Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

A Meta-Learning Approach to the Regularized Learning – Case Study: Blood Glucose Prediction

V. Naumova, S. Pereverzyev, S. Sampath

RICAM-Report 2011-31

A Meta-Learning Approach to the Regularized Learning – Case Study: Blood Glucose Prediction

V. Naumova, S. V. Pereverzyev, S. Sivananthan

Abstract

In this paper we present a new scheme of a kernel-based regularization learning algorithm, in which the kernel and the regularization parameter are adaptively chosen on the base of previous experience with similar learning tasks. The construction of such scheme is motivated by the problem of prediction of the blood glucose levels of diabetic patients. We describe how the proposed scheme can be used for this problem and report the results of the tests with real clinical data as well as compare them with existing literature.

1. Introduction

In this paper we present a meta-learning approach to choosing the kernels and regularization parameters in regularized kernel-based learning algorithms. The concept of meta-learning presupposes that above-mentioned components of the algorithms are selected on the base of previous experience with similar learning tasks. Therefore, selection rules developed in this way are intrinsically problem-oriented. In this paper we demonstrate the proposed meta-learning approach on a problem from diabetes technology, but it will be also seen how its main ingredients (e.g., Theorem 1) can be exploited in other applications.

The massive increase in the incidence of diabetes is now a major global healthcare challenge, and the treatment of diabetes is one of the most complicated therapies to manage, because of the difficulty in predicting blood glucose (BG) levels of diabetic patients.

Recent progress in diabetes technology is related to the so-called Continuous Glucose Monitoring (CGM) systems which provide, almost in real-time, an indirect estimation of current blood glucose that is highly valuable for the insulin therapy of diabetes [16]. However, it would be much more preferable to use CGM for predicting dangerous episodes of hypo- and hyperglycemia, when BG-concentration goes outside the normal range (70-180 mg/dL).

In its simplest form, diabetes therapy is based on rules that are used to estimate the necessary amount of insulin injection to prevent hyperglycemia or possibly of additional snacks to prevent hypoglycemia. Keeping in mind [40] that the onset of insulin occurs within 10-30 minutes, and the onset of meal responses on glucose levels occurs approximately within 5-10 minutes, it is important to know future BG-level at least 10-30 minutes ahead of time.

On the other hand, it should be noted that CGM technologies report interstitial glucose (IG) concentration, and a time lag of approximately 10-15 minutes exists between real BG-concentrations and IG-values obtained via CGM [18]. Therefore, to mitigate effects of this time lag and increase therapeutic benefit, a prediction of glucose with a prediction horizon (*PH*) of 60-75 minutes is also of great interest, especially for automation of glucose control [30].

From the literature we know that nowadays there are mainly two approaches to predict the future blood glucose based upon patient's current and past blood glucose values. One of them uses the time-series methodology [13, 28, 35, 43], while another one employes artificial neural networks techniques [29, 30, 33].

But time-series predictors seem to be too sensitive to gaps in the data, which may frequently appear when available blood glucose meters are used. As to neural networks predictors, they need long training periods and much more information to be set up.

Therefore, in this paper we describe a novel approach that is based on the idea to use the regularized learning algorithms in predicting blood glucose. These algorithms are well understood now [2, 8, 9, 14, 19], and it is known that their performance essentially depends on the choice of the regularization parameters and, which is even more important, on the choice of the kernels generating Reproducing Kernel Hilbert Spaces (RKHS), in which the regularization is performed [5, 10, 22, 26, 42]. As it was realized [31], in the context of blood glucose prediction these algorithmic instances cannot be a priori fixed, but need to be adjusted to each particular prediction input.

Thus, a regularized learning based predictor should learn how to learn kernels and regularization parameters from input. Such a predictor is constructed as a result of a process of learning to learn, or "meta-learning" [37]. In this way we have developed the Fully Adaptive Regularized Learning (FARL) approach to the blood glucose prediction. This approach is described in the patent application [32] filed jointly by Austrian Academy of Sciences and Novo Nordisk A/S (Denmark). The developed approach allows the construction of blood glucose predictors which, as it has been demonstrated in the extensive clinical trials, outperform the state-of-the-art algorithms. Moreover, it turns out that in the context of the blood glucose prediction the FARL approach is more advanced than other meta-learning technologies such as k-Nearest Neighbors (k-NN) ranking [41].

To facilitate further discussion, this paper is structured into 4 additional sections. Section 2 explains the details of the regularized learning approach to BG-prediction and indicates its issues and concerns. Section 3 specifies the framework of meta-learning for kernel-based regularized learning algorithms and three different types of operations required for performing meta-learning, in particular, how these operations are processed within the proposed FARL approach. In Section 4 we present a performance comparison of the FARL-based predictors with the current state-of-the-art BG-prediction methods and k-NN meta-learning. The paper concludes with Section 5 on current and future developments.

2. A Traditional Learning Theory Approach: Issues and Concerns

Throughout this paper we consider the problem of blood glucose prediction. Mathematically this problem can be formulated as follows. Assume that at the time moment $t = t_0$ we are given m preceding estimates $g_0, g_{-1}, g_{-2}, \dots, g_{-m+1}$ of a patient's BG-concentration sampled correspondingly at the time moments $t_0 > t_{-1} > t_{-2} > \dots > t_{-m+1}$ within the sampling horizon $SH = t_0 - t_{-m+1}$. The goal is to construct a predictor that uses these past measurements to predict BG-concentration as a function of time $g = g(t)$ for n subsequent future time moments $\{t_j\}_{j=1}^n$ within the prediction horizon $PH = t_n - t_0$ such that $t_0 < t_1 < t_2 < \dots < t_n$.

At this point, it is noteworthy to mention that CGM systems provide estimations $\{g_i\}$ of BG-values every 5 or 10 minutes, such that $t_i = t_0 + i\Delta t,\ i = -1, -2, \dots$, where $\Delta t = 5$ (min) or $\Delta t = 10$ (min). For mathematical details see [26].

Thus, the promising concept in the diabetes therapy management is the prediction of the future BG-evolution using CGM data [39]. The importance of such prediction has been shown by several applications [4, 28].

From the above discussion, one can see that the CGM technology allows us to form a training set $\mathbf{z} = \{(x_\mu, y_\mu), \mu = 1, 2, \dots, M\}$, $|\mathbf{z}| = M$, where

$$\begin{array}{rcl} x_\mu & = & ((t^\mu_{-m+1}, g^\mu_{-m+1}), \dots, (t^\mu_0, g^\mu_0)) \in (\mathbb{R}^2_+)^m, \\ y_\mu & = & ((t^\mu_1, g^\mu_1), \dots, (t^\mu_n, g^\mu_n)) \in (\mathbb{R}^2_+)^n, \\ \text{and} & & t^\mu_{-m+1} < t^\mu_{-m+2} < \dots < t^\mu_0 < t^\mu_1 < \dots < t^\mu_n \end{array} \tag{1}$$

are the moments at which patient's BG-concentrations were estimated by CGM system as $g^\mu_{-m+1}, \dots, g^\mu_0, \dots, g^\mu_n$. Moreover, for any $\mu = 1, 2, \dots, M$ the moments $\{t^\mu_j\}_{j=-m+1}^n$ can be chosen such that $t^\mu_0 - t^\mu_{-m+1} = SH,\ t^\mu_n - t^\mu_0 = PH$, where SH and PH are the sampling and prediction horizons of interest respectively.

Given a training set it is rather natural to consider our problem in the framework of supervised learning [8, 14, 19, 38, 45], where the available input-output samples (x_μ, y_μ) are assumed to be drawn independently and identically distributed (i.i.d.) according to an unknown probability distribution. Originally, in [17] it is stated that the consecutive CGM readings $\{g_i\}$ taken from the same subject within a relatively short time are highly interdependent. At the same time, CGM readings that are separated by more than 1 hour in time could be considered as (linearly) independent. Therefore, using the supervised learning framework we are forced to consider vector-valued input-output relations $x_\mu \to y_\mu$ instead of scalar-valued ones $t^\mu_i \to g^\mu_i$. Moreover, we will assume that $(t^\mu_i, g^\mu_i), \mu = 1, 2, \dots, M$, are sampled in such a way that $|t^\mu_i - t^{\mu+1}_i| > 1$ (hour).

In this setting, a set $\mathbf{z}$ is used to find (a vector-valued) function $f_{\mathbf{z}} : (\mathbb{R}_+^2)^m \to (\mathbb{R}_+^2)^n$ such that for any new BG-observations

$$x = ((t_{-m+1}, g_{-m+1}), \ldots, (t_0, g_0)) \in (\mathbb{R}_+^2)^m \tag{2}$$

with $t_{-m+1} < t_{-m+2} < \ldots < t_0$, $t_0 - t_{-m+1} = SH$, the value $f_{\mathbf{z}}(x) \in (\mathbb{R}_+^2)^n$ is a good prediction of the future BG-sample

$$y = ((t_1, g_1), \ldots, (t_n, g_n)) \in (\mathbb{R}_+^2)^n, \tag{3}$$

where $t_0 < t_1 < \ldots < t_n$, $t_n - t_0 = PH$.

Note that in such vector-valued formulation the problem still can be studied with the use of the standard scheme of supervised learning [9, 23], where it is assumed that $f_{\mathbf{z}}$ belongs to an RKHS $\mathcal{H}_K$ generated by a kernel K.

Then $f_{\mathbf{z}} = f_{\mathbf{z}}^\lambda \in \mathcal{H}_K$ is constructed as the minimizer of the functional

$$\frac{1}{|\mathbf{z}|} \sum_{\mu=1}^{|\mathbf{z}|} \|f(x_\mu) - y_\mu\|^2_{(\mathbb{R}^2)^n} + \lambda \|f\|^2_{\mathcal{H}_K}, \tag{4}$$

where λ is a regularization parameter.

Recall [9, 23] that a Hilbert space $\mathcal{H}$ of vector-valued functions $f : X \to (\mathbb{R}^2)^n$, $X \subset (\mathbb{R}^2)^m$, is called an RKHS if for any $x \in X$ the value $f(x)$ admits a representation $f(x) = K_x^* f$, where $K_x^* : \mathcal{H} \to (\mathbb{R}^2)^n$ is a Hilbert-Schmidt operator, which is the adjoint of $K_x : (\mathbb{R}^2)^n \to \mathcal{H}$. Similar to the scalar case the inner product $\langle \cdot, \cdot \rangle = \langle \cdot, \cdot \rangle_K$ can be defined in terms of the kernel $K(x,t) = K_x^* K_t$ for every $x, t \in X$.

The standard scheme (4) raises two main issues that should be clarified before its usage. One of them is how to choose a regularization parameter λ and another one, which is even more important, is how to choose the space $\mathcal{H}_K$, where the regularization should be performed, or, which is the same thing, the kernel K that generates this space. Several approaches to address these issues have been proposed in the past few years [5, 10, 20, 22, 36, 47].

All of them attempt to choose a kernel K "globally" for the whole given training set $\mathbf{z}$, but they do not account for particular features of input x_μ. As the result, if some new input-output pair (x_μ, y_μ) is added to the training set $\mathbf{z}$, then, in accordance with known approaches, a kernel selection procedure should be started from scratch, which is rather costly. In essence, known techniques [5, 10, 20, 22, 47] do not learn how to select a kernel K and a regularization parameter λ for each new input x in question.

In the next section we introduce a meta-learning approach which is free from the above-mentioned shortcoming and allows us to adjust K and λ "locally" to each new input x on the basis of the previous learning experience with the examples (x_μ, y_μ) from a given training set $\mathbf{z}$.

3. Meta-Learning Approach to Choosing a Kernel and a Regularization Parameter

First of all, let us note that the choice of the regularization parameter λ is completely depends on the choice of the kernel. For the fixed kernel K, there is a variety of strategies that can be used to select a regularization parameter λ. Among them are the discrepancy principle [24, 25, 34], balancing principle [10, 21], heuristically motivated quasi-optimality criterion [15, 44]. Thus, keeping in mind this remark, we will think about λ as a functional of K, i.e. $\lambda = \lambda(K)$.

This observation motivates us to focus mainly on the choice of the kernel K as it can make significant difference in performance [3, Section 2.4].

As we already mentioned in the previous section, in most of the known approaches [5, 10, 20, 22, 47] the chosen kernel K and the regularization parameter λ are "reasonable," in some sense, for the whole training set $\mathbf{z} = \{(x_\mu, y_\mu)\}$, but they are not necessarily optimal for a particular pair $(x_\mu, y_\mu) \in \mathbf{z}$. In this section, as a way to overcome this drawback, we describe our approach to the kernel choice problem, which is based on the concept of meta-learning.

According to this approach, the meta-learning process can be divided into three phases / operations.

In the first phase, which can be called optimization, the aim is to find for each input-output pair (x_μ, y_μ), $\mu = 1, 2, \ldots, M$, a favorite kernel $K = K^\mu$ and a regularization parameter $\lambda = \lambda_\mu$, which in some sense optimize a prediction of y_μ from x_μ. This operation can be cast as the set of M search problems, where for each pair (x_μ, y_μ) we are searching over some set of admissible kernels.

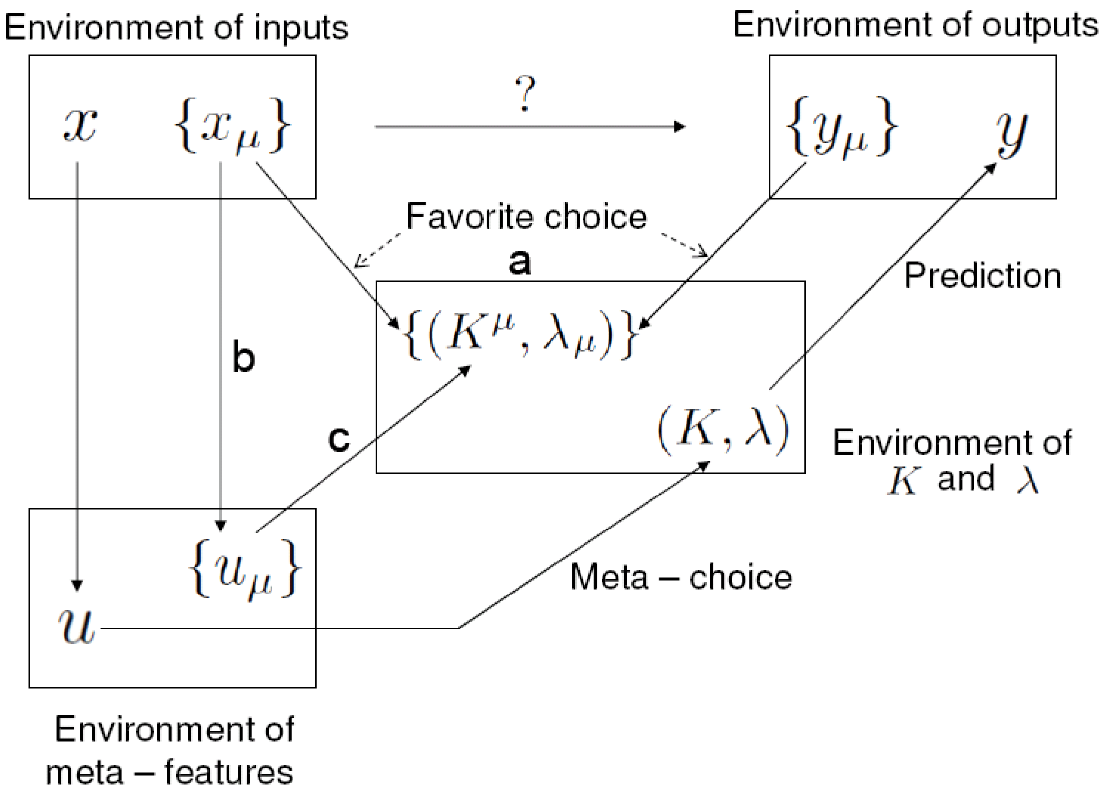


Figure 1: Meta-learning approach to choosing K and λ for the regularized kernel-based prediction: Optimization phase (a-arrows), Meta-features choice (b-arrow), Learning at meta-level (c-arrow) and Meta-choice of (K, λ) for prediction

Note that in the usual learning setting a kernel is also sometimes found as the solution of some optimization operation [5, 10, 20, 22, 47], but in contrast to our meta-learning based approach, the problem is formulated for the whole training set. As a result, such a kernel choice should be executed from scratch each time when a new input-output pair (x_μ, y_μ) is added to the training set. Moreover, as it was already mentioned several times, the kernel chosen in this way is not necessarily optimal for a particular input-output pair.

The second phase of our meta-learning based approach consists in choosing and computing the so-called meta-features $\{u_\mu\}$ of inputs $\{x_\mu\}$ from the training set. The design of adequate meta-features should capture and represent the properties of an input x_μ that influence the choice of a favorite kernel K^μ used for predicting y_μ from x_μ. This second phase of meta-learning is often driven by heuristics [3, Section 3.3]. In [41] the authors discuss a set of 14 possible input characteristics, which can be used as meta-features. In our approach, we use one of them, namely a two-dimensional vector $u_\mu = (u_\mu^{(1)}, u_\mu^{(2)})$ of the coefficients of "least squares regression line" $g^{lin} = u_\mu^{(1)} t + u_\mu^{(2)}$ that produces the "best linear fit" linking the components $t = (t_{-m+1}^\mu, t_{-m+2}^\mu, \ldots, t_0^\mu)$ and $g = (g_{-m+1}^\mu, g_{-m+2}^\mu, \ldots, g_0^\mu)$, which form the input x_μ. Heuristic reason for choosing such a meta-feature will be given below.

Note that in the present context one may, in principle, choose an input x_μ itself as a meta-feature. But, as it will be seen below, such a choice would essentially increase the dimensionality of the optimization problem in the final phase of the meta-learning. Moreover, since the inputs x_μ are formed by potentially noisy measurements (t_i^μ, g_i^μ), the use of low dimensional meta-features $u_\mu = (u_\mu^{(1)}, u_\mu^{(2)})$ can be seen as a regularization (denoising) by dimension reduction and as an overfitting prevention.

The final phase of the meta-learning consists of constructing the so-called meta-choice rule that explains the relation between the set of meta-features of inputs and the parameters of favorite algorithms found in the first phase of the meta-learning (optimization). This phase is sometimes called learning at meta-level. If above mentioned meta-choice rule is constructed, then for any given input x the parameters of a favorite prediction algorithm can be easily found by applying this rule to the meta-feature u calculated for the input x in question.

Recall that in the present context, the first two phases of the meta-learning result in the transformation of the original training set $\mathbf{z} = \{(x_\mu, y_\mu)\}$ into new ones, where the meta-features u_μ are paired with the parameters of favorite kernels K^μ and $\lambda_\mu = \lambda(K^\mu)$.

Then, in principle, any learning algorithm can be employed on these new training sets to predict the parameters of the favorite kernel K and $\lambda = \lambda(K)$ for the input x in question. For example, in [36] these parameters are predicted by means of least squares method that is performed in RKHS generated by the so-called histogram kernel. Note that such approach can be used only for sufficiently simple sets of admissible kernels $\mathcal{K}$ (only linear combinations of some a priori fixed kernels are considered in [36]). Moreover, as it has been also noted by the authors [36], in general, the histogram kernel does not take into account specific knowledge about the problem at hand. So, the approach [36] can only loosely be considered as a meta-learning.

At the same time, one of the most popular algorithm suggested in the meta-learning literature for learning at meta-level is the so-called k-Nearest Neighbors (k-NN) ranking [3, 41]. The algorithm can be interpreted as a learning in the space of piecewise constant functions.

One of the novelties of our approach is that a regularization in RKHS is used not only in the first phase, but also in learning at meta-level. Of course, in this case the kernel choice issue arises again, and it will be addressed in the same manner as in the first phase. But, what is important, the corresponding optimization needs to be performed only once and only with the transformed training set (u_μ, K^μ) from just one patient. This means that the blood glucose predictor based on our approach can be transported from patient to patient without any additional re-adjustment. Such a portability is desirable and will be demonstrated in experiments with real clinical data. Moreover, it will be shown that the use of k-NN ranking at meta-level results in the blood glucose predictor, which is outperformed by the predictor based on our approach.

In general, the meta-learning approach is schematically illustrated in Figure 1. The following subsections contain detailed description of all the operations needed to install and set our meta-learning based predictor.

3.1. Optimization operation

The ultimate goal of the optimization operation is to select such kernel K and regularization parameter λ that allow to achieve good performance for given data. To describe the choice of favorite K and λ for each input-output pair $(x_\mu, y_\mu) \in (\mathbb{R}_+^2)^m \times (\mathbb{R}_+^2)^n$ from the training set $\mathbf{z}$ we rephrase vector-valued formalism in terms of ordinary scalar-valued functions similar to how it was done in [9]. Moreover, we will describe the optimization operation in general terms, since, as it has been mentioned above, in our approach this operation should be performed at the first and at the last phases of meta-learning. As a result, a nature of training sets of input-output pairs involved in the optimization process will be different at different phases.

Let input and output environments U and V be compact sets in $\mathbb{R}^d$ and $\mathbb{R}$ respectively.

Let us also assume that we are given two sets of input-output pairs $W_1, W_2 \subset U \times V$ governed by the same input-output relation. The first set can be used for constructing regularized approximations of the form

$$F_\lambda = F_\lambda(\cdot; K, W_1) = \arg\min T_\lambda(f; K, W_1), \tag{5}$$

$$T_\lambda(f; K, W_1) = \frac{1}{|W_1|} \sum_{(u_i,v_i)\in W_1} |f(u_i) - v_i|^2 + \lambda \|f\|^2_{\mathcal{H}_K}, \tag{6}$$

where K is a kernel defined on U, and, as before, λ is a regularization parameter, which is chosen in dependence on K, so that we can write $\lambda = \lambda(K)$ and

$$F_\lambda = F_{\lambda(K)}(\cdot; K, W_1) = \sum_{(u_i,v_i)\in W_1} c_i^\lambda K(\cdot, u_i).$$

Due to the Representer Theorem [46], a real vector $\mathbf{c}^\lambda = (c_i^\lambda)$ of coefficients is defined as $\mathbf{c}^\lambda = (\lambda|W_1|\mathbb{I} + \mathbb{K})^{-1}\mathbf{v}$, here $\mathbf{v} = (v_i)$ and $\mathbb{K} = (K(u_i, u_j))$, $\mathbb{I}$ are the corresponding Gramm matrix and the unit matrix of the size $|W_1| \times |W_1|$ respectively.

The second set W_2 is used for estimating the performance of a particular approximation F_λ, which is measured by the value of the functional

$$P(F_\lambda; W_2) = \frac{1}{|W_2|} \sum_{(u_i,v_i)\in W_2} \rho(F_\lambda(u_i), v_i), \tag{7}$$

where $\rho(\cdot,\cdot)$ is a continuous function of two variables. We note that the function $\rho(\cdot,\cdot)$ can be adjusted to the intended use of the approximations F_λ.

Finally, we choose our favorite K^0 and λ^0 as minimizers of the functional

$$Q_\theta(K, \lambda, W_1, W_2) = \theta T_\lambda(F_\lambda(\cdot; K, W_1); K, W_1) + (1-\theta) P(F_\lambda(\cdot; K, W_1); W_2) \tag{8}$$

over a given set of admissible kernels $\mathcal{K}$ and over an interval $[\lambda_{\min}, \lambda_{\max}]$ of possible $\lambda-$values. Note that the parameter θ here takes the values from $[0, 1]$ and can be seen as a performance regulator on the sets W_1 and W_2. Taking $\theta > \frac{1}{2}$, we put more emphasize on the ability to mimic the input data from W_1, while for θ closer to zero, we are more interested in making a generalization from those data. The minimization of the functional (8) is performed in the first and the last phases of the meta-learning. In the first case we minimize (8) with $\theta = 0$, while in the second case we put $\theta = \frac{1}{2}$.

Now we formulate the main result of this subsection justifying existence of the kernel K^0 and the regularization parameter λ^0 that minimize the functional (8).

Theorem 1. (Kernel Choice Theorem [26])

Let $\mathcal{K}(U)$ be the set of all kernels defined on $U \subset \mathbb{R}^d$. Let also Ω be a compact metric space and $G : \Omega \to \mathcal{K}(U)$ be a continuous map in the sense that for any $u, \hat{u} \in U$ the function

$$\omega \mapsto K_\omega(u, \hat{u}) \in \mathbb{R}$$

is continuous on Ω, where for $\omega \in \Omega$ the kernel $K_\omega \in \mathcal{K}(U)$ is given as $K_\omega = G(\omega)$ and $K_\omega(u, \hat{u})$ is the value of the kernel $K_\omega \in \mathcal{K}(U)$ at $u, \hat{u} \in U$.

Define

$$\mathcal{K} = \mathcal{K}(\Omega, G) = \{K : K = G(\omega),\ K \in \mathcal{K}(U),\ \omega \in \Omega\}$$

be the set of kernels parameterized via G by elements of Ω.

Then for any parameter choice rule $\lambda = \lambda(K) \in [\lambda_{\min}, \lambda_{\max}]$, $\lambda_{\min} > 0$ there are $K^0 = K^0(W_1, W_2)$ and $\lambda^0 \in [\lambda_{\min}, \lambda_{\max}]$ such that

$$Q_\theta(K^0, \lambda^0, W_1, W_2) = \inf\{Q_\theta(K, \lambda(K), W_1, W_2),\ K \in \mathcal{K}(\Omega, G)\}.$$

Note that, as it has been pointed out in [26], in contrast to usual approaches, the technique described by the Theorem 1 is more oriented towards the prediction of the value of unknown function outside of the scope of available data. For example, in [22] it has been suggested to choose the kernel $\overline{K} = \overline{K}(W, \lambda)$ as the minimizer of the functional $T_\lambda(F_\lambda(\cdot; K, W); K, W)$, where $W = W_1 \cup W_2$ and λ is given a priori.

Thus, the idea of [22] is to recover the kernel $\overline{K}$ generating the space where the unknown function of interest lives from given data, and then use this kernel for constructing the predictor $F_\lambda(\cdot; \overline{K}, W)$.

Although feasible, this approach may fail in the prediction outside of the scope of available data as it was shown in [26]. In contrast, the approximant F_λ based on the kernel chosen in accordance with the Theorem 1 exhibits good prediction properties, see [26] for more details.

To illustrate the assumptions of the Kernel Choice Theorem 1, we consider two cases, which are needed to set up our meta-learning predictor. In both cases the quasi-balancing principle [10] is used as a parameter choice rule $\lambda = \lambda(K) \in [10^{-4}, 1]$.

In the first case, we use the data (1), and for any $\mu = 1, 2, \dots, M$ define the sets

$$W_1 = W_{1,\mu} = x_\mu = ((t^\mu_{-m+1}, g^\mu_{-m+1}), \dots, (t^\mu_0, g^\mu_0)), \qquad t^\mu_0 - t^\mu_{-m+1} = SH,$$
$$W_2 = W_{2,\mu} = y_\mu = ((t^\mu_1, g^\mu_1), \dots, (t^\mu_n, g^\mu_n)), \qquad t^\mu_n - t^\mu_0 = PH.$$

In this case, the input environment U is assumed to be a time interval, i.e. $U \subset (0, \infty)$, while the output environment $V = [0, 450]$ is the range of possible BG-values (in mg/dL).

For this case, we choose $\Omega = \{\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3, \omega_i \in [10^{-4}, 3], i = 1, 2, 3\}$, and the set of admissible kernels is chosen as

$$\mathcal{K}(\Omega, G) = \{K : K(t, \tau) = (t\tau)^{\omega_1} + \omega_2 e^{-\omega_3 (t-\tau)^2}, \quad (\omega_1, \omega_2, \omega_3) \in \Omega\}. \tag{9}$$

For such a choice, the continuous map G parametrizing the admissible kernels is defined as $G : \omega = (\omega_1, \omega_2, \omega_3) \to K_\omega(t, \tau) = (t\tau)^{\omega_1} + \omega_2 e^{-\omega_3 (t-\tau)^2}$, where $t, \tau \in U$. It is easy to see that for any $\omega = (\omega_1, \omega_2, \omega_3) \in [10^{-4}, 3]^3$, the kernel $K_\omega(t, \tau) = G(\omega)(t, \tau)$ is positive definite and for any fixed $t, \tau \in U$ its value continuously depends on ω.

To apply the Theorem 1 in this case, we modify the functional $P(\cdot, W_{2,\mu})$ involved in the representation of (8) as in [27] with the idea to penalize heavily the failure in detection of dangerous glycemic levels.

As a result of the application of the Theorem 1, we relate input-output BG-observations (x_μ, y_μ) to the parameters $\omega^0 = \omega^0_\mu = (\omega^0_{1,\mu}, \omega^0_{2,\mu}, \omega^0_{3,\mu})$ of our favorite kernels $K^0 = K^{0,\mu} = K_{\omega^0_\mu}$ and $\lambda_\mu = \lambda^0_\mu$. As we already mentioned, the corresponding optimization is executed only for the data set of one particular patient. Thus, the operation in this case does not require considerable computational effort and time.

The second case of the use of the Theorem 1 corresponds to the optimization, that should be performed at the final phase of the meta-learning. We consider the transformed data sets $\mathbf{z}_i = \{(u_\mu, \omega^0_{i,\mu}), \ \mu = 1, 2, \dots, M\}, \ i = 1, 2, 3,$ obtained after performing the first two meta-learning operations.

In this case the input environment U is formed by two-dimensional meta-features vectors $u_\mu \in \mathbb{R}^2$ computed for the inputs x_μ, i.e. $U \subset \mathbb{R}^2$, whereas the output environment $V = [10^{-4}, 3]$ is the range of parameters ω_i of the kernels from (9).

Recall that at the final meta-learning phase the goal is to assign the parameters $\omega^0 = (\omega^0_1, \omega^0_2, \omega^0_3)$, λ^0 of favorite algorithm to each particular input x, and such assignment should be made by comparing the meta-feature u calculated for x with the meta-features u_μ of inputs x_μ, for which the favorite parameters have been already found at the first meta-learning phase.

In the meta-learning literature one usually makes the above mentioned comparison by using some distance between meta-feature vectors u and u_μ. For two-dimensional meta-features $u = (u^{(1)}, u^{(2)})$, $u_\mu = (u^{(1)}_\mu, u^{(2)}_\mu)$ one of the natural distances is the weighted Euclidean distance

$$|u - u_\mu|_\gamma := (\gamma_1 (u^{(1)} - u^{(1)}_\mu)^2 + \gamma_2 (u^{(2)} - u^{(2)}_\mu)^2)^{\frac{1}{2}}$$

that potentially may be used in the meta-learning ranking methods in the same way as the distance suggested in [41] (see also Section 3.3 below). Here we refine this approach by learning the dependence of parameters λ^0, ω^0_i, $i = 1, 2, 3$, on the meta-feature u in the form of functions

$$F(u) = \sum_{\mu=1}^{M} c_\mu \varphi_\omega(|u - u_\mu|_\gamma),$$

where $\omega = (\omega_1, \omega_2, \omega_3, \omega_4) \in \Omega = [0,2] \times [0,15] \times [0,2] \times [0,15]$, $\varphi_\omega(\tau) = \tau^{\omega_1} + \omega_2 e^{-\omega_3 \tau^{\omega_4}}$, and corresponding coefficients c_μ for $\lambda^0, \omega_i^0,\ i = 1,2,3$, are defined in accordance with the formula (15) below.

It means that the final meta-learning phase can be implemented as the optimization procedure described in the Theorem 1, where the set of admissible kernels is chosen as follows

$$\mathcal{K} = \mathcal{K}_\gamma(\Omega, G) = \{K : K_{\omega,\gamma}(u, \hat{u}) = M^{-1} \sum_{\mu=1}^{M} \varphi_\omega(|u - u_\mu|_\gamma)\varphi_\omega(|\hat{u} - u_\mu|_\gamma),\ \omega \in \Omega\}. \tag{10}$$

It is clear that the conditions of the Theorem 1 are satisfied with this choice.

To apply the optimization procedure above, we rearrange the sets $\mathbf{z}_i$, so that $\mathbf{z}_i = \{(u_{\mu_k}, \omega^0_{i,\mu_k})\}$, where $\omega^0_{i,\mu_k} < \omega^0_{i,\mu_{k+1}}$, $k = 1, 2 \ldots, M-1$, and define the sets W_1, W_2 as follows:

$$W_1 = W_{1,i} = \{(u_{\mu_k}, \omega^0_{i,\mu_k}),\ k = 3, \ldots, M-2\}, \quad W_2 = W_{2,i} = \mathbf{z}_i \setminus W_{1,i},$$

so that the performance estimation sets $W_2 = W_{2,i}$ contain two smallest and two largest values of the corresponding parameters.

Moreover, for the considered case we use the functional (7) with $\rho(f, v) = |f - v|^2$.

Then for $i = 1,2,3$, using the optimization procedure described in the Theorem 1 one can find the kernels $K^0 = K_i^0 \in \mathcal{K}_\gamma(\Omega, G)$ determined by the values of parameters that are presented in Table 1. In addition, using in the same way the set $\{(u_\mu, \lambda_\mu^0)\}$ one can obtain the kernel $K_4^0 \in \mathcal{K}_\gamma(\Omega, G)$ which parameters are also given in Table 1.

Table1. The parameters of the kernels from (10), which are selected for learning at meta-level

	γ_1	γ_2	ω_1	ω_2	ω_3	ω_4
K_1^0	1	0	1.6	5	0.001	0.016
K_2^0	1	0	1.2	0.001	3	0.01
K_3^0	1	0	0	1	0.001	0.003
K_4^0	1	1	0.2	0.02	0.1	0.2

Summing up, as the result of the optimization operations we, at first, find for each input-output pair (x_μ, y_μ), $\mu = 1, 2, \ldots, M$, the parameters of the favorite kernel $K^0 = K^{0,\mu}$ from (9) and $\lambda^0 = \lambda_\mu^0 \in [10^{-4}, 1]$. Then using these found parameters we construct the kernels $K^0 = K_i^0$, $i = 1,2,3,4$, from (10) that will relate $(K^{0,\mu}, \lambda_\mu^0)$ with corresponding meta-features u_μ.

In both cases the minimization of the corresponding functionals (8) was performed by a full search over grids of parameters ω determining the kernels from (9) and (10). Of course, the application of the full search method is computationally intensive, but, as we already mentioned, in our application this minimization procedure should be performed only once and only for one particular patient.

Remark 1. *From the above discussion, it is obvious that the approach described in the Theorem 1 requires to split available data into two sets of input-output pairs $W_1, W_2 \subset U \times V$. Note that in the recent paper [36] data splitting has been also used for identifying the favorite kernel from the set of admissible ones. In our terms, the approach [36] suggests to choose the kernel as follows*

$$K^0 = \arg\min_{K \in \mathcal{K}} T_\lambda(F_\lambda(\cdot; K, W_1); K, W_1 \cup W2), \tag{11}$$

where in contrast to the Theorem 1, the value of the regularization parameter λ is assumed to be a priori given.

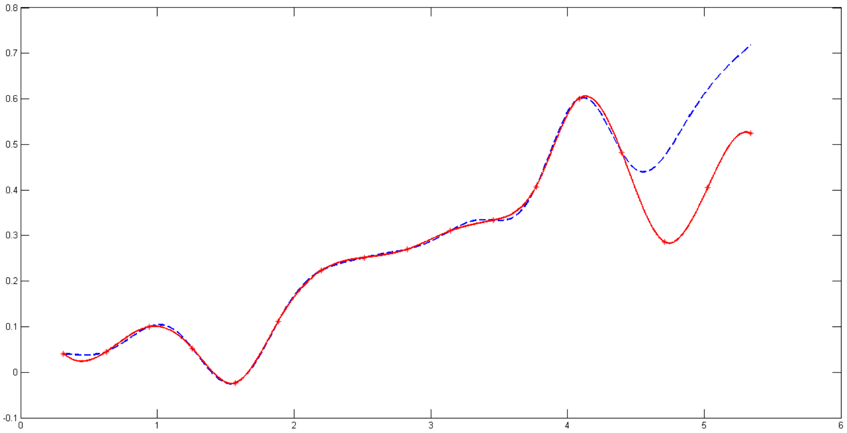


Figure 2: The performance of the approximant $F_\lambda(\cdot; K^0, W_1 \cup W_2)$ (dotted line) based on the kernel $K^0(u, \hat{u}) = (u\hat{u})^{1.74} + 1.26e^{-5.54(u-\hat{u})^2}$, chosen in accordance with (11) for $\lambda = 10^{-4}$

Using the same example as in [10, 22, 26], one can show that the approach (11) may not be suitable for the prediction outside of the scope of available data. Indeed, following [22], we consider the target function

$$f(u) = 0.1\left(u + 2\left\{e^{-8\left(\frac{4\pi}{3}-u\right)^2} - e^{-8\left(\frac{\pi}{2}-u\right)^2} - e^{-8\left(\frac{3\pi}{2}-u\right)^2}\right\}\right) \tag{12}$$

and the training set $\mathbf{z} = \{(u_i, v_i), i = 1, 2, \ldots, 14\}$ *consisting of points* $u_i = \frac{\pi i}{10}$ *and* $v_i = f(u_i) + \xi_i$, *where* ξ_i *are random values sampled uniformly in the interval* $[-0.02, 0.02]$. *Note that the function (12) belongs to an RKHS generated by the kernel* $\overline{K}(u, \hat{u}) = u\hat{u} + e^{-8(u-\hat{u})^2}$, *and we are interested in the reconstruction of its values for* $u > 1.4\pi$, *i.e. outside of the scope of available data.*

To illustrate the approach (11), at first, we define the sets W_1, W_2 *similar to [26] :*

$$W_1 = \{(u_i, v_i),\ i = 1, 2, \ldots, 7\}, \quad W_2 = \mathbf{z} \setminus W_1 = \{(u_i, v_i),\ i = 8, 9, \ldots, 14\}. \tag{13}$$

In our experiment, we explore the influence of the regularization parameter λ *on the performance of the approximation* $F_\lambda(\cdot; K^0, W_1 \cup W_2)$ *with the kernel* K^0 *chosen in accordance with (11) for several* λ *fixed independently of* K. *The favorite kernel* K^0 *is chosen from the set (9) with* $\Omega = \{\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3,\ \omega_1, \omega_3 \in [10^{-4}, 3],\ \omega_2 \in [10^{-4}, 8]\}$. *Note that this set contains the kernel* $\overline{K}$ *generating the target function (12). Here, as in [22], the value of the regularization parameter* λ *is taken from the set* $\{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1\}$.

It is instructive to note that for all considered values of the regularization parameter λ *the approximants based on the kernels (11) do not allow an accurate reconstruction of the values of the target function* $f(u)$ *for* $u > 1.4\pi$. *Typical examples are shown in Figures 2 and 3.*

At the same time, from [26] we know that the approximant $F_{\lambda(K^0)}(\cdot; K^0, W_1 \cup W_2)$ *based on the kernel* K^0 *chosen in accordance with the Theorem 1 provides us with an accurate reconstruction of* $f(u)$ *for* $u > 1.4\pi$.

3.2. Heuristic operation

The goal of this operation is to extract special characteristics $\{u_\mu\}$ called meta-features of inputs $\{x_\mu\}$ that can be used for explaining the relation between $\{x_\mu\}$ and the parameters of optimal algorithms predicting training outputs $\{y_\mu\}$ from $\{x_\mu\}$. Note that it is common belief [3, Section 3.3] that such meta-features should reflect the nature of the problem to be solved.

Keeping in mind that practically all predictions of the future blood glucose concentration are currently based on a linear extrapolation of glucose values [17], it seems to be natural to consider the vector $u_\mu = (u_\mu^{(1)}, u_\mu^{(2)})$ of

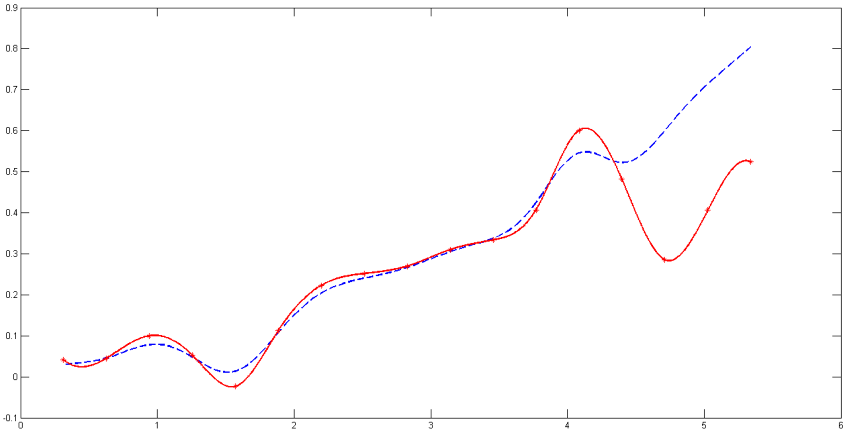


Figure 3: The performance of the approximant $F_\lambda(\cdot; K^0, W_1 \cup W_2)$ (dotted line) based on the kernel $K^0(u, \hat{u}) = (u\hat{u})^{1.89} + 3e^{-8(u-\hat{u})^2}$, chosen in accordance with (11) for $\lambda = 0.1$

coefficients of a linear extrapolator $g_\mu^{lin}(t) = u_\mu^{(1)}t + u_\mu^{(2)}$, producing the best linear fit for given input data $x_\mu = ((t^\mu_{-m+1}, g^\mu_{-m+1}), \cdots, (t^\mu_0, g^\mu_0))$, as a good candidate for being a meta-feature of x_μ.

Then for any given input $x = ((t_{-m+1}, g_{-m+1}), \cdots, (t_0, g_0))$ the components of the corresponding meta-feature $u = (u^{(1)}, u^{(2)})$ are determined by the linear least squares fit as follows

$$u^{(1)} = \sum_{i=-m+1}^{0} \frac{(t_i - \bar{t})(g_i - \bar{g})}{\sum\limits_{i=-m+1}^{0} (t_i - \bar{t})^2}, \qquad u^{(2)} = \bar{g} - u^{(1)}\bar{t}, \tag{14}$$

here $\bar{a}$ is an average.

Note that in principle the linear extrapolator $g^{lin}(t) = u^{(1)}t + u^{(2)}$ can be used for predicting the future BG-concentration from x. But, as it can been seen from [39], for prediction horizons of clinical interest ($PH > 10$ min) such a predictor is outperformed by more sophisticated algorithms. Therefore, we are going to use the coefficient vector $u = (u^{(1)}, u^{(2)})$ only as a meta-feature (lable) of the corresponding prediction input.

3.3. *Learning at Meta-level*

The goal of the final phase of the meta-learning approach, which is also called learning at meta-level, is the construction of the so-called meta-choice rule for selecting the vector $\omega = (\omega_1, \omega_2, \omega_3)$ of the parameters of favorite algorithm that will be applied to input x in question labeled by a meta-feature u. Recall, at this stage the above mentioned meta-choice rule is constructed on the basis of the transformed training sets $\mathbf{z}_i = \{(u_\mu, \omega^0_{i,\mu})\}$, $i = 1, 2, 3$.

In this section, we describe two meta-choice rules. The first one, the k-Nearest Neighbors (k-NN) ranking, is one of the most popular methods in meta-learning literature. This method has been suggested in [41] and the idea behind it is to identify a set of k meta-features $\{u_\mu\}$ containing the ones that are most similar to the considered meta-feature u, and then combine the corresponding $\{\omega^0_\mu\}$ to select the vector ω for the new input x. In their numerical experiments the authors [41] observed the clear tendency for the accuracy of k-NN ranking to decrease with increasing number of k neighbors. Therefore, we consider only 1-NN ranking method as one that produces more accurate results than other k-NN rankings.

Using [41] we describe how the 1-NN ranking can be adjusted to the task of the blood glucose prediction, in particular, to deal with the transformed training sets $\mathbf{z}_i$. The use of 1-NN ranking meta-learning involves three following steps:

1. Calculate the distances between the meta-feature $u = (u^{(1)}, u^{(2)})$ of the input x in question and all other $u_\mu = (u_\mu^{(1)}, u_\mu^{(2)}),\ \mu = 1, 2, \ldots, M$ as follows:

$$dist(u, u_\mu) = \sum_{i=1}^{2} \frac{|u^{(i)} - u_\mu^{(i)}|}{\max(u_\mu^{(i)}) - \min(u_\mu^{(i)})}.$$

2. Find $\mu_* \in \{1, 2, \ldots, M\}$ such that

$$dist(u, u_{\mu_*}) = \min\{dist(u, u_\mu),\ \mu = 1, 2, \ldots, M\}.$$

3. For the input x in question take the vector $\omega = \omega_{\mu_*}^0$ that results in the choice of the kernel $K^0 = K_{\omega_{\mu_*}^0}$ from the set (9) and $\lambda = \lambda_{\mu_*}^0$.

The second meta-choice rule, which is proposed by us, is based on the Kernel Choice Theorem 1, or more specifically, on the kernels $K_1^0(u, \hat{u}), \ldots, K_4^0(u, \hat{u})$ obtained in the second case of its application. This rule can be executed as follows:

1. Using the transformed training sets $\mathbf{z}_i = \{(u_\mu, \omega_{i,\mu}^0)\},\ i = 1, 2, 3$ and $\{(u_\mu, \lambda_\mu^0)\}$, we define the following functions $\omega_i^0 = \omega_i^0(u),\ i = 1, 2, 3,\ \lambda^0 = \lambda^0(u)$ of a meta-feature vector $u \in \mathbb{R}^2$:

$$\begin{array}{rcl} \omega_i^0 & = & \arg\min\left\{\frac{1}{M}\sum\limits_{\mu=1}^{M}(\omega(u_\mu) - \omega_{i,\mu}^0)^2 + \alpha_i\|\omega\|_{\mathcal{H}_{K_i^0}}^2\right\},\ i = 1, 2, 3, \\ \lambda^0 & = & \arg\min\left\{\frac{1}{M}\sum\limits_{\mu=1}^{M}(\lambda(u_\mu) - \lambda_\mu^0)^2 + \alpha_4\|\lambda\|_{\mathcal{H}_{K_4^0}}^2\right\}, \end{array} \tag{15}$$

where the regularization parameters $\alpha_i = \alpha_i(K_i^0) \in [\lambda^0, 1],\ \lambda^0 = 10^{-4}$ are chosen in accordance with the quasi-balancing principle [10].

2. Calculate the meta-feature $u = u(x) \in \mathbb{R}^2$ for a prediction input x in question and choose the kernel and the regularization parameter as follows:

$$\begin{array}{rcl} K(t, \tau) & = & K_{\omega^0(u)}(t, \tau) = (t\tau)^{\omega_1^0(u)} + \omega_2^0(u)e^{-\omega_3^0(u)(t-\tau)^2}, \\ \lambda^0 & = & \lambda^0(u). \end{array} \tag{16}$$

Once any of the above mentioned meta-choice rules are employed, the prediction $g(t)$ of the future BG-concentration for the time moment $t \in [t_0, t_0 + PH]$ can be constructed from the past BG-estimates

$$x = ((t_{-m+1}, g_{-m+1}), \ldots, (t_0, g_0)),\ t_0 - t_{-m+1} = SH$$

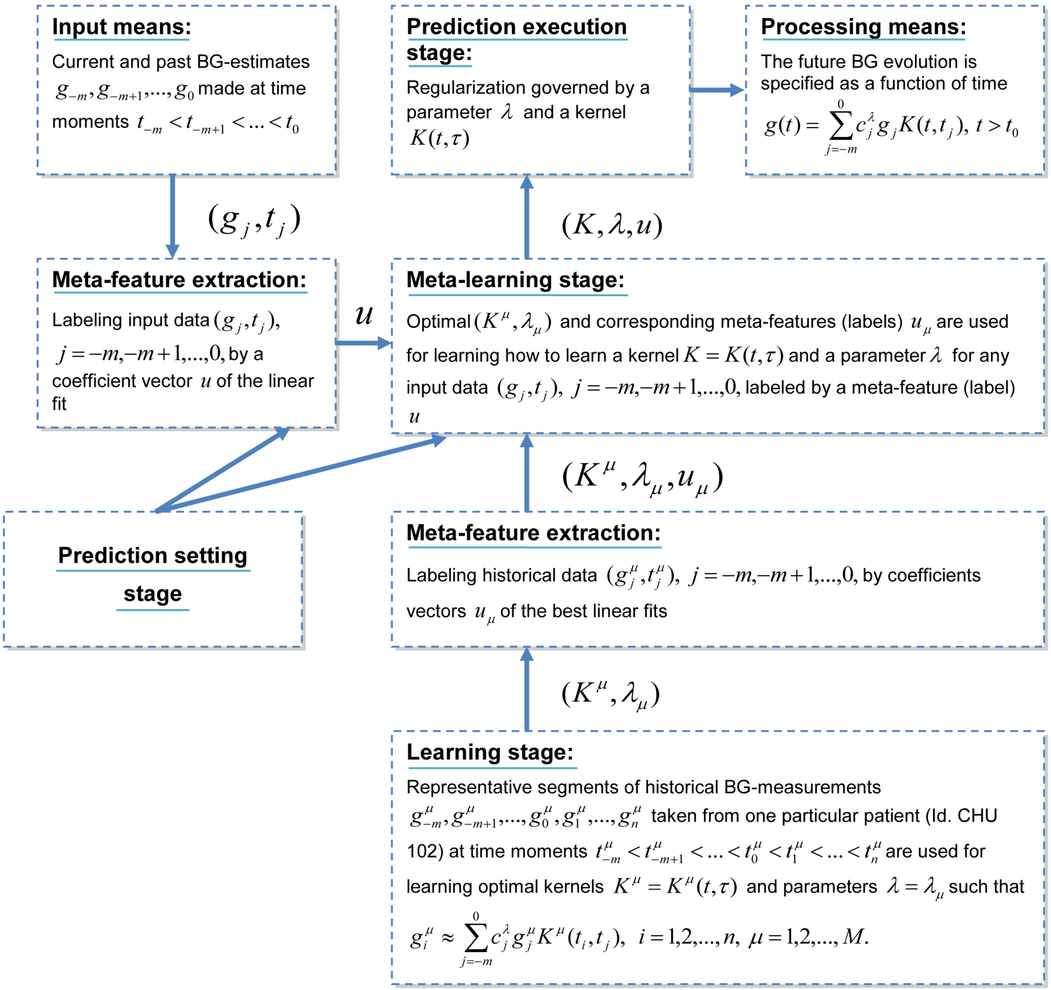
as follows.

At first, we calculate a meta-feature vector $u = u(x) = (u^{(1)}, u^{(2)})$ as the result of the heuristic operation (14). Then using employed meta-choice rule, we specify a kernel $K = K_{\omega^0(u)}$ from the set (9) and $\lambda = \lambda^0(u)$.

Finally, the prediction $g = g(t)$ is defined by means of the regularization performed in the space $\mathcal{H} = \mathcal{H}_K$. Here one may use, for example, two iterations of the Tikhonov regularization, defined as follows:

$$\begin{array}{rcl} g^{(0)} & = & 0, \\ g^{(\nu)} & = & \arg\min\left\{\dfrac{1}{m}\sum\limits_{i=-m+1}^{0}(g(t_i) - g_i)^2 + \lambda\|g - g^{(\nu-1)}\|_{\mathcal{H}_K}^2\right\},\ \nu = 1, 2, \\ g(t) & = & g^{(2)}(t), \end{array} \tag{17}$$

where λ is chosen from $[\lambda^0(u), 1]$ by means of the quasi-balancing principle [10].



Input means:

Current and past BG-estimates $g_{-m}, g_{-m+1}, \ldots, g_0$ made at time moments $t_{-m} < t_{-m+1} < \ldots < t_0$

Prediction execution stage:

Regularization governed by a parameter λ and a kernel $K(t,\tau)$

Processing means:

The future BG evolution is specified as a function of time $g(t) = \sum_{j=-m}^{0} c_j^{\lambda} g_j K(t,t_j),\ t > t_0$

Meta-feature extraction:

Labeling input data (g_j, t_j), $j = -m, -m+1, \ldots, 0$, by a coefficient vector u of the linear fit

Meta-learning stage:

Optimal (K^{μ}, λ_{μ}) and corresponding meta-features (labels) u_{μ} are used for learning how to learn a kernel $K = K(t,\tau)$ and a parameter λ for any input data (g_j, t_j), $j = -m, -m+1, \ldots, 0$, labeled by a meta-feature (label) u

Prediction setting stage

Meta-feature extraction:

Labeling historical data (g_j^{μ}, t_j^{μ}), $j = -m, -m+1, \ldots, 0$, by coefficients vectors u_{μ} of the best linear fits

Learning stage:

Representative segments of historical BG-measurements $g_{-m}^{\mu}, g_{-m+1}^{\mu}, \ldots, g_0^{\mu}, g_1^{\mu}, \ldots, g_n^{\mu}$ taken from one particular patient (Id. CHU 102) at time moments $t_{-m}^{\mu} < t_{-m+1}^{\mu} < \ldots < t_0^{\mu} < t_1^{\mu} < \ldots < t_n^{\mu}$ are used for learning optimal kernels $K^{\mu} = K^{\mu}(t,\tau)$ and parameters $\lambda = \lambda_{\mu}$ such that

$$g_i^{\mu} \approx \sum_{j=-m}^{0} c_j^{\lambda} g_j^{\mu} K^{\mu}(t_i, t_j),\quad i = 1,2,\ldots,n,\ \mu = 1,2,\ldots,M.$$

(g_j, t_j)
(K, λ, u)
u
$(K^{\mu}, \lambda_{\mu}, u_{\mu})$
(K^{μ}, λ_{μ})

Figure 4: Meta-learning approach to BG-prediction: Fully Adaptive Regularized Learning Algorithm

4. Case-Study: Blood Glucose Prediction

In this section, we compare the performance of the state-of-the-art BG-predictors [30, 35] with that of meta-learning based predictors described in Section 3. It is remarkable, in retrospect, that in all cases the meta-learning based predictors outperform their counterparts in terms of clinical accuracy. Even more, for some prediction horizons BG-predictors based on the FARL approach perform at the level of the clinical accuracy achieved by CGM systems, providing the prediction input. Clearly, in general such accuracy cannot be beaten by CGM-based predictors.

The performance assessment has been made with the use of two different assessment metrics known from the literature. One of them is the classical Point Error Grid Analysis (EGA) [7]. It uses a Cartesian diagram, in which the predicted values are displayed on the y-axis, whereas the reference values are presented on the x-axis. This diagram is subdivided into 5 zones: A, B, C, D and E. The points that fall within zones A and B represent sufficiently accurate or acceptable glucose results, points in zone C may prompt unnecessary corrections, points in zones D and E represent erroneous and incorrect treatment.

Another assessment metric is the Prediction Error Grid Analysis (PRED-EGA) [39] that has been designed especially for BG-prediction assessment. PRED-EGA uses the same format as the Continuous Glucose Error Grid Analysis (CG-EGA)[6], which was originally developed for an assessment of the clinical accuracy of CGM systems. To be precise, PRED-EGA records reference glucose estimates paired with the estimates predicted for the same moments and look at two essential aspects of the clinical accuracy: rate error grid analysis and point error grid analyses. As a result, it calculates combined accuracy in three clinically relevant regions, hypoglycemia (<70 mg/dL), euglycemia (70-180 mg/dL), and hyperglycemia (>180 mg/dL). In short, it provides three estimates of the predictor performance in each of the three regions: Accurate (Acc.), Benign (Benign) and Erroneous (Error). In contrast to the original CG-EGA, PRED-EGA takes into account that predictors provide a BG-estimation ahead of time, and it paves a new way to estimating the rates of glucose changes.

The performance tests have been made with the use of clinical data from two trials executed within EU-project "DIAdvisor"[12] at the Montpellier University Hospital Center (CHU), France, and at the Institute of Clinical of Experimental Medicine (IKEM), Prague, Czech Republic.

In the first trial (DAQ-trial), each clinical record of a diabetic patient contains nearly 10 days of CGM data collected with the use of CGM system Abbott's Freestyle Navigator® [1], having a sampling frequency $\Delta t = 10$ (min), while in the second trial CGM data were collected during three days with the use of the system DexCom® SEVEN® PLUS [11] that has a sampling frequency $\Delta t = 5$ (min).

For comparison with the state-of-the-art, we consider two BG-predictors described in the literature, such as data-driven autoregressive model-based predictor (AR-predictor) proposed in [35] and neural network model-based predictor (NNM-predictor) presented in [30].

It is instructive to see that these predictors require more information to produce a BG-prediction than is necessary for our approach. More precisely, AR-predictors use as an input past CGM-measurements sampled every minute. As to NNM-predictors, their inputs consist of CGM-measurements sampled every 5 minutes, as well as meal intake, insulin dosage, patient symptoms and emotional factors.

On the other hand, the FARL-based predictor uses as an input only CGM-measurements from the past 25 minutes (in case of DexCom devices), or 30 minutes (in case of Abbott sensors) and, what is more important, these measurements do not need to be equi-sampled.

Recall that in Section 3 we already mentioned such important feature of our algorithm as portability from individual to individual. To be more specific, for learning at meta-level we use CGM-measurements performed only with one patient (patient ID: CHU102). These measurements were collected during one day of the DAQ-trial with the use of Abbott sensor.

The training data set $\mathbf{z} = \{(x_\mu, y_\mu),\ \mu = 1, 2, \ldots, M\},\ M = 24$, was formed from the data of the patient CHU102 with the sampling horizon $SH = 30$ minutes and the training prediction horizon $PH = 30$ minutes. The application of the procedure described in the Theorem 1 in the first case transforms the training set $\mathbf{z}$ into the values $\omega_\mu^0 = (\omega_{1,\mu}^0, \omega_{2,\mu}^0, \omega_{3,\mu}^0),\ \lambda_\mu^0,\ \mu = 1, 2, \ldots, M$, defining the favorite kernel and regularization parameters.

Then, the transformed training sets $\{(x_\mu, y_\mu)\} \to \{(u_\mu, \omega_\mu^0)\}, \{(u_\mu, \lambda_\mu^0)\},\ \mu = 1, 2, \ldots, 24$, were used for learning at meta-level with FARL method, as well as with 1-NN ranking method.

At first, the obtained fully trained BG-predictors have been tested without any readjustment on the data that were collected during 8 days from other 10 patients taking part in DAQ-trial.

This number of patients is comparable with those used for testing AR- and NNM-predictors, but testing periods for that predictors were shorter than ours. Moreover, a portability from patient to patient was demonstrated only for AR-predictor, and only for 2 patients [35]. As to NNM-predictors [30], they were trained with the use of data from 17 patients and tested on data from 10 other patients.

To assess the clinical accuracy of compared predictors we employ EGA since this performance measure was used in [30, 35] to quantify the accuracy of AR- and NNM-predictors.

In the case of the prediction horizons $PH = 30$ (min) and $PH = 60$ (min), the clinical accuracy of the FARL-predictors is demonstrated in Tables 2 and 6. For the same prediction horizons the comparison of the FARL-predictors with AR-predictors [35], as well as with the predictors based on 1-NN ranking, can be made by using Tables 4, 5 and 6, 7 respectively.

Table 2. Performance of FARL-predictors for PH = 30 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
CHU101	85.07	14.93	-	-	-
CHU102	94.38	5.62	-	-	-
CHU105	93.26	6.74	-	-	-
CHU107	91.69	8.03	-	0.28	-
CHU108	87.31	12.69	-	-	-
CHU115	96.18	3.05	-	0.76	-
CHU116	93.26	6.74	-	-	-
IKEM305	89.88	9.29	-	0.83	-
IKEM306	89.81	10.19	-	-	-
IKEM309	92.12	7.88	-	-	-
Average	**91.3**	**8.51**	**-**	**0.19**	**-**

Table 3. Performance of 1-NN ranking predictors for PH = 30 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
CHU101	82.84	17.16	-	-	-
CHU102	92.13	7.87	-	-	-
CHU105	90.64	9.36	-	-	-
CHU107	86.9	12.25	-	0.85	-
CHU108	88.43	11.57	-	-	-
CHU115	92.75	6.49	-	0.76	-
CHU116	90.64	9.36	-	-	-
IKEM305	89.55	9.95	0.17	0.33	-
IKEM306	90.78	9.22	-	-	-
IKEM309	89.16	10.84	-	-	-
Average	**89.38**	**10.41**	**0.02**	**0.19**	**-**

Table 4. Performance of AR-predictors for PH = 30 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
6-6	85.3	13.3	-	1.4	-
6-8	84.4	14.2	-	1.4	-
8-6	82.2	15%	-	2.8	-
8-8	90	9.8	-	0.2	-
Average	**85.48**	**13.07**	**-**	**1.45**	**-**

Table 5. Performance of AR-predictors for PH = 60 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
6-6	66.2	31.1	0.6	2.1	-
6-8	64.2	32.5	0.2	3.1	-
8-6	60.7	32.9	0.8	5.4	-
8-8	72.9	25.1	-	2.0	-
Average	**66**	**30.4**	**0.4**	**3.15**	**-**

Table 6. Performance of FARL-predictors for PH = 60 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
CHU101	70.15	29.85	-	-	-
CHU102	76.03	23.97	-	-	-
CHU105	78.28	21.72	-	-	-
CHU107	73.24	26.48	-	0.14	1.14
CHU108	69.4	30.6	-	-	-
CHU115	77.48	20.61	-	1.91	-
CHU116	76.4	22.1	0.75	0.75	-
IKEM305	79.27	18.57	0.33	1.66	0.17
IKEM306	75.73	22.82	0.49	0.97	-
IKEM309	75.37	24.63	-	-	-
Average	**75.14**	**24.13**	**0.16**	**0.54**	**0.13**

Table 7. Performance of 1-NN ranking predictors for PH = 60 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
CHU101	63.06	36.57	-	-	0.37
CHU102	56.93	43.07	-	-	-
CHU105	50.19	49.81	-	-	-
CHU107	41.13	54.79	-	3.66	0.42
CHU108	73.13	26.87	-	-	-
CHU115	51.15	43.89	-	4.96	-
CHU116	34.46	62.55	-	3	-
IKEM305	66.83	31.01	0.33	1.66	0.17
IKEM306	48.06	47.57	-	4.37	-
IKEM309	41.38	52.22	-	6.4	-
Average	**52.63**	**44.84**	**0.03**	**2.4**	**0.1**

Tables 8-10 can be used for the comparison of the FARL-predictors against the predictors based on neural networks modeling and on 1-NN ranking. These tables display the prediction accuracy for $PH = 75$ (min), since only this horizon was discussed in [30].

From the comparison of Tables 2-10 one can expect that the proposed FARL-predictors have higher clinical accuracy than their counterparts based on data-driven autoregressive models or on neural networks models.

Table 8. Performance of FARL-predictors for PH = 75 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
CHU101	68.28	31.72	-	-	-
CHU102	68.91	30.71	-	0.37	-
CHU105	70.41	29.59	-	-	-
CHU107	72.83	27.17	-	-	-
CHU108	64.55	35.45	-	-	-
CHU115	67.18	31.3	-	1.53	-
CHU116	71.91	25.09	1.5	1.5	-
IKEM305	71.64	25.04	-	2.82	0.5
IKEM306	67.96	28.16	2.43	1.46	-
IKEM309	64.04	35.47	-	0.49	-
Average	**68.77**	**29.97**	**0.39**	**0.82**	**0.05**

Table 9. Performance of 1-NN ranking predictors for PH = 75 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
CHU101	61.19	38.43	-	-	0.37
CHU102	46.82	52.81	-	0.37	-
CHU105	36.7	49.81	-	-	-
CHU107	30.7	62.96	-	5.49	0.85
CHU108	66.04	33.96	-	-	-
CHU115	41.98	51.53	-	6.49	-
CHU116	26.22	68.91	-	4.87	-
IKEM305	58.87	37.98	0.33	2.32	0.5
IKEM306	36.41	58.25	-	5.34	-
IKEM309	35.96	52.71	-	11.33	-
Average	**44.09**	**50.73**	**0.03**	**3.62**	**0.17**

Table 10. Performance of NNM-predictors for PH = 75 (min)

Patient ID	A (%)	B (%)	C (%)	D (%)	E (%)
1	57.2	38	1.5	3.3	-
2	38.7	40.3	1.2	19	7
3	58.2	37.3	0.5	3.9	-
4	58.8	28.4	0.2	12.2	0.4
5	68.2	24.4	1.2	6.2	-
6	64.9	30.4	0.3	4.5	-
7	42.4	37.7	-	19.4	0.5
8	71.8	28.2	-	-	-
9	71.9	23.7	-	4.4	-
10	78.6	18.6	-	2.8	-
Average	**62.3**	**30**	**0.4**	**7.1**	**0.1**

Note that the accuracy reported in Tables 2-10 was measured against the estimates of the blood glucose given by a commercial CGM system, which, in fact, reads the glucose in the interstitial fluid and is not always accurate in reporting the glucose concentration in the blood (see, for example [26]). Although such CGM systems provide the inputs for predictors, the goal is to predict the real blood glucose.

Therefore, it is interesting to estimate the prediction accuracy against the blood glucose measurements. We can do this with the use of clinical data from another "DIAdvisor" trial (1F-trial) performed at IKEM, where the objective was to check whether a predictor based on the described approach can provide accurate BG-predictions during provocation of hypo- and hyperglycemia. In that trial 6 patients were asked to make one provoked hyperglycemia and one provoked hypoglycemia by applying, respectively, lower dose of insulin (minus 30% of usual dose) and higher dose of insulin (plus 30% of usual dose) before their two lunches.

In contrast to previous study, DexCom sensors were used for providing the prediction input. Besides CGM-measurements, a special blood sampling schedule was adopted to measure real blood glucose concentration by Yellow Springs Instrument (YSI) analyzer during the provocation periods. Blood samples were collected every five to ten minutes during at least 2.5 hours from the beginning of each test. Overall, for each patient 120 blood samples are available for performing the comparison. Such frequently sampled BG-measurements can be used as references in PRED-EGA, which is proven to be very rigorous metric for the assessment of the clinical accuracy of the predictors [39].

For the considered trial it is important to note that the tested FARL glucose prediction system was not specifically readjusted for performing during provocation of hypo- and hyperglycemia. Moreover, the tested system was not readjusted for receiving prediction inputs from DexCom CGM system, which has a different sampling frequency than Abbott used previously. Therefore, the tested system reports the prediction profiles for time moments / horizons $PH = 0, 10, 20, 30$ (min), determined by the Abbott's Freestyle Navigator sampling frequency $\Delta t = 10$ (min), while new prediction profiles are produced every 5 minutes, since DexCom system provides prediction inputs with this frequency.

But what is probably even more important, is that, as in the previous trial, the tested FARL glucose prediction system was not readjusted to any of the patients participating in the trial. More precisely, the prediction process was performed in accordance with (14), (15), (16) and determined with the data of the patient CHU102. Nevertheless, the tested prediction system performed quite well, as it can be seen in Tables 11, 12 and 13, displaying the assessment results produced by PRED-EGA with reference to YSI blood glucose values. The assessment has been made for predictions with the horizons $PH = 0,\ 10,\ 20$ (min) respectively.

Table 11. Performance of FARL-predictors with reference to YSI for $PH = 0$ (min)

Patient	BG ≤ 70 (mg/dL) (%)			BG 70-180 (mg/dL) (%)			BG > 180 (mg/dL) (%)		
ID	Acc.	Benign	Error	Acc.	Benign	Error	Acc.	Benign	Error
305	75	-	25	98.61	1.39	-	94.44	5.56	-
308	100	-	-	92.65	5.88	1.47	100	-	-
310	100	-	-	91.67	3.33	5	95.56	2.22	2.22
311	84.62	15.38	-	69.84	20.63	9.52	70.97	16.13	12.9
320	85.71	14.29	-	75.68	18.92	5.41	87.1	3.23	9.68
308	100	-	-	93.2	5.83	0.97	100	-	-
Avg.	**90.89**	**4.94**	**4.17**	**86.94**	**9.33**	**3.73**	**91.35**	**4.52**	**4.13**

Table 12. Performance of FARL-predictors with reference to YSI for $PH = 10$ (min)

Patient	BG ≤ 70 (mg/dL) (%)			BG 70-180 (mg/dL) (%)			BG > 180 (mg/dL) (%)		
ID	Acc.	Benign	Error	Acc.	Benign	Error	Acc.	Benign	Error
305	84.21	-	15.79	100	-	-	96.97	-	3.03
308	100	-	-	81.82	13.64	4.55	93.94	6.06	-
310	100	-	-	91.38	3.45	5.17	95.74	2.13	2.13
311	75	16.67	8.33	58.33	31.25	10.42	75	16.67	8.33
320	85.71	14.29	-	72.97	24.32	2.7	81.48	-	18.52
308	100	-	-	93.26	5.62	1.12	100	-	-
Avg.	**90.82**	**5.16**	**4.02**	**82.96**	**13.05**	**3.99**	**90.52**	**4.14**	**5.34**

Table 13. Performance of FARL-predictors with reference to YSI for $PH = 20$ (min)

Patient	BG ≤ 70 (mg/dL) (%)			BG 70-180 (mg/dL) (%)			BG > 180 (mg/dL) (%)		
ID	Acc.	Benign	Error	Acc.	Benign	Error	Acc.	Benign	Error
305	78.95	-	21.05	91.3	8.7	-	96.77	-	3.23
308	81.82	-	18.18	80.65	19.35	-	100	-	-
310	100	-	-	92.45	7.55	-	96.08	-	3.92
311	58.33	16.67	25	60.42	31.25	8.33	68	8	24
320	71.43	14.29	14.29	76.92	20.51	2.56	84	-	16
308	100	-	-	90.91	9.09	-	100	-	-
Avg.	**81.75**	**5.16**	**13.09**	**82.11**	**16.08**	**1.81**	**90.81**	**1.33**	**7.86**

PRED-EGA with reference to YSI blood glucose measurements can be also used to assess a CGM sensor, which in such a context could be viewed as an oracle knowing the future prediction input, or as a predictor with the horizon $PH = 0$ (min). The results of such an assessment are shown in Table 14.

The comparison of Tables 11-14 shows that during provocation of hypo- and hyperglycemia the predictions provided by the tested system for $PH = 0,\ 10$ (min) are in average clinically more accurate than the corresponding BG-estimations given by employed CGM device. For $PH = 20$ (min) the accuracy of the tested system is at the level of the CGM accuracy, except for one patient (Patient ID: 311). The effect that for some horizons the tested prediction system can outperform the CGM device, providing prediction inputs, may be explained by the fact that the system takes into account a history of previous measurements and a training in the behavior of CGM to be predicted.

Thus, the performance tests highlight such important features of the presented meta-learning based approach as a portability from individual to individual, as well as from sensor to sensor, without readjustment, the possibility to use data with essential gaps in measurements, and the ability to perform at the level of the clinical accuracy, achieved by approved CGM systems.

5. Conclusions and Future Developments

We have presented a meta-learning approach to choosing the kernels and regularization parameters in kernel-based regularization learning algorithms. This approach allows the development of a new design of a blood glucose predictor for diabetic patients that has been successfully tested in several clinical trials and demonstrated attractive features, which are not inherent to the algorithms known from the literature.

Table 14. Performance of DexCom sensors with reference to YSI

Patient	BG ≤ 70 (mg/dL) (%)			BG 70-180 (mg/dL) (%)			BG > 180 (mg/dL) (%)		
ID	Acc.	Benign	Error	Acc.	Benign	Error	Acc.	Benign	Error
305	75	-	25	100	-	-	89.19	2.7	8.11
308	91.67	8.33	-	91.78	8.22	-	94.74	5.26	-
310	100	-	-	86.57	11.94	1.49	95.74	-	4.26
311	85.71	14.29	-	86.57	11.94	1.49	77.14	20	2.86
320	85.71	14.29	-	78.95	15.79	5.26	76.47	5.88	17.65
308	100	-	-	91.89	6.31	1.8	100	-	-
Avg.	**89.68**	**6.15**	**4.17**	**89.29**	**9.03**	**1.67**	**88.88**	**5.64**	**5.48**

Moreover, in [32] it has been described how the new design can be naturally extended to the prediction from other types of inputs containing not only past BG-estimations but also information about special events, such as meals or physical activities. The predictors based on the extended design also perform quite well, and it gives a hint that the main ingredients of the proposed approach can be exploited in other applications.

More specifically, the optimization procedure described in the Theorem 1 can at first transform a given training data set into a set of parameters defining the favorite kernels, and then the analogous procedure can be performed for learning at meta-level to construct a rule that allows the choice of a favorite kernel for any prediction input in question.

The present paper shows that the meta-learning approach based on this two-steps optimization is rather promising and deserves further development.

Acknowledgment

This research has been performed in the course of the project “DIAdvisor” [12] funded by the European Commission within 7-th Framework Programme. The authors gratefully acknowledge the support of the “DIAdvisor” – consortium.

References

[1] Abbott Diabetes Care, http://www.abbottdiabetescare.com, 2010.

[2] F. Bauer, S. Pereverzev, L. Rosasco, On regularization algorithms in learning theory, J. Complexity 23 (2007) 52–72.

[3] P. Brazdil, C. Giraud-Carrier, C. Soares, R. Vilalta, Metalearning: Applications to Data Mining, Springer-Verlag, Berlin Heidelberg, 2009.

[4] B. Buckingham, H.P. Chase, E. Dassau, E. Cobry, P. Clinton, V. Gage, K. Caswell, J. Wilkinson, F. Cameron, H. Lee, B.W. Bequette, F.J. Doyle III, Prevention of nocturnal hypoglycemia using predictive alarm algorithms and insulin pump suspension, Diabetes Care 33 (2010) 1013–1018.

[5] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Machine Learning 46 (2002) 131159.

[6] W.L. Clarke, S. Anderson, L. Farhy, M. Breton, L. Gonder-Frederick, D. Cox, B. Kovatchev, Evaluating the clinical accuracy of two continuous glucose sensors using Continuous glucose–error grid analysis, Diabetes Care 28 (2005) 2412–2417.

[7] W.L. Clarke, D. Cox, L.A. Gonder-Frederick, W. Carter, S.L. Pohl, Evaluating clinical accuracy of systems for self-monitoring of blood glucose, Diabetes Care 10 (1987) 622–628.

[8] F. Cucker, S. Smale, On the mathematical foundations of learning, Bull. Amer.Math. Soc. (N.S.) 39 (2002) 1–49 (electronic).

[9] E. De Vito, A. Caponnetto, Optimal rates for the regularized least-squares algorithm, Foundations of Computational Mathematics 7 (2007) 331–368.

[10] E. De Vito, S.V. Pereverzyev, L. Rosasco, Adaptive kernel methods using the balancing principle, Foundations of Computational Mathematics 10 (2010) 455–479.

[11] DexCom: Continuous Glucose Meter, http://www.dexcom.com, 2011.

[12] DIAdvisor: personal glucose predictive diabetes advisor, http://www.diadvisor.eu, 2008.

[13] M. Eren-Oruklu, A. Cinar, L. Quinn, D. Smith, Estimation of future glucose concentrations with subject-specific recursive linear models, Diabetes Technol Ther 11 (2009) 243–253.

[14] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Comp. Math. 13 (2000) 1–50.

[15] S. Kindermann, A. Neubauer, On the convergence of the quasi-optimality criterion for (iterated) Tikhonov regularization, Inverse Problems and Imaging 2 (2008) 291–299.

[16] D.W. Klonoff, Continuous glucose monitoring: roadmap for 21-st diabetes therapy, Diabetes Care 28 (2005) 1231–1239.

[17] B. Kovatchev, W. Clarke, Peculiarities of the continuous glucose monitoring data stream and their impact on developing closed-loop control technology, J. Diabetes Sci Technol. 2 (2008) 158–163.

[18] B. Kovatchev, D. Shields, M. Breton, Graphical and numerical evaluation of continuous glucose sensing time lag, Diabetes Technol Ther 11 (2009) 139–143.

[19] V. Kůrková, M. Sanguineti, Approximate minimization of the regularized expected error over kernel models, Mathematics of Operations Research 33 (2008) 747–756.

[20] G.R.G. Lanckriet, N. Christianini, L. Ghaoui, P. Bartlett, M. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 2772.

[21] O. Lepskij, On a problem of adaptive estimation in Gaussian white noise, Theor. Probab. Appl. 35 (1990) 454–466.

[22] C.A. Micchelli, M. Pontil, Learning the kernel function via regularization, J. Mach. Learn. Res. 6 (2005) 1099–1125.

[23] C.A. Micchelli, M. Pontil, On learning vector-valued functions, Neural Computation 17 (2005) 177–204.

[24] V. Morozov, On the solution of functional equations by the method of regularization, Soviet Math. Dokl. 7 (1966) 414–417.

[25] V. Morozov, Methods for Solving Incorrectly Posed Problems, Springer-Verlag, New York, 1984.

[26] V. Naumova, S.V. Pereverzyev, S. Sivananthan, Extrapolation in variable RKHSs with application to the blood glucose reading, Inverse Problems 27 (2011) 075010, 13 pp.

[27] V. Naumova, S.V. Pereverzyev, S. Sivananthan, Reading blood glucose from subcutaneous electric current by means of a regularization in variable Reproducing Kernel Hilbert Spaces, in: 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, Florida, USA, pp. 5158–5163.

[28] C. Palerm, B.W. Bequette, Hypoglycemia detection and prediction using continuous glucose monitoring - a study on hypoglycemic clamp data, J Diabetes Sci Technol. 1 (2007) 624–629.

[29] S. Pappada, B. Cameron, P. Rosman, Development of neural network for prediction of glucose concentration in Type 1 diabetes patients, J Diabetes Sci Technol. 2 (2008) 792–801.

[30] S. Pappada, B. Cameron, P. Rosman, R. Bourey, T. Papadimos, W. Olorunto, M. Borst, Neural networks-based real-time prediction of glucose in patients with insulin-dependent diabetes, Diabetes Technol Ther 13 (2011) 135–141.

[31] S. Pereverzev, S. Sivananthan, Regularized learning algorithm for prediction of blood glucose concentration in “no action period”, in: 1st International Conference on Mathematical and Computational Biomedical Engineering – CMBE2009, Swansea, UK, pp. 395–398.

[32] S. Pereverzyev, S. Sivananthan, J. Randløv, S. McKennoch, Glucose predictor based on regularization networks with adaptively chosen kernels and regularization parameters, EP 11163219.6, 2011.

[33] C. Perez-Gandia, A. Facchinetti, G. Sparacino, C. Cobelli, E.J. Gomez, M. Rigla, A. deLeiva, M.E. Hernando, Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring, Diabetes Technol Ther 12 (2010) 81–88.

[34] D. Phillips, A technique for the numerical solution of certain integral equations of the first kind, J. Assoc. Comput. Mach. 9 (1962) 84–97.

[35] J. Reifman, S. Rajaraman, A. Gribok, W.K. Ward, Predictive monitoring for improved management of glucose levels, J Diabetes Sci Technol. 1 (2007) 478–486.

[36] U. Rückert, S. Kramer, Kernel-based inductive transfer, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Machine Learning and Knowledge Discovery in Databases, volume 5212 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2008, pp. 220–233.

[37] T. Schaul, J. Schmidhuber, Metalearning, Scholarpedia 5 (2010) 4650.

[38] B. Schölkopf, A. Smola, Learning with Kernels, The MIT Press, Cambridge, Massachusetts, 2002.

[39] S. Sivananthan, V. Naumova, C. Dalla Man, A. Facchinetti, E. Renard, C. Cobelli, S. Pereverzyev, Assessment of blood glucose predictors: The Prediction-Error Grid Analysis, Diabetes Technol Ther 13 (2011) 787–796.

[40] L. Snetselaar, Nutrition counseling skills for the nutrition care process, Jones and Bartlett Publishers, 2009.

[41] C. Soares, P.B. Brazdil, P. Kuba, A Meta-Learning Approach to Select the Kernel Width in Support Vector Regression, Machine Learning 54 (2004) 195–209.

[42] V. Solo, Selection of tuning parameters for support vector machine, in: IEEE ICASSP, Philadelphia, PA, pp. 237–240.

[43] G. Sparacino, F. Zanderigo, S. Corazza, A. Maran, Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series, IEEE Trans Biomed Eng 54 (2007) 931–937.

[44] A.N. Tikhonov, V.B. Glasko, Use of the regularization methods in non-linear problems, volume 5, USSR Comput. Math. Phys., 1965.

[45] V.N. Vapnik, Statistical learning theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control, A Wiley-Interscience Publication, John Wiley & Sons Inc., New York, 1998.

[46] G. Wahba, Spline Models for Observational Data, volume 59 of *Series in Applied Mathematics*, CBMS-NSF Regional Conf., SIAM, 1990.

[47] Y. Xu, H. Zhang, J. Zhang, Reproducing kernel Banach spaces for machine learning, J. Mach. Learn. Res. 10 (2009) 27412775.