of a linear model where applied to build bed side equations to estimate equilibrated urea from intradialysis urea samples and anthropometric data in 109 hemodialyzed patients. Estimation, selection and validation of the model were implemented in R language (www.r-project.org) (see appendix).Prior to fit a model, the appropriate number of factors (A) ,the best cost (*C*) and epsilon (*ε*) pairs values were chosen for PLS and SVM, respectively. For this purpose, a 15 fold cross validation strategy was applied over 70% randomly chosen patients from the data set. In the PLS case, models including 1 to *A* factors with *A=1, 2, 3, 4* and *5* were tested. For each model the cross validation root mean prediction error (RMPE) was calculated. Then the expected value of the RMPE over all partitions was obtained. The model achieving the smaller RMPE mean was chosen. For the linear SVM case, a *Cxε* 10x10 grid searches was performed. The ranges were from *4* to *6* for *C* and from *0.001* to *2* for *ε*. A linear SVM model was built for each (*C*,*ε*) pairs and the cross validation RMPE was calculated and compared. The smaller RMPE mean was used as selection criteria. The

predictive ability of the fitted models was evaluated using a 20 fold cross-validation strategy over the whole data set. The data set was split in 20 consecutive sets of equal size and 19 were alternatively used for **β** estimation and one for prediction from the estimated model.

3. Results

In table 2, cross validation statistics for PLS models with different number of factors is shown. Table 2 summarizes mean and standard deviation of Mean Prediction Error (RMPE) and mean and standard deviation of correlations between estimated and measured Ueq (R). It is possible to see that a PLS model with 3 or 4 components are very competitive. We chose a linear fit with 3 Factors because it yields the lowest RMPE with a parsimonious model

# Factors	$\overline{\text{RMPE}}$
1	27.03
1,2	20.69
1,2,3	19.28
1,2,3,4	19.69
1,2,3,4,5	19.82

Table 2. Expected prediction error for PLS model with different number of factors.

In Fig.1 the achieved RMPE of the SVM models are shown for each $C\times\varepsilon$ grid point. The chosen $C\times\varepsilon$ pair was C= *4.2222* and ε= *0.2223* (filled circle in Fig.1)

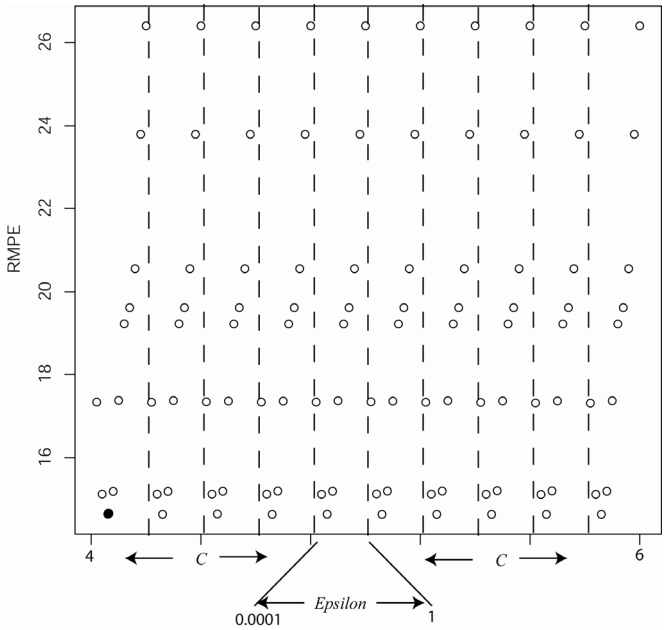


Fig. 1. Cross-validation MSE for each $C\times\varepsilon$ combination in the SVMR algorithm. The best $C\times\varepsilon$ combination pair is indicated with a filled circle

Once the PLS and SVM models where selected, *i.e.* a 3 PLS factor model and a SVM trained with C= *4.2222* and ε= *0.2223*, the 3 methods (OLS, $PLS_{A=3}$ and $SVM_{C=4.222,\varepsilon=0.2223}$) where evaluated over the whole data set with a 20-fold cross-validation strategy. In Fig. 2 the relative prediction error (%PE) vs. true equilibrated Urea and its corresponding smooth trend are shown for the three estimation strategies. In open circles the OLS (dashed smooth trend) approach, in * PLS errors (dot-dashed smooth trend) and in "+" symbol the SVM errors (dotted smooth trend). It is possible to see that OLS and PLS performs almost equal with a small tendency to increased over estimation for PLS in high Ueq values (the PLS smooth trend curve shows greater %PE than in the other cases). On the contrary, SVM performs better for low Ueq (dotted smooth trend closer to zero %PE). In the midrange of Ueq the three methods performs similar. All the methods tend to overestimate small Ueq values and under estimate high Ueq values.

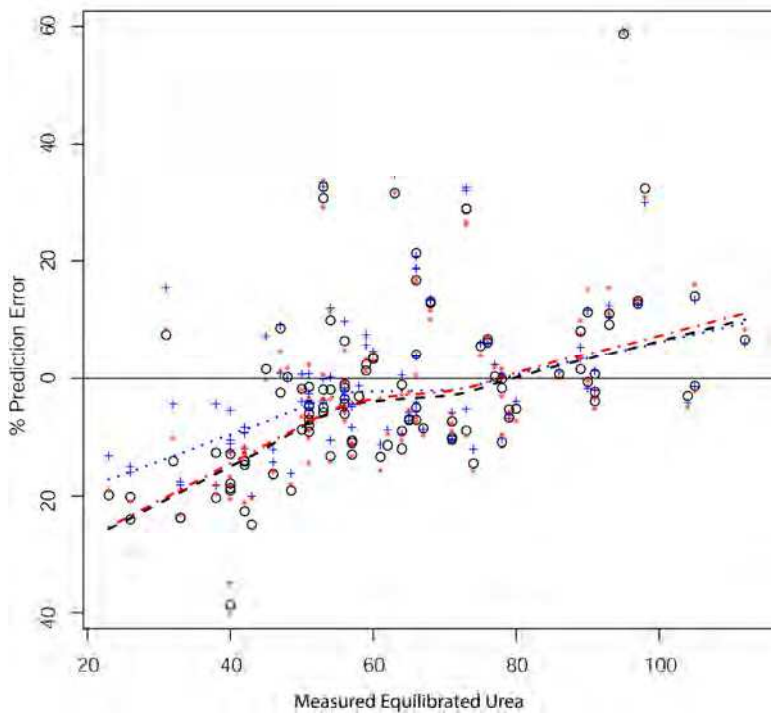


Fig. 2. 20-Fold cross-validation % prediction errors (%PE) for each tested model. Open circles for OLS model, "*" for PLS and "+" for SVMR. The smooth trend curve for each model is also presented (see text for references)

In Table 3, summary statistics for PE and the number of data points which have a %PE in the ±10 and ±20 ranges is shown. The PLS model achieves the lowest %PE and SVM the highest but with lesser standard deviation across runs. In terms of median we can see that all the methods tend to overestimate the response, however SVM presents the lower median of %PE suggesting robustness to outliers.

	Prediction Error			Percentage of data points with %PE in the range	
	Mean	SD	Median	-10≤%PE≤10	-20≤%PE≤20
OLS	0.08	9.59	-2.44	55.05%	85.32%
PLS	0.06	9.60	-.255	55.96%	87.16%
SVM	1.08	9.26	-1.72	63.30%	90.83%

Table 3. Summary statistics for prediction errors and number of data points laying in the ±10 and ±20 %PE interval

In Fig. 3 the distribution for the $\hat{\boldsymbol{\beta}}$ coefficients that weights each input variable ($\beta_1$for U0, $\beta_2$for U_{120}, $\beta_3$for U_{240}, β_4 for Bw_0, and β_5 for Uf) in the input scale (equation 8 for PLS and 13 for SVM) are shown. It is possible to see that coefficient β_5 (associated to Uf) is very variable. This coefficient is mainly estimated as positive by OLS, negative by PLS case and both by SVM. In the first two cases, β_5 was statistically different from zero ("t test" $p<0.01$). SVM estimation of $\boldsymbol{\beta}$ seems to be more robust than the other cases. In particular, the β coefficient related to Uf (β_5) shows significant less dispersion than in the other models. In the OLS and PLS cases, all except Uf coefficient, show similar behaviour. The Uf coefficient for PLS is the most variant among the rest.

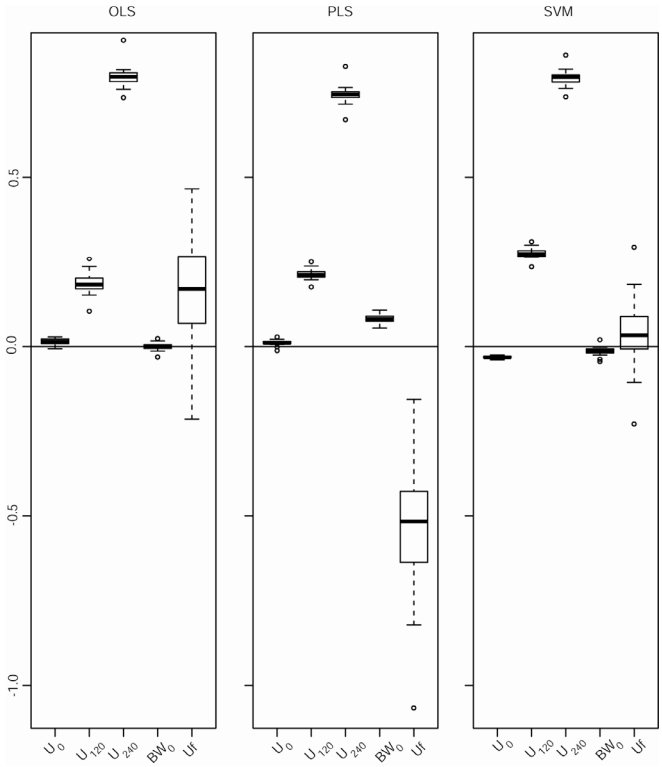


Fig. 3. Distribution of the $\hat{\boldsymbol{\beta}}$ coefficients for each input variable from the 20-Fold cross-validation.

3.1 Bed side equations for equilibrated urea prediction

Final models were built using the whole patients and using the parameters found in the previous section (for PLS and SVM). We found that the coefficients estimated using the full data set (equations 14 to 15) where similar to the mean of the cross validation coefficients for OLS and SVM. On the contrary, coefficients estimated by PLS where different when using the whole data set compared to those estimated in the cross validation evaluation.
In the OLS case the final bed side equation was the following:

$$\hat{Y} = 3.02449 + 0.01381 \cdot U_0 + 0.18576 \cdot U_{120} + 0.79713 \cdot U_{240} - 0.00028 \cdot BW_0 + 0.16252 \cdot Uf \quad (14)$$

In the PLS case and accounting only the first three PLS factors the achieved model is

$$\hat{Y} = 0.84616 + 0.00810 \cdot U_0 + 0.20652 \cdot U_{120} + 0.75386 \cdot U_{240} + 0.06862 \cdot BW_0 - 0.26812 \cdot Uf \quad (15)$$

For the SVM we get

$$\hat{Y} = 4.27754 - 0.03362 \cdot U_0 + 0.27904 \cdot U_{120} + 0.78921 \cdot U_{240} - 0.01210 \cdot BW_0 + 0.02323 \cdot Uf \quad (16)$$

The SVM identify 77 support vectors. This means that the $\hat{\boldsymbol{\beta}}$ coefficients were estimated using only %70 of the data base. On the contrary, the other two methods require the full data set to build the solution.

4. Discussion

In this work we show how to build linear models from three different linear regression estimation procedures relying on different optimization algorithms. Ordinary Least squares is based on the minimization of the sum of squared residuals while Partial Least Squares uses maximization of co-variance information by means of repetitive deflation of the input and output matrices based on correlation. Finally, the Support Vector Machine Regression is based on the empirical risk minimization of non-linear loss function. Theoretically, none of the method requires any specific assumption; however, it is known that if the observed variable (the equilibrated urea in this case) follows a normal distribution, the statistical significance of the $\boldsymbol{\beta}$ coefficients estimated by OLS and PLS can be proved.
Even though all the models predict similarly well, they show different estimates not only in value but also in sign for U_0, body weight and ultrafiltration. Analyzing the "raw" data relationships between these variables (see Fig. 4) and urea rebound $\left(U_{eq} - U_{240}\right)/U_{eq}$ it is possible to see the known [Gotch & Kleen, 2005] slightly inverse relationship (see smooth trend curves) between BW and Uf with urea rebound. This behaviour seems to be capture for Uf by PLS (negative β_5). The β_5 estimated by OLS method seems to follow the positive linear relationship mostly found in the Uf vs Ueq pairs plot. The SVM finds a solution in between, estimating much smaller values for β_5 than the others two. For the case of body weight coefficient (β_4), estimations by OLS and SVM are smaller than for PLS, however, SVM method captures the known small tendency between BW and urea rebound. In this sense, PLS is able to capture known biological relationships while still providing broad ranges for the estimation of the Uf coefficient. On the opposite OLS does not reflect the biological effect of Uf. The SVM method provides an in-between solution providing small estimates of the Uf coefficient. Thus, those methods that account for co-linearity (PLS and in some extent SVM) provide better solutions than OLS which do not account for it.

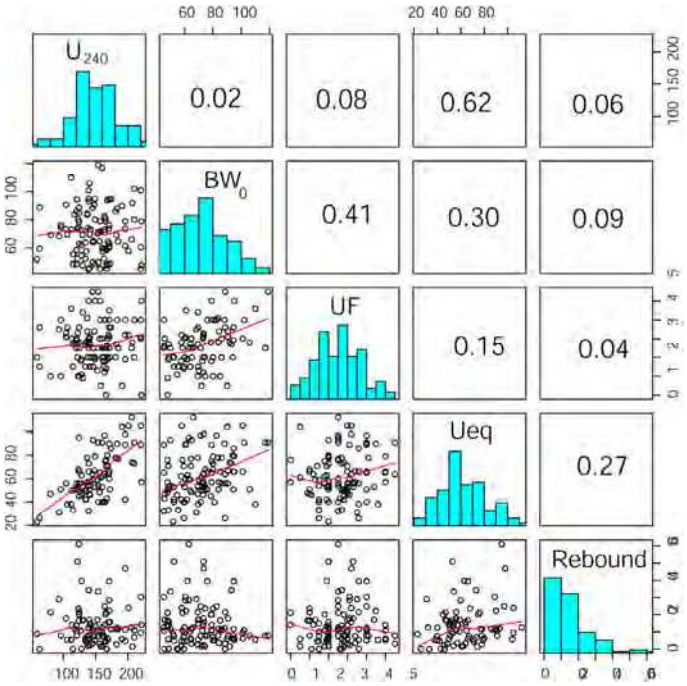


Fig. 4. Pairs plots and correlation coefficients between U_{240}, BW_0 Uf, Ueq and urea rebound.

We showed that by means of linear models we were able to build bedside equations that can be easily implemented in any calculator or electronic spreadsheet such as Excel®.

All the presented methods performed better than traditional methods (Smye et al, 1999) over the same data (Fernández et al, 2001) suggesting the appropriateness of the simple linear approaches. In addition, each hemodialysis centre can build its own predictor based on its own patient population by following the described process or implementing the accompanying source code (see appendix).

In this work we show that the use of an intradialysis sample (U_{120}) provided valuable information to predict the equilibrated urea. Smye et al. (1999) were the first to use an intradialysis sample to model Ueq. In clinical practice the extraction of an additional blood urea sample could be very problematic. In a recent publication (Fernandez et al, 2008) we showed that a linear model built without this urea sample can also provide accurate Ueq estimation. Future challenges for Ueq prediction by linear models are emerging with the implementation of different HD schedule proposals based on the variation of session time and/or weekly frequency.

5. Appendix: R source code for OLS, PLS and SVM linear models for estimate equilibrated urea

In order to apply the R (www.r-project.org) algorithm to build the linear models presented in this work, we assume that the patient data base is stored in a comma separated values (CSV) file as follows (any electronic spreadsheet program allows to save CSV files).

PatientID	U_0	U_{120}	U_{240}	BW_0	Uf	Ueq
1	121	63	47	94.5	2.9	51
2	166	87	68	59.4	1.4	71
3	196	68	40	61.6	1.9	42
4	167	73	43	45.7	2.6	43
5	128	64	46	54.8	1.1	46
6	127	77	50	72.6	1.8	56
7	139	49	28	45.3	2.5	32
…	…	…	…	…	…	…
…	…	…	…	…	…	…

Table 4. Data base in comma separated file format. The R code assumes this file for processing (PP: Body weight)

```
#set path to data directory
path=paste(base.path,"C:/...../UreaData/",sep="")
#we assume that the data base is a comma separated file
datafile="naneofpatientdatabase.csv"
#read file
udata=read.csv(paste(path,datafile,sep=""),h=T)
#load required libraries
library(pls)
library(e1071)
### SVM auxiliary functions ################
coef.svm<-function(model){
# return the beta coefficients in the possibly standardized space
  w=t(model$coefs)%*%model$SV
  w=c(-model$rho,w)
  names(w)=c("(Intercept)",attr(model$terms,"term.labels"))
  return(w)
}

getScaledCoef.svm=function(obj){
# return the beta coefficients in the raw scale
  if(obj$kernel!=0) stop("getScaledCoef: only for linear svm")
  ym=unlist(obj$y.scale["scaled:center"])
  sdy=unlist(obj$y.scale["scaled:scale"])
  xm=unlist(obj$x.scale["scaled:center"])
  sdx=diag(1/unlist(obj$x.scale["scaled:scale"]))
  beta=coef(obj)
  b0=beta["(Intercept)"]

  sdbeta=sdx%*%beta[-1]

  Intercept=b0-sdy*(xm%*%sdbeta)+ym
  betanew=sdy*sdbeta
  ret=c(Intercept,t(betanew))
  names(ret)=names(beta)
 return(ret)
}
```

```
########################################
## PLS auxiliary functions #############
getScaledCoef.pls=function(obj,comps=1:obj$ncomp){
  if(missing(comps) || is.null(comps))
    comps =1:model$ncomp
  beta = rowSums(coef(obj,comps=comps,cumulative=F),2)
  ym=obj$Ymeans
  xm=obj$Xmeans
  if(is.null(obj$scale)==TRUE){
   sdx=1
  }else sdx=diag(1/unlist(obj$scale))

  sdbeta=sdx%*%beta

  Intercept= ym-(xm%*%beta)
  betanew=sdbeta
  ret=c(Intercept,betanew)
  names(ret)=c("(Intercept)",attr(mpls$terms,"term.labels"))
 return(ret)
}
#######################################

####  Data set for model selection ###
samp=sample(nrow(udata),round(0.7*nrow(udata),0))
seldata=udata[samp,]
kfold=cvsegments(length(samp),15)#15 fold CV

####  SVM parameters selection ##########
scalar=TRUE
RESsvm=NULL
RESMsvm=NULL # cross validation result matrix
#set grid CxEps ranges
C.range=seq(4,6,length.out=10)
epsilon.range=seq(.0001,1,length.out=10)
for(cc in C.range){
  for(ep in epsilon.range){
   RESsvm=NULL
    for(cv in 1:length(kfold)){
        bestmod=svm(Ueq~Upre+U120+Upos+PP+UF,data=seldata[-
kfold[[cv]],],cost=cc,epsilon=ep,kernel="lin",scale=scalar)
        pp=predict(bestmod,seldata[kfold[[cv]],])
        #save squared errors, and correlation
        RESsvm=rbind(RESsvm,
c(sqrt(sum((pp-seldata[kfold[[cv]],"Ueq"])^2)),
```

```
 for(cv in 1:length(kfold)){
        bestmod=plsr(Ueq~Upre+U120+Upos+PP+UF,data=pdata[-
kfold[[cv]],],comps=nc)
        pp=predict(bestmod,newdata=pdata[kfold[[cv]],],
type="response",cumulative=F,comps=1:nc)
        RESpls=rbind(RESpls,
c(sqrt(sum((pp-
pdata[kfold[[cv]],"Ueq"])^2)),cor(pp,pdata[kfold[[cv]],"Ueq"])))

    }
   RESMpls=rbind(RESMpls,c(nc,mean(RESpls[,1]),
sd(RESpls[,1]),mean(RESpls[,2]),sd(RESpls[,2])))
  }

b=which.min(RESMpls[,2])
RESMpls[b,] #print the best parameters
##########################################
##.....................................###
#### Testing the final models #############
cv.sets=cvsegments(nrow(udata),20)#20 cross validation partition

model=formula("Ueq~Upre+U120+Upos+PP+UF")

### OLS method ####
coefslmnc=NULL
predlmnc=NULL
Ylmnc=NULL
i=2
for(i in 1:20){
  mlmnc=lm(Ueq~Upre+U120+Upos+PP+UF,data=pdata[-cv.sets[[i]],])

predlmnc=c(predlmnc,predict(mlmnc,pdata[cv.sets[[i]],]))#prediction
  Ylmnc=c(Ylmnc,pdata[cv.sets[[i]],"Ueq"])#real data

  coefslmnc=rbind(coefslmnc,coef(mlmnc))
}
errlm=Ylmnc-predlmnc#prediction error
errrellm=100*errlm/Ylmnc #relative error
#ploting and smooth trend lines
plot(y=predlmnc-Ylmnc,x=Ylmnc,ylim=c(-65,60))
lines(stats::lowess(x=Ylmnc, y=predlmnc-Ylmnc),col="black",lty=2)

###  SVM method ###
coefssvm=NULL
```

```
errsvm=Ysvm-predsvm #prediction error
errrelsvm=100*errsvm/Ysvm #relative prediction errors
points(y=predsvm-Ysvm,x=Ysvm,col="blue",pch="+")
lines(stats::lowess(x=Ysvm, y=predsvm-Ysvm),col="blue",lty=3)
abline(h=0)
### PLS method ###
coefspls=NULL
predpls=NULL
Ypls=NULL
sccpls=NULL
for(i in 1:20){

  mpls=plsr(Ueq~Upre+U120+Upos+PP+UF,data=pdata[-
cv.sets[[i]],],scale=T)
  predpls=c(predpls,predict(mpls,newdata=pdata[cv.sets[[i]],],
type="response",cumulative=F,comps=1:RESMpls[b,1]))
  Ypls=c(Ypls,na.omit(pdata[cv.sets[[i]],"Ueq"]))
  coefspls=rbind(coefspls,coef(mpls))
  sccpls=rbind(sccpls,getScaledCoef.pls(mpls,1:3) )
}
#boxplot(coefspls)
errpls=Ypls-predpls
errrelpls=100*errpls/Ypls

points(y=predpls-Ypls,x=Ypls,col="red",pch="*")
lines(stats::lowess(x=Ypls, y=predpls-Ypls),col="red",lty=4)
abline(h=0)

#### FULL model (final beta) #####

coef(lm(model,data=udata))
getScaledCoef(svm(model,data=udata,scale=T,kernel="lin",type="eps-
reg",cost=RESMsvm[b,1],epsilon=RESMsvm[b,2]))
getScaledCoef.pls(plsr(Ueq~Upre+U120+Upos+PP+UF,data=udata,scale=T)
,1:3)
```

6. References

Abdi H. (2003) Partial Least Squares (PLS) Regression. In: Lewis-Beck M, Bryman A, Futting T (Eds). *Encyclopedia of Social Sciences Research Methods*. Thousand Oaks, CA.

Cristianini, N., Shawe-Taylor, J.,(2000) *An Introduction to Support Vector Machines*, Cambridge University Press,.

Daugirdas J. (1995) Simplified equations for monitoring kt/v, pcrn, ekt/v and epcrn. *Adv. in Renal Replacement Therapy*. 2(4) 295-304,

Depner, T.A. (1999) History of dialysis quantification. *Sem. Dial*. 12:S1:14-19

Eknoyan, G.; Beck, G.J.; Cheung, A.K. et al (2002). Effect of dialysis dose and membrane flux in maintenance hemodialysis. *New Engl J Med* 347: 2010–2019

Fernández EA, Valtuille R, Willshaw P, Perazzo CA.(2001) Using Artificial Intelligence to predict the Equilibrated Blood Urea Concentration. *Blood Purification*. 19(3) 271-285

Fernández EA, Valtuille R, Willshaw P, Balzarini M. (2008) Partial Least Squares Regression: A Valueble Method for Modeling Molecular Behaviour in Hemodialysis. *Annals of Biomedical Engineering*. DOI: 10.1007/s10439-008-9492-1,

Ghu J, Yang J, Chen IU lai Y. (1998) Prediction of equlibrated BUN by an artificial neural network in high efficient hemodialysis. *Am. J. Kid. Dis*. 3: 638-646.

Gotch, F.A. (1990) *Kinetic modeling in hemodialysis*. W: Nissenson A.R., Fine R.N., Gentile D.: Clinical dialysis, 2nd ed., Appleton and Lange, Norwalk, CT

Gotch,FA, Keen, M. (2005) *Kinetic modeling in Hemodialysis in: Clinical Dialysis*,fourth edition,Nissenson,a and Fine,R,editors.

Locatelli, F.; Hannedouche, T.; Jacobson, S. et al (1999). The effect of membrane permeability on ESRD: design of a prospective randomized multicentre trial. *J. of Nephrol*. 12: 85-88.

Locatelli, F. (2003) Dose of dialysis, convection and hemodialysis patients outcome- what the HEMO study doesn't tell us: the European viewpoint. *Nephrol. Dail. Transp.* 18:1061-1065

Roa LM, Prado M. (2004) The role of urea kinetic modeling in assessing the adequacy of dialysis. Crit. Rev. Biomed. Eng. 32 (5-6): 461-539,

Tattersal J, Detakats D, Chamney P, Greenwood R, Farrington K. (1996) The post dialysis rebound: Predicting and quantifying its effect on Kt/V. *Kidney Int*. 50(6) 2094-2102,

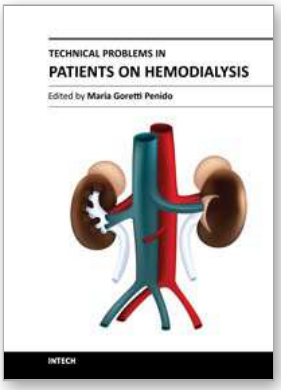
Shawe-Taylor J and Cristianini N. Kernel. (2005) *Methods for pattern analysis*. Cambridge UP. Cambridge

Smye S.W,. Will E.J, Lindley E.J. (2002) Postdialysis and Equilibrium Urea Concentrations. *Blood Purification*. 20: 189-189,

Smye S, Tattersal J, Will E. (1999) Modeling the post-dialysis rebound: The reconciliation of current formulas. *ASAIO*. 45(6) 562-569

Wold S, Sjöström M, Eriksson L. (2001) PLS-regression: a basic tool of chemometrics. *Chem. Int. Lab. Sys*. 58: 109-130

Vapnik. VN.(2000) *The Nature of Statistical Learning*. 2d ED. Springer



Technical Problems in Patients on Hemodialysis

Edited by Prof. Maria Goretti Penido

ISBN 978-953-307-403-0
Hard cover, 312 pages
Publisher InTech
Published online 07, December, 2011
Published in print edition December, 2011

This book provides an overview of technical aspects in treatment of hemodialysis patients. Authors have contributed their most interesting findings in dealing with hemodialysis from the aspect of the tools and techniques used.Each chapter has been thoroughly revised and updated so the readers are acquainted with the latest data and observations in the area, where several aspects are to be considered. The book is comprehensive and not limited to a partial discussion of hemodialysis. To accomplish this we are pleased to have been able to summarize state of the art knowledge in each chapter of the book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821