

Case-base Maintenance of a Personalised and Adaptive CBR Bolus Insulin Recommender System for Type 1 Diabetes

Ferran Torrent-Fontbona^{a,*}, Joaquim Massana^a, Beatriz López^a

^aUniversity of Girona, Campus Montilivi, building EPS4, 17071 Girona, Catalonia

Abstract

People with type 1 diabetes must control their blood glucose level through insulin infusion either with several daily injections or with an insulin pump. However, estimating the required insulin dose is not easy. Recommender systems, mainly based on Case-Based Reasoning (CBR), are being developed to provide recommendations to users. These systems are designed to keep the experiences or cases of the user in a case-base, which requires maintenance to keep system's response accurate and efficient. This paper proposes a case-base maintenance methodology that combines case-base redundancy reduction and attribute weight learning. Contrary to previous approaches designed for classification problems, the maintenance methodology presented in this paper deals with numerical recommendations. It can manage a potentially huge case-base due to the combinatorial derived from the number of attributes used to represent a case. The proposed approach has been tested using the UVA/PADOVA type 1 diabetes simulator and the results demonstrate that it can accomplish better levels of accuracy than other insulin recommender systems mentioned in the literature, when a large number of attributes is considered.

Keywords: Case-based reasoning, insulin recommender system, case-base maintenance, attribute weight learning, patient empowerment, diabetes

1. Introduction

In 2015, the International Diabetes Federation said that Europe had the highest incidence rates of Type 1 Diabetes Mellitus (T1DM) with 140,000 children already having this disease and 21,600 new cases every year (International Diabetes Federation, 2015). Moreover, the American Diabetes Association estimates that there are 1.25 million American adults and children with T1DM.

People with T1DM suffer from autoimmune destruction of the beta cells of the endocrine pancreas, which disables its ability to regulate Blood Glucose (BG). Therefore, people with T1DM need to administer insulin to control their BG level. Large intervention trials, (Diabetes

*Corresponding author

Email addresses: ferran.torrent@udg.edu (Ferran Torrent-Fontbona), joaquin.massana@udg.edu (Joaquim Massana), beatriz.lopez@udg.edu (Beatriz López)

URL: <http://eia.udg.edu/~ftorrent/> (Ferran Torrent-Fontbona), <http://eia.udg.edu/~blopez/> (Beatriz López)

Preprint submitted to Expert Systems with Applications

November 12, 2018

Control and Complications Trial Research Group, 1993), have highlighted the need for accurate and timely insulin dosage to avoid hyperglycaemia¹ and hypoglycaemia².

To control BG, two types of insulin are combined: basal insulin (slow-acting insulin) and bolus insulin (fast-acting insulin). Basal insulin is usually adjusted by clinicians in collaboration with patients during periodic visits. Whereas, bolus insulin doses are estimated on demand, usually at each meal, by the patients themselves using two numerical parameters, Insulin-to-Carbohydrates Ratio (ICR) and Insulin Sensitivity Factor (ISF), along with information about the meal and current BG level. These two parameters are estimated by the patients in collaboration with the clinicians.

There are several bolus calculators on the market to help people to calculate the adequate bolus dosage (Schmidt & Nørgaard, 2014) and their use has been proved useful in improving patients' glycaemic self-control (Doyle et al., 2014; Lepore et al., 2012; Shashaj et al., 2008). However, they do not achieve optimal glycaemic control because of the complexity of glucose metabolism.

The literature presents other approaches based on artificial intelligence that achieve more accurate glycaemic control (Brown, 2015; Herrero et al., 2017; Torrent-Fontbona & Lopez, 2018). Cased-Based Reasoning (CBR), (Aamodt & Plaza, 1994), emerges as one of the favourites methodologies for insulin recommender systems. CBR relies on the use of past experiences stored in a case-base to provide recommendations. However, most of the researchers do not study in depth the problem of maintaining the case-base efficient and accurate. Instead, they usually select a few attributes for describing a case (e.g. physical activity and time of day), omitting the fact that insulin and glucose metabolism depend on many factors such as type of activity (aerobic or anaerobic), ambient temperature, size of the meal, stress, change of the day routine (e.g. weekend or workday) or tiredness (Herrero et al., 2017; Facchinetti et al., 2016; Galloway et al., 1981; Bangstad et al., 2009).

This paper proposes a methodology of case-base maintenance for bolus recommender systems, which consists of two steps: case-base redundancy reduction and learning the relevance of the attributes. The proposed methodology outperforms the CBR bolus recommender system of (Torrent-Fontbona & Lopez, 2018) when tested with the UVA/PADOVA T1DM simulator (Visentin et al., 2014).

The paper is organised as follows: first, related literature is presented; second, the bolus recommender system is described; third, the maintenance methodology proposed in this paper is detailed; fourth, the experimentation and the results are explained; finally, the paper ends with the conclusions.

2. Related Work

This paper presents a case-base maintenance methodology for a CBR bolus recommender system including an algorithm for learning the relevance of the attributes. This section presents related literature in terms of CBR case-base maintenance algorithms and attribute weight learning.

¹Hyperglycaemia can cause, in the long term, micro-vascular complications such as retinopathy, nephropathy and neuropathy, and macro-vascular complications such as coronary heart disease, stroke and peripheral vascular disease.

²Hypoglycaemia may result in clumsiness, trouble talking, loss of consciousness, seizures, or death.

2.1. Case-base maintenance

The use of CBR for developing insulin recommender system was first explored in (Poerschke, 2004). The author proposed to follow the four steps methodology of CBR - retrieve, reuse, revise, retain - proposed in (Aamodt & Plaza, 1994), but leaving key decisions to the user. Nevertheless, she dismissed the implementation of a CBR system because it exceeded the technological limits of the time. Brown extended this work in (Brown, 2015) by proposing and implementing a CBR bolus recommender system, but he did not tackle the maintenance of the case-base and assumed that all query cases with positives outcomes will be stored in the case-base. Despite not tackling the maintenance of the case-base, he pointed out the practical infeasibility of a system with hundreds of cases in the case-base due to the computation time needed in the retrieve step.

The authors in (Herrero et al., 2015a, 2017) propose another bolus recommender system based on CBR. It consists of a combination of CBR and a run-to-run algorithm. The authors devote the retrieve and reuse steps to estimate the accurate ICR of a new case from the most similar one in the case-base. Then, they use a run-to-run algorithm, (Wang et al., 2009), in the revise step to correct the ICR if necessary. Instead of a retain step, the authors propose a case-base with fixed cases and they only allow the correction of the ICRs. Therefore, they are proposing a kind of concept drift approach (Lu et al., 2016), where any incoming information is used to update the target variable of the model, i.e. the ICR of the cases in the case-base. A similar approach is presented in (Herrero et al., 2015b), which differs from that in (Herrero et al., 2015a, 2017) in the revise step, where it corrects the proposed ICR using the area under the postprandial glucose curve, instead of the minimum glucose level of the postprandial glucose curve. Thus, in practice, the methods have a small case-base with approximately nine cases or less. Conversely, this paper proposes a methodology capable of maintaining the case-base efficient and accurate even when cases have several attributes with a large combinatorial.

Maintenance of the case-base has been the focus of in-depth research. First approaches to case-base maintenance can be found in (Leake & Wilson, 1998), and in the utility concept defined in (Smyth & Cunningham, 1996), which gather the situation in which adding more cases to a case-base could reduce rather than improve system efficiency. In (Roth-Berghofer & Reinartz, 2001), case-base maintenance is analysed from a knowledge engineering perspective, providing some guidance to developers. The need for case forgetting and case adding is related to the concept drift, (Delany et al., 2005), as the capacity of a CBR system to adapt to changes in the problem space. These previous concepts were gathered in (Lopez de Mantaras et al., 2005), where the authors highlighted the need to extend the traditional retain step in a more complex maintenance phase with two additional steps: a review step, capable of monitoring quality of the system knowledge, i.e. the case-base, and a restore step, which selects and applies maintenance operations (e.g. deletion or merging of cases).

The literature exhibits several algorithms for the review-restore process. Some examples are the Iterative Case Filtering (ICF) (Brighton & Mellish, 2002), the Coverage Redundancy Reduction (CRR) (Delany et al., 2005), and the Instance-Based learning Algorithm³ (Aha et al., 1991). Current trends on case-base maintenance techniques focus on dealing with big data issues (Juarez et al., 2018): volume, i.e. improving and defining new redundancy reduction techniques like in (Chebel-Morello et al., 2015); variety, i.e. exploring case-base redefinitions due to the impact of dealing with heterogeneous data like in (Leake & Schack, 2015); velocity, i.e. providing a fast answer to massive data flows like in (Lupiani et al., 2014; Lu et al., 2016); and value i.e. providing best solutions in complex problems like in (Yamamoto et al., 2015).

Nevertheless, these approaches were thought to be CBR systems for classification, i.e. prediction of the label of a case for a limited number (usually two) of labels. Conversely, the insulin

recommender system needs to predict a real number. Thus, this paper presents a maintenance phase, which includes retain, review and restore steps capable of reducing case-base redundancy when cases have a numeric solution, the ICR, and adds an attribute weight learning step which is responsible for adapting the weight of each attribute according to its relevance.

2.2. Attribute weight learning

CBR is very sensitive on the similarity function used in the retrieve phase, which usually depends on the relevance of case features expressed by weights. To obtain the weights, several machine learning techniques have been proposed. Decision trees are proposed in (Doan et al., 2008) and gradient descent and genetic algorithm in (Doan et al., 2006) to learn attribute weights and use CBR for predicting costs of residential building projects. The use of genetic algorithms for weight learning is also explored in (Pla et al., 2013). The authors in (Park & Han, 2002) propose to use expert knowledge through an analytic hierarchy process to set the weights of the attributes. A Naive Bayes learner is proposed in (Yeow et al., 2014) to learn attribute weights of a CBR forensic autopsy decision support system. All such methods, however, are batch, meaning that all the data should be available before its application, while there are other incremental approaches that could be more suitable for recommender systems under continuous change. For example, an incremental attribute weight learning algorithm is proposed in (Lamontagne & Bergeron-guyard, 2014). The authors assume that several recommendations are shown to the user and she ranks these recommendations according to her preferences. Then this information is used to adjust attribute weights to achieve the solution ranking provided by the user.

The closest work, regarding attribute weight learning, to the method proposed in this paper is (Salzberg, 1991), where the author presented the Nested Generalised Exemplar (NGE) theory, in which exemplars (or cases) are generalised and described by nested hyperrectangles in a Euclidean n -space that could overlap. He also proposed algorithm EACH in order to classify new cases according to the distance to the stored hyperrectangles and incrementally learn from these new cases. The learning procedure consists of modifying the size of the hyperrectangles and the weights of the corresponding attributes depending on whether the outcome of the recommendation was correct or incorrect.

Therefore, the attribute weight learning algorithm proposed in (Salzberg, 1991) aimed to bring closer or farther the new case and the closest hyperrectangle depending on whether the classification was correct or incorrect. In particular, Salzberg proposed to increase weights of those attributes that matched and decrease those that did not match if the classification was incorrect. If the classification was correct, weights are adjusted in the opposite direction. However, as pointed out in (Wettschereck & Dietterich, 1995), this procedure has different problems and it achieved poor results. Conversely, the authors in (Wettschereck & Dietterich, 1995) proposed not to incrementally update attribute weights but to calculate them according to mutual information using the stored cases in order to give low weight to those attributes which provide no information for classification and high weight to those that provide reliable information.

After (Salzberg, 1991; Wettschereck & Dietterich, 1995), modifications to the NGE theory were presented in (Martin, 1995), modifications to the classification algorithms were proposed in (Zaharie et al., 2011) and implementations of these classification algorithms for applications such as clustering or predicting student's performance are explored in (Hamidzadeh et al., 2015) and (Kulkarni & Ade, 2014), respectively. Nevertheless, no other modifications to the attribute weight learning algorithms of (Salzberg, 1991; Wettschereck & Dietterich, 1995) have been proposed. Conversely to (Salzberg, 1991; Wettschereck & Dietterich, 1995), this paper proposes to update attribute weights when the solution is a numeric value and not nominal. Moreover, this

paper proposes to incrementally adjust weights as in (Salzberg, 1991) but according to mutual information as in (Wettschereck & Dietterich, 1995).

Attribute learning helps to deal with a considerable amount of attributes, but an alternative for dealing with high dimensionality is to split the initial set of data into different groups. This is the case of (Thanh et al., 2017), which proposes the use of neutrosophic clusters to determine the interactions between several recommender systems, so that similar persons can be treated according to their group. A neutrosophic cluster defines memberships functions to determinant clusters as well as to ambiguity (points in cluster boundaries) and outliers clusters (Guo & Sengur, 2015). While this approach is attractive, current results incur in a high computational cost and requires further research to run in an evolving environment, in practice. On the other hand, CBR for insulin recommender systems implies the highest clusterization level: one cluster per patient, because weight learning is performed upon an individual basis, i.e. upon the individual history data of a single person.

The amount of attributes is also related to the sparsity problem when only few values are available or there is no overlapping between two attribute cases (Grčar et al., 2006). In such situations, similarity functions used in the retrieve step, and which make use of the weight of attributes, are unreliable unless defined in an alternative way. In some application domains, it is possible to address the sparsity problem by enriching the information with additional data as categories, distances, time and social network data. This is the case of (Baral & Li, 2016), where the authors propose a new similarity function for points of interest recommendations (e.g. locations) as an average of category and distance sensitive information. In the case of the insulin recommender systems, the sparsity problem does not happen because cases are repetitive experiences about the same variables, whereas in specific situations (e.g. fever), nowadays the cases are considered as outliers because no truth outcome is known.

Other alternative similarity functions are proposed in (Rosaci, 2007), where the authors model each user behaviour with its own ontology, and define similarities among ontologies to carry out recommendations. The learning problem in that case, however, is about ontology learning as a way of learning the similarity function. The ontology approach could be of interest as a future research in the insulin recommender problem, when there is a bulk of case-bases from different persons, and some knowledge could be transferred from one to another to solve the cold start problem, i.e. the initialisation of the case base. To that end, similarity measures on graphs like (Ontañón & Plaza, 2012) could be taken into account in addition to (Rosaci, 2007).

3. Background: PepperRec

Recently, the Patient Empowerment through Predictive decision support (PEPPER) project developed a bolus recommender system based on CBR for people with T1DM called PepperRec (Torrent-Fontbona & Lopez, 2018), with limited maintenance operations. This work is based on PepperRec.

PepperRec describes and contextualises cases using the following attributes: past aerobic physical activity, quantified with four values, planned aerobic activity during the postprandial phase, also quantified with four values, and the time of day, quantified with 24 values. The solution to each case is the ICR, which enables the calculation of the ISF through Equation (1) as proposed in (Walsh et al., 2011), where W is the weight of the user (in kg) and κ is a constant,

e.g. $\kappa = 341.94$ (in mg/dl).

$$ISF = \frac{\kappa \cdot ICR}{W} \quad (1)$$

Upon calculation of the ICR and ISF, the needed bolus dose is calculated by Equation (2), where CHO is the quantity of carbohydrates ingestion (in g), G_c is the BG level (in mg/dl), G_{sp} is the target BG level (in mg/dl), and IOB is the insulin on board, i.e. remaining active insulin from previous doses.

$$B = \frac{CHO}{ICR} + \frac{G_c - G_{sp}}{ISF} - IOB \quad (2)$$

For estimating the ICR of each case, PepperRec follows a CBR methodology, which consists of four steps:

1. Retrieve: Retrieval of the k nearest cases using the Euclidean distance.
2. Reuse: Estimation the new ICR using the weighted mean of the ICR of the retrieved cases based on their similarity to the new case.
3. Revise: Correction of the previously calculated ICR based on the outcome obtained, i.e. the postprandial BG
4. Retain: Storage of the new case if necessary.

At step four, Torrent-Fontbona & Lopez (2018) propose a simple methodology to update the case-base, which consists of removing existing cases in the case-base if a new case, which has equal attributes to other cases, is added. This is performed due to the assumption that the old case is redundant because it has the same ICR, or it is obsolete because it has a different ICR. Nevertheless, this simplistic approach is not able to keep the case-base small and efficient when the number of attributes describing the cases is large. Consequently, the authors decided to describe cases with only three attributes: past and planned aerobic physical activity and time of day.

4. Case-base Maintenance

This paper proposes a novel approach for editing the case-base and learning the relevance of the attributes of the bolus recommender system PepperRec to overcome the current limitations of the system regarding the efficiency of the case-base if the number of considered attributes increases in respect to those proposed in (Torrent-Fontbona & Lopez, 2018).

Thus, it is proposed to include two new steps in PepperRec for CBR maintenance: (i) redundancy reduction, and (ii) attributes weight learning. Figure 1 shows the complete CBR methodology proposed for PepperRec. The new steps proposed by this paper are highlighted in a blue square.

4.1. Redundancy Reduction

The case-base of a CBR system must be efficient and accurate. Namely, redundant cases should be removed because they do not improve the performance. Techniques such as ICF and CRR monitor (review) the reachability and/or the coverage of the cases to then (restore) decide whether it is worth removing a case from the case-base. The reachability of a target case is the number of cases that can be used to properly solve the target case (Lu et al., 2016). The coverage

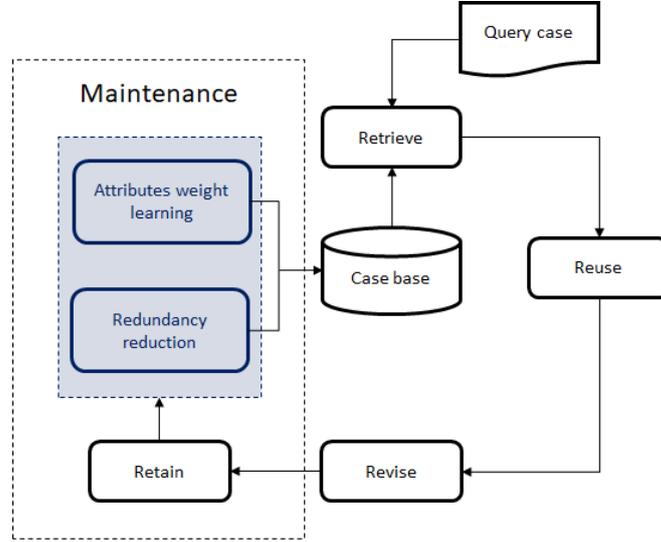


Figure 1: Methodology of the proposed bolus recommender system, which extends PepperRec methodology with a maintenance phase. The steps proposed in this paper are highlighted in a blue square.

of a target case is the number of cases that can be solved properly using the target case (Lu et al., 2016).

These techniques are thought for classification problems where it is clear when a case has properly contributed to solve another case or not (valid or not valid cases). However, in insulin recommender systems where the solution is a real number, for example the ICR, these techniques are not feasible. Therefore, this paper proposes a new method to label a case as valid or not using a tolerance measure in the bolus error.

The permitted error $\Delta B = |B - \tilde{B}|$ is defined as the allowed maximum absolute difference between the optimal bolus B of a case A and the proposed bolus \tilde{B} , i.e. a bolus recommended after reducing the case-base (without case A). However, while the bolus is the outcome prompted to the user, the real product of the CBR methodology is the ICR. Thus, a relationship between the bolus error ΔB and ICR tolerance is necessary.

Using Equations (1) and (2), the bolus B calculation is approximated as follows

$$B = \frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} - IOB \quad (3)$$

Thus, \tilde{B} can be derived from the approximate ICR, \tilde{ICR} , as follows

$$\tilde{B} = \frac{CHO}{\tilde{ICR}} + \frac{G_c - G_{sp}}{\kappa \cdot \tilde{ICR}/W} - IOB \quad (4)$$

According to Equations (3) and (4), the permitted bolus error is defined by Equation (5), where ICR is the correct insulin-to-carbohydrates ratio (the ICR of case A that has been removed) and

$I\tilde{C}R$ is the approximated one.

$$\Delta B = \left| \frac{CHO}{I\tilde{C}R} + \frac{G_c - G_{sp}}{\kappa \cdot I\tilde{C}R/W} - \left(\frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} \right) \right| \quad (5)$$

Since ΔB is the maximum permitted bolus error, Equation (5) can be rearranged as

$$\frac{1}{I\tilde{C}R} \left(CHO + \frac{G_c - G_{sp}}{\kappa/W} \right) \in \left[\frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} - \Delta B, \frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} - \Delta B \right] \quad (6)$$

and, thus, $I\tilde{C}R$ has to fulfil Equation (7).

$$I\tilde{C}R \in \left[\frac{CHO + \frac{G_c - G_{sp}}{\kappa/W}}{\frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} + \Delta B}, \frac{CHO + \frac{G_c - G_{sp}}{\kappa/W}}{\frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} - \Delta B} \right] \quad (7)$$

Considering Equation (8), then, Equation (7) can be expressed as Equation (9).

$$B + IOB = \frac{CHO}{ICR} + \frac{G_c - G_{sp}}{\kappa \cdot ICR/W} \quad (8)$$

$$I\tilde{C}R \in \left[ICR \frac{B + IOB}{B + IOB + \Delta B}, ICR \frac{B + IOB}{B + IOB - \Delta B} \right] \quad (9)$$

In case that $IOB = 0$, the ICR tolerance range is given by

$$I\tilde{C}R \in \left[ICR \frac{B}{B + \Delta B}, ICR \frac{B}{B - \Delta B} \right] \quad (10)$$

Thus, it is assumed that an ICR of a case is valid to solve another case if it fulfils Equation (9), where $I\tilde{C}R$ is the ICR of the former case.

Upon setting this constraint, measures such as the coverage and the reachability can be easily calculated and traditional techniques like CRR, (Delany et al., 2005), or ICF, (Brighton & Mellish, 2002) can be applied to reduce the redundancy of the case-base. CRR sorts cases in descending order of coverage, then, it iteratively adds cases to the new case-base, but when a new case is added, the cases belonging to its coverage are removed from the cases-pending-to-be-added list. Therefore, CRR assumes that cases with lower coverage are those in the borders and, therefore, are the ones to be included in the case-base. On the other hand, ICF calculates the reachability and coverage of each case and removes those whose reachability is greater than the coverage, i.e. those that could be solved by more cases than the cases that could solve the evaluated case.

4.2. Attributes Weight Learning

This paper aims to extend PepperRec to deal with an undetermined number of attributes, whose relevance is not known. Thus, it proposes to learn the relevance of the attributes using a methodology inspired by the algorithm presented in (Salzberg, 1991).

The approach aims to adjust the weights of the attributes every time a new case C_{new} is stored in the case-base. Algorithm 1 details the procedure for the proposed approach. It consists of first selecting the n cases in the case-base with the most similar solution (ICR) to C_{new} (step 3). Next, it calculates the minimum and maximum difference, $minD$ and $maxD$, between the ICR

of C_{new} and any ICR of the n selected cases (steps 4 and 5). Note that X_0 is the selected case with most similar ICR to C_{new} and $X_{||X||}$ is the case with most different ICR to C_{new} . Then, for each selected case and each attribute, the weight of the attribute is increased proportionally to the similarity between its ICR and the ICR of C_{new} , if the values of the attribute (for the selected case and the new one) are equal (step 10). Otherwise, the weight is decreased (step 12). Note that the similarity is defined as $\frac{maxD-d}{maxD-minD}$ and, therefore, it is normalised between the similarities of the n selected cases of step 3. The allowed increment/decrement is proportional to ϵ , so, inversely proportional to the number of attributes to ensure that it does not exceed a percentage, for example, ten percent of the average weight. Thus, in step 2, $beta$ defines this percentage and $\frac{1}{N_{attributes}}$ the average weight. Step 1 defines n proportional to the size of the case-base according to α . At the end (steps 16 to 19), attributes weight are normalised to sum one.

Algorithm 1 Attributes weight learning algorithm

Initialisation: $\alpha = 0.1, \beta = 0.1$

Require: $caseBase, C_{new}$

```

1:  $n \leftarrow \lfloor \alpha \times size(caseBase) \rfloor$ 
2:  $\epsilon \leftarrow \frac{\beta}{N_{attributes}}$ 
3:  $X \leftarrow retrieveCasesWithSimilarICR(n, caseBase, ICR_{C_{new}})$ 
4:  $minD \leftarrow |ICR_{X_0} - ICR_{C_{new}}|$ 
5:  $maxD \leftarrow |ICR_{X_{||X||}} - ICR_{C_{new}}|$ 
6: for each case  $\in X$  do
7:    $d \leftarrow |ICR_{C_{new}} - ICR_{case}|$ 
8:   for each attribute  $\in$  ATTRIBUTES do
9:     if attributecase = attributeCnew then
10:      weightattribute  $\leftarrow$  weightattribute +  $\epsilon \frac{maxD-d}{maxD-minD}$ 
11:     else
12:      weightattribute  $\leftarrow$  weightattribute -  $\epsilon \frac{maxD-d}{maxD-minD}$ 
13:     end if
14:   end for
15: end for
16:  $sum \leftarrow \sum_{attribute \in ATTRIBUTES} weight_{attribute}$ 
17: for each attribute  $\in$  ATTRIBUTES do
18:   weightattribute  $\leftarrow \frac{weight_{attribute}}{sum}$ 
19: end for

```

This approach aims to bring those cases with a similar solution (ICR) closer. This may help to reduce the number of unnecessary cases, because cases with similar or equal ICR share the same region of the problem space, they can be pruned with redundancy reduction algorithms.

5. Results and Discussion

The proposed approach was tested simulating eleven virtual adults throughout 180 days with 20 repetitions with the UVA/PADOVA T1DM simulator (version 3.2), (Kovatchev et al., 2009; Visentin et al., 2014). This simulator is the only one approved by the Food and Drug Administration as a substitute for preclinical trials for certain insulin treatments, including insulin recommender systems (Visentin et al., 2014). The simulator has three main parts: (i) *in silico* subjects with glucose-insulin models based on data from real people, (ii) *in silico* CGM sensor which

models calibration errors, loss of sensitivity depending on BG level, etc., and (iii) *in silico* insulin pump which models the discrete insulin delivery and subcutaneous kinetics. The selected CGM model was the default CGM provided by the simulator and the chosen insulin pump was the one without infusion error in order to test the effectiveness of bolus recommendations.

UVA/PADOVA simulator does not consider intra-day variability of insulin sensitivity and ICR variability depending on physical activity. However, such variability has been modelled and introduced to the simulator by Imperial College Artificial Pancreas group as explained in (Herrero et al., 2017). UVA/PADOVA simulator (and modified versions) has been used to test other insulin recommender systems (Liu et al., 2013; Herrero et al., 2015a,b, 2017; Torrent-Fontbona & Lopez, 2018; Torrent-Fontbona et al., 2017) and the methodology proposed in this paper.

The selected daily quantity pattern of carbohydrate dose intake was a uniform random distribution with mean 70 g (breakfast), 100 g (lunch) and 80 g (dinner) and standard deviation of 10% of the quantity in g. Carbohydrate dose intake estimation uncertainty was modelled as a uniform distribution between -30% and $+10\%$. Carbohydrate intakes pattern and estimation are similar to the ones used in Herrero et al. (2015a, 2017); Torrent-Fontbona & Lopez (2018).

In order to test the performance of the proposed algorithm when dealing with multiple attributes and large combinatorial, additional dummy attributes were added to the CBR recommender system to evaluate the performance of the proposed methodology. In particular, we added eight new attributes: one binary attribute, two attributes with four possible nominal values and five new attributes with three possible values. This involves 7776 times more possible combinations meaning that the case-base could grow up to 373,248 cases.

To analyse the performance of the proposed methodology, the following methods were tested:

- PepperRec according to (Torrent-Fontbona & Lopez, 2018) (state-of-the-art scenario).
- PepperRec and maintenance, which consists of PepperRec with the new methodology described in this paper (redundancy reduction and weights learning). Both, CRR and ICF have been tested as the redundancy reduction technique.
- PepperRec and maintenance without weights learning. Thus, only redundancy reduction methodology is considered using CRR as it achieved better performance than ICF.
- PepperRec and maintenance without redundancy reduction. Thus, only weights learning is considered.

All methodologies were tested with the number of nearest neighbours k equal to 1 and 3 in the retrieve step.

Basal insulin was optimised using the algorithm described in (Torrent-Fontbona, 2018).

5.1. Case-base Efficiency

Case-base efficiency is evaluated in terms of the health results and the size of the case-base. The health outcome is measured according to the portion of time the BG level of the subjects is in, below and above the glycaemic target range, [70, 180] mg/dl.

Table 1 shows the results obtained by the tested methodologies in the last 30 days of the simulations, as it is assumed that convergence is reached, at most, after 150 simulated days. It shows that the inclusion of additional (dummy) attributes in PepperRec reduces the time in the target range down to approximately 77.58% and the size of the case-base is large (more than 370 cases). Considering that simulations consists of 180 days and there are three recommendations

Table 1: Averages of the 11 virtual adults of the mean and standard deviation of the portion of time (%) in, below and above the glycaemic target range, [70, 180], using the proposed methodology (PepperRec and maintenance) with $k = 1$ and $k = 3$ in the retrieve and CRR or ICF as redundancy reduction techniques.

Methods	Time in target (%)	Time Below target (%)	Time above target (%)	Case-base size
PepperRec and maintenance ($k = 1$ and CRR)	79.95 \pm 3.01	8.05 \pm 2.50	12.00 \pm 1.49	140.38 \pm 46.86
PepperRec and maintenance ($k = 3$ and CRR)	80.01 \pm 2.92	8.15 \pm 2.55	11.84 \pm 1.30	59.47 \pm 18.01
PepperRec and maintenance ($k = 1$ and ICF)	79.61 \pm 3.16	8.31 \pm 2.56	12.88 \pm 1.47	117.63 \pm 32.59
PepperRec and maintenance ($k = 3$ and ICF)	80.14 \pm 3.14	8.23 \pm 2.89	11.63 \pm 1.34	110.96 \pm 38.10
PepperRec and maintenance without weights learning ($k = 1$ and CRR)	77.93 \pm 4.19	10.09 \pm 3.96	11.99 \pm 1.86	209.25 \pm 125.14
PepperRec and maintenance without weights learning ($k = 3$ and CRR)	77.78 \pm 4.00	10.68 \pm 3.72	11.54 \pm 1.88	105.65 \pm 79.01
PepperRec and maintenance without redundancy reduction ($k = 1$)	77.85 \pm 3.51	9.37 \pm 2.84	12.78 \pm 1.56	310.57 \pm 41.90
PepperRec and maintenance without redundancy reduction ($k = 3$)	78.09 \pm 3.55	9.41 \pm 2.99	12.49 \pm 1.53	305.40 \pm 47.81
PepperRec with dummy attributes ($k = 1$)	77.58 \pm 3.35	9.75 \pm 2.78	12.66 \pm 1.49	378.90 \pm 50.54
PepperRec with dummy attributes ($k = 1$)	77.66 \pm 3.49	9.86 \pm 3.11	12.48 \pm 1.50	372.89 \pm 57.65

per day, the system stores around 70% of cases. This confirms that the retain step of (Torrent-Fontbona & Lopez, 2018) is not enough to keep the case base efficient.

The inclusion of a redundancy reduction algorithm (for example, CRR) helps to reduce the case-base while the accuracy is maintained, that is time in, below and above the target are similar and the case-base size is between one quarter and one half times the size using only PepperRec. This reduction is especially significant when three neighbours, $k = 3$, are used because the proposed ICRs in a weighted average of three cases are more alike when the CRR algorithm checks if a case is solvable with another one.

Nevertheless, it is the combination of redundancy reduction and weights learning which increases the accuracy of the system with a time in the target range of up to approximately 80%. Moreover, it also reduces the size of the case-base in respect of using CRR without weights learning. Therefore, learning the weights of the attributes helps to bring closer cases that have similar solutions, and then, this shared region of the problem space can be modelled with fewer cases. It is remarkable that when using CRR, the weights learning, and $k = 3$, the size of the case-base is kept to about 60 cases, which is about 15% of the size with PepperRec alone or 10% of the cases generated.

The results achieved using PepperRec and attribute weight learning, but not a redundancy reduction technique, indicate that the accuracy of the recommendations is maintained, since the time in, below and above the target are like those achieved with PepperRec alone, but the size of the case-base is reduced by about 15-20%.

These results are aligned with the hypothesis that attribute weight learning contributes in the pruning of the case-base. But, it is especially effective when it is combined with a redundancy reduction technique such as CRR or ICF.

5.2. Attributes Weight Learning Analysis

Figure 2 illustrates the evolution of the weights of the attributes along time. It shows how the weights of attributes that have a correlation with the ICR (past and planned aerobic activity and time of day) increase along time, and the others decrease. Figure 3 presents the attributes' weight (average and standard deviation) at the end of the simulations, demonstrating that the weights of relevant attributes are significantly higher than the other at the end of the simulations. It is remarkable that the weight of the additional (dummy) attributes is not affected by the number of possible values they could have.

5.3. Discussion

PepperRec, using only past and planned activity and time of day, achieves an amount of time in the glycaemic target range of approximately 84% using the UVA/PADOVA simulator according

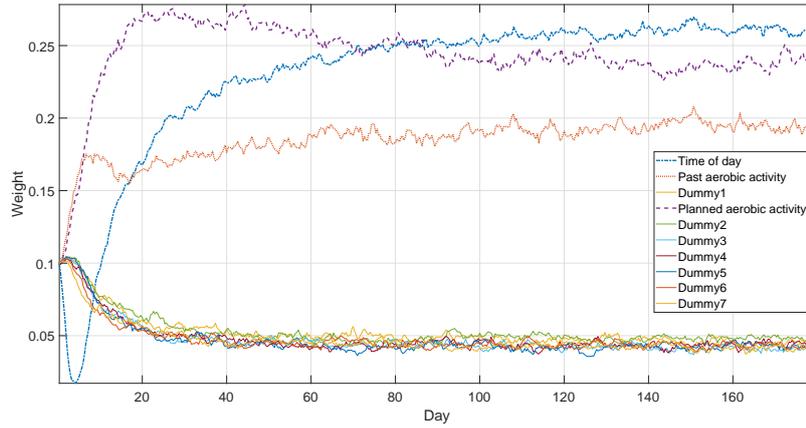


Figure 2: Attributes' weight evolution using Algorithm 1, CRR and $k = 3$.

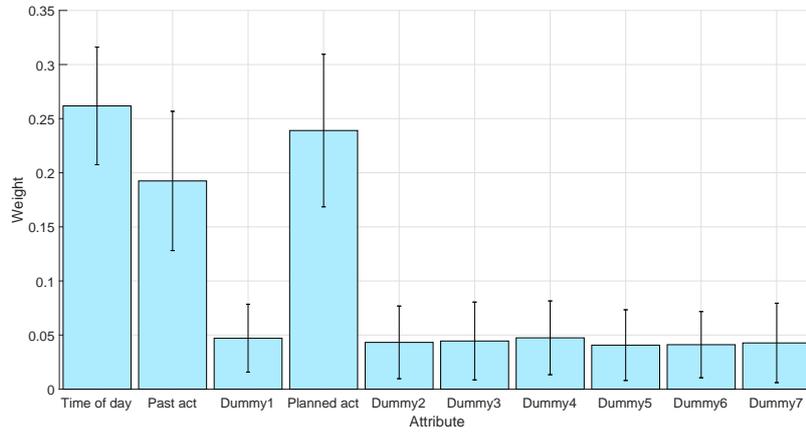


Figure 3: Attributes' weight (average and standard deviation between the 20 repetitions) at the end of the simulations using Algorithm 1, CRR and $k = 3$.

to (Torrent-Fontbona & Lopez, 2018). Similarly, the methodology proposed in (Herrero et al., 2017) achieves a percentage of time in the target of approximately 80% using also past and planned physical activity and time of day as the only attributes describing the cases. These three attributes are the only ones considered by the simulator, which are intra-day variability and variability due to physical activity, but glucose and insulin metabolism depend on more factors. When the number of attributes is increased, PepperRec is not capable of keeping the case-base small and efficient as it stores about 70% of cases. This reduces the accuracy of the recommendations and the achieved amount of time in the glycaemic target range, but this also dissuades its implementation on devices such as smartphones.

On the other hand, the proposed maintenance methodology, which combines case-base redundancy reduction and attribute weight learning, achieves a percentage of time in the glycaemic target range of approximately 80%. This value is much like to the methodology proposed in (Herrero et al., 2017) and, despite this value being slightly worse than that achieved by PepperRec with only physical activity and time of day as attributes, the results demonstrate that the proposed methodology can maintain a case-base small and efficient and achieve a good level of accuracy in the bolus recommendations when the number of attributes is increased. Consequently, the proposed approach is preferable to PepperRec in practice, because it can deal with more attributes and glucose metabolism depends on many factors. Moreover, since the case-base only keeps information about the user, it can learn the relevance of these factors and, then, personalise and adapt the weight of the attributes to it, while PepperRec and the method in (Herrero et al., 2017) cannot. This is remarkable because the list of factors that have an impact on glucose metabolism is known, but their relevance is not known because it depends on each person.

6. Conclusions

This paper presents a maintenance methodology in a numerical CBR recommender system, PepperRec, for bolus recommendation. The proposed approach combines case-base redundancy reduction and attribute weight learning with the objective of having an efficient, personalised and adaptive CBR bolus recommender system. In this regard, it is the first bolus recommender system in the literature thought to deal with many attributes and not just those considered by the UVA/PADOVA type 1 diabetes simulator, that is, time of day or physical activity.

The method has been tested with the UVA/PADOVA type 1 diabetes simulator and the results demonstrate that other state-of-the-art CBR bolus recommender systems reduce the accuracy of the recommendations when there are many attributes. This shows that they are not designed for such a situation since the size of the case-base grows without limit. On the other hand, the results demonstrate that the proposed methodology can keep the case-base small and efficient and learn the relevance of the attributes, making it suitable for its implementation for general use and in portable devices.

Despite the results achieved with the UVA/PADOVA simulator, the proposed methodology should be tested in an environment where factors other than physical activity and time of day have an impact on glucose and insulin metabolism. This means that this methodology should be tested with real people with T1DM because there are no simulators with this capacity.

Acknowledgment

This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No. 689810, www.pepper.eu.com/, PEPPER, and

the grant of the University of Girona 20162018 (MPCUdG2016). The work has been developed with the support of the research group SITES awarded with distinction by the Generalitat de Catalunya (SGR 20142016).

Members of the PEPPER Group are as follows: project leader: C. Martin; project management team: D. Brown, J. Massana, P. Herrero Vinas, L. Nita, J. Masoud, J.M. Fernandez-Real, B. López; co-investigators: A. Aldea, D. Duce, M. Fernandez-Balsells, P. Gay, P. Georgiou, R. Harrison, B. Innocenti, Y. Leal, N. Oliver, R. Petite, M. Reddy, C. Roman, J. Shapley, F. Torrent-Fontbona, M. Waite and M. Wos.

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7, 39–59.
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.
- Bangstad, H.-J., Danne, T., Deeb, L., Jarosz-Chobot, P., Urakami, T., & Hanas, R. (2009). Insulin treatment in children and adolescents with diabetes. *Pediatric Diabetes*, 10, 82–99.
- Baral, R., & Li, T. (2016). Maps: A multi aspect personalized poi recommender system. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 281–284). ACM.
- Brighton, H., & Mellish, C. (2002). Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery*, 6, 153–172.
- Brown, D. (2015). *Temporal case-based reasoning for insulin decision support*. Ph.D. thesis Oxford Brookes University.
- Chebel-Morello, B., Haouchine, M. K., & Zerhouni, N. (2015). Case-based maintenance: Structuring and incrementing the case base. *Knowledge-Based Systems*, 88, 165–183.
- Delany, S. J., Cunningham, P., Tsybal, A., & Coyle, L. (2005). A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, 18, 187–195.
- Diabetes Control and Complications Trial Research Group (1993). The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *The New England Journal of Medicine*, 329, 977–986.
- Doan, S. Z., Arditi, D., & Günaydn, H. M. (2006). Determining Attribute Weights in a CBR Model for Early Cost Prediction of Structural Systems. *Journal of Construction Engineering and Management*, 132, 1092–1098.
- Doan, S. Z., Arditi, D., & Murat Günaydin, H. (2008). Using Decision Trees for Determining Attribute Weights in a Case-Based Model of Early Cost Prediction. *Journal of Construction Engineering and Management*, 134, 146–152.
- Doyle, F. J., Huyett, L. M., Lee, J. B., Zisser, H. C., & Dassau, E. (2014). Closed-loop artificial pancreas systems: engineering the algorithms. *Diabetes care*, 37, 1191–7.
- Facchinetti, A., Del Favero, S., Sparacino, G., & Cobelli, C. (2016). Modeling Transient Disconnections and Compression Artifacts of Continuous Glucose Sensors. *Diabetes Technology & Therapeutics*, 18, 264–272.
- Galloway, J. A., Spradlin, C. T., Nelson, R. L., Wentworth, S. M., Davidson, J. A., & Swarner, J. L. (1981). Factors Influencing the Absorption, Serum Insulin Concentration, and Blood Glucose Responses After Injections of Regular Insulin and Various Insulin Mixtures. *Diabetes Care*, 4, 366–376.
- Grčar, M., Mladenič, D., Fortuna, B., & Grobelnik, M. (2006). Data Sparsity Issues in the Collaborative Filtering Framework. In *In: Nasraoui O., Žaičane O., Spiliopoulou M., Mobasher B., Masand B., Yu P.S. (eds) Advances in Web Mining and Web Usage Analysis. WebKDD 2005. Lecture Notes in Computer Science, vol 4198*. (pp. 58–76).
- Guo, Y., & Sengur, A. (2015). NCM: Neutrosophic c-means clustering algorithm. *Pattern Recognition*, 48, 2710–2724.
- Hamidzadeh, J., Monsefi, R., & Sadoghi Yazdi, H. (2015). IRAHC: Instance Reduction Algorithm using Hyperrectangle Clustering. *Pattern Recognition*, 48, 1878–1889.
- Herrero, P., Bondia, J., Adewuyi, O., Pesl, P., El-Sharkawy, M., Reddy, M., Toumazou, C., Oliver, N., & Georgiou, P. (2017). Enhancing automatic closed-loop glucose control in type 1 diabetes with an adaptive meal bolus calculator in silico evaluation under intra-day variability. *Computer Methods and Programs in Biomedicine*, 146, 125–131.
- Herrero, P., Pesl, P., Bondia, J., Reddy, M., Oliver, N., Georgiou, P., & Toumazou, C. (2015a). Method for automatic adjustment of an insulin bolus calculator: In silico robustness evaluation under intra-day variability. *Computer Methods and Programs in Biomedicine*, 119, 1–8.
- Herrero, P., Pesl, P., Reddy, M., Oliver, N., Georgiou, P., & Member, S. (2015b). Advanced Insulin Bolus Advisor based on Run-To-Run Control and Case-Based Reasoning. *IEEE Journal of Biomedical and Health Informatics*, 19, 1087–1096.
- International Diabetes Federation (2015). *2015 Diabetes Atlas*. Technical Report International Diabetes Federation.
- Juarez, J. M., Craw, S., Lopez-Delgado, J. R., & Campos, M. (2018). Maintenance of Case Bases: Current Algorithms after Fifty Years. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence* (pp. 5457–5463).
- Kovatchev, B. P., Breton, M., Dalla Man, C., & Cobelli, C. (2009). In Silico Preclinical Trials: A Proof of Concept in

- Closed-Loop Control of Type 1 Diabetes. *Journal of Diabetes Science and Technology*, 3, 44–55.
- Kulkarni, P., & Ade, R. (2014). Prediction of Student's Performance based on Incremental Learning. *International Journal of Computer Applications*, 99, 10–16.
- Lamontagne, L., & Bergeron-guyard, A. (2014). Learning Case Feature Weights from Relevance and Ranking Feedback. In W. Eberle, & C. Boonthum-Denecke (Eds.), *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference* (pp. 301–306).
- Leake, D., & Schack, B. (2015). Flexible Feature Deletion: Compacting Case Bases by Selectively Compressing Case Contents. In *Case-Based Reasoning Research and Development. ICCBR 2015. Lecture Notes in Computer Science, vol 9343* (pp. 212–227).
- Leake, D. B., & Wilson, D. C. (1998). Categorizing case-base maintenance: Dimensions and directions. In *Advances in Case-Based Reasoning* (pp. 196–207). Springer Berlin Heidelberg.
- Lepore, G., Dodesini, A., Nosari, I., Scaranna, C., Corsi, A., & Trevisan, R. (2012). Bolus calculator improves long-term metabolic control and reduces glucose variability in pump-treated patients with Type 1 diabetes. *Nutrition, Metabolism and Cardiovascular Diseases*, 22, e15–e16.
- Liu, S.-W., Huang, H.-P., Lin, C.-H., & Chien, I.-L. (2013). Fuzzy-Logic-Based Supervisor of Insulin Bolus Delivery for Patients with Type 1 Diabetes Mellitus. *Industrial & Engineering Chemistry Research*, 52, 1678–1690.
- Lopez de Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., AAMODT, A., & WATSON, I. (2005). Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review*, 20, 215.
- Lu, N., Lu, J., Zhang, G., & Lopez de Mantaras, R. (2016). A concept drift-tolerant case-base editing technique. *Artificial Intelligence*, 230, 108–133.
- Lupiani, E., Juarez, J. M., & Palma, J. (2014). A Proposal of Temporal Case-Base Maintenance Algorithms. In *Case-Based Reasoning Research and Development. ICCBR 2014. Lecture Notes in Computer Science, vol 8765* (pp. 260–273).
- Martin, B. (1995). *Instance-based learning: nearest neighbour with generalisation*. Ph.D. thesis University of Waikato.
- Ontañón, S., & Plaza, E. (2012). Similarity measures over refinement graphs. *Machine Learning*, 87, 57–92.
- Park, C.-S., & Han, I. (2002). A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction. *Expert Systems with Applications*, 23, 255–264.
- Pla, A., López, B., Gay, P., & Pous, C. (2013). eXiT*CBR.v2: Distributed case-based reasoning tool for medical prognosis. *Decision Support Systems*, 54, 1499–1510.
- Poerschke, C. (2004). *Development and evaluation of an intelligent handheld insulin dose advisor for patients with Type 1 diabetes*. Ph.D. thesis Oxford Brookes University.
- Rosaci, D. (2007). CILIOS: Connectionist inductive learning and inter-ontology similarities for recommending information agents. *Information systems*, 32, 793–825.
- Roth-Berghofer, T., & Reinartz, T. (2001). Mama: A maintenance manual for case—based reasoning systems. In *Case-Based Reasoning Research and Development* (pp. 452–466). Springer Berlin Heidelberg.
- Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning*, 6, 251–276.
- Schmidt, S., & Nørgaard, K. (2014). Bolus Calculators. *Journal of Diabetes Science and Technology*, 8, 1035–1041.
- Shashaj, B., Busetto, E., & Sulli, N. (2008). Benefits of a bolus calculator in pre- and postprandial glycaemic control and meal flexibility of paediatric patients using continuous subcutaneous insulin infusion (CSII). *Diabetic Medicine*, 25, 1036–1042.
- Smyth, B., & Cunningham, P. (1996). The utility problem analysed. In *Advances in Case-Based Reasoning* (pp. 392–399). Springer Berlin Heidelberg.
- Thanh, N. D., Ali, M., & Others (2017). A novel clustering algorithm in a neutrosophic recommender system for medical diagnosis. *Cognitive Computation*, 9, 526–544.
- Torrent-Fontbona, F. (2018). Adaptive basal insulin recommender system based on Kalman filter for type 1 diabetes. *Expert Systems with Applications*, 101, 1–7.
- Torrent-Fontbona, F., & Lopez, B. (2018). Personalised Adaptive CBR Bolus Recommender System for Type 1 Diabetes. *IEEE Journal of Biomedical and Health Informatics*, (pp. 1–1).
- Torrent-Fontbona, F., Lopez, B., & Pozo-alonso, A. (2017). A CBR-based bolus recommender system for type 1 diabetes. In *Proceedings of the 2nd International Workshop on Artificial Intelligence for Diabetes* (pp. 9–14). Vienna.
- Visentin, R., Dalla Man, C., Kovatchev, B., & Cobelli, C. (2014). The University of Virginia/Padova Type 1 Diabetes Simulator Matches the Glucose Traces of a Clinical Trial. *Diabetes Technology & Therapeutics*, 16, 428–434.
- Walsh, J., Roberts, R., & Bailey, T. (2011). Guidelines for Optimal Bolus Calculator Settings in Adults. *Journal of Diabetes Science and Technology*, 5, 129–135.
- Wang, Y., Gao, F., & Doyle, F. J. (2009). Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19, 1589–1600.
- Wetschereck, D., & Dietterich, T. G. (1995). An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 19, 5–27.

- Yamamoto, Y., Kawabe, T., Kobayashi, Y., Tsuruta, S., Sakurai, Y., & Knauf, R. (2015). A Refined Case Based Genetic Algorithm for Intelligent Route Optimization. In *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 698–704).
- Yeow, W. L., Mahmud, R., & Raj, R. G. (2014). An application of case-based reasoning with machine learning for forensic autopsy. *Expert Systems with Applications*, *41*, 3497–3505.
- Zaharie, D., Perian, L., & Negru, V. (2011). A view inside the classification with Non-Nested Generalized Exemplars. In *IADIS European Conference Data Mining* (pp. 19–26).