Automatic Detection of Excessive Glycemic Variability for Diabetes Management

Matthew Wiley, Razvan Bunescu, Cindy Marling
School of Electrical Engineering and Computer Science
Ohio University, Athens, Ohio 45701, USA
Email: mw350004@ohio.edu, bunescu@ohio.edu, marling@ohio.edu

Jay Shubrook, Frank Schwartz
ARHI Diabetes and Endocrine Center
Ohio University College of Osteopathic Medicine
Athens, Ohio 45701, USA
Email: shubrook@ohio.edu, schwartf@ohio.edu

***Abstract*—Glycemic variability, or fluctuation in blood glucose levels, is a significant factor in diabetes management. Excessive glycemic variability contributes to oxidative stress, which has been linked to the development of long-term diabetic complications. An automated screen for excessive glycemic variability, based on the readings from continuous glucose monitoring (CGM) systems, would enable early identification of at risk patients. In this paper, we present an automatic approach for learning variability models that can routinely detect excessive glycemic variability when applied to CGM data. Naïve Bayes (NB), Multilayer Perceptron (MP), and Support Vector Machine (SVM) models are trained and evaluated on a dataset of CGM plots that have been manually annotated with respect to glycemic variability by two diabetes experts. In order to alleviate the impact of noise, the CGM plots are smoothed using cubic splines. Automatic feature selection is then performed on a rich set of pattern recognition features. Empirical evaluation shows that the top performing model obtains a state of the art accuracy of 93.8%, substantially outperforming a previous NB model.**

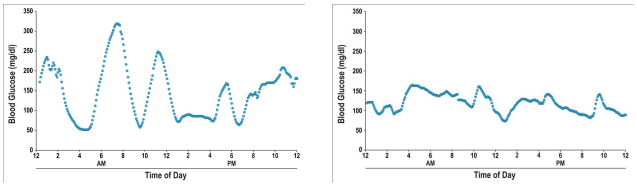
***Index Terms*—classification, glycemic variability, diabetes**

I. Introduction

Over 220 million people have diabetes, a disease in which the body fails to effectively produce or use insulin, an essential hormone needed to convert food into energy [1]. Although diabetes can not be cured, it can be treated and managed. The focus of diabetes management is blood glucose control. By keeping blood glucose profiles as close to normal as possible, patients can delay or avoid serious diabetic complications, including heart disease, kidney failure, blindness, and strokes [2]. Patients are routinely monitored for hyperglycemia, or high blood glucose levels, as well as for hypoglycemia, or low blood glucose levels. There is a recent growing awareness that excessive glycemic variability, or fluctuation in blood glucose levels, also contributes to diabetic complications [3], [4], [5], [6], [7], [8]. However, patients are not yet routinely screened for excessive glycemic variability in clinical practice. There is no definitive metric for glycemic variability; nor is there any available tool to detect excessive glycemic variability [9]. Yet, diabetes specialists readily recognize excessive glycemic variability when they see it in blood glucose plots. This situation motivated the use of machine learning classification to capture physician perception in an automated screen for excessive glycemic variability.

II. Preliminary Study

A preliminary study was conducted in [**?**] to assess the feasibility of automatically classifying blood glucose profiles as exhibiting excessive or acceptable glycemic variability. A synopsis of this previous effort is provided in this section to set the stage for the current work. Blood glucose profiles of diabetes patients were obtained via continuous glucose monitoring (CGM) systems with sensors that sample blood glucose levels at 5-minute intervals. Actual CGM plots illustrating excessive and acceptable glycemic variability are shown in Figure 1.



(a) Excessive glycemic variability (b) Acceptable glycemic variability

Fig. 1. Blood glucose plots obtained through continuous glucose monitoring.

Two physicians were individually shown approximately 300 CGM plots and asked to rate them as excessively variable or not. The physicians concurred in their classifications of 218 plots. These were given as training examples to a naïve Bayes classifier, using as features three domain dependent metrics: mean amplitude of glycemic excursion (MAGE), distance traveled (DT), and excursion frequency (EF).

MAGE was the first glycemic variability metric [10], and it remains the most respected [9]. MAGE calculates the mean distance between the local minima and maxima of a blood glucose plot. Only distances exceeding the standard deviation of the blood glucose values are included in the aggregate. DT and EF were devised to account for aspects of variability not captured by MAGE. DT is the sum of the distances between each pair of consecutive data points; it captures overall fluctuation. EF counts the number of blood glucose excursions. These are the distances between the local minima

and maxima. Only distances greater than 75 mg/dl are included in the count.

To evaluate performance, each physician was asked to classify 100 randomly selected blood glucose plots twice. The naïve Bayes classifier had 85% accuracy on the 61 plots for which physicians gave consistent and concordant classifications. This preliminary work established proof of concept and was previously reported in [**?**]. The rest of this paper describes our subsequent research on methods for improving the glycemic variability classification performance.

III. Supervised Learning for Excessive Glycemic Variability Detection

Our new approach to detecting excessive glycemic variability is developed along three orthogonal directions: noise elimination, feature engineering, and learning algorithms. First, we describe methods for smoothing CGM data to eliminate random noise from the CGM sensors. We then propose a rich set of classical pattern recognition features with the aim of modeling aspects of blood glucose variability not captured by MAGE, DT, and EF. Optimal subsets of features are created by running automatic feature selection algorithms on a separate development dataset, either independent of, or guided by, the actual learning algorithm. Finally, we train and evaluate Support Vector Machines (SVM) and Multi-layer Perceptrons (MP), two general learning algorithms known to obtain state-of-the-art generalization performance in a wide array of domains, and compare them against a previously proposed Naïve Bayes (NB) approach.

IV. Noise Elimination

Smoothing and filtering CGM data has gained interest in the diabetes technology community recently [11], [12]. This is because CGM sensors do not record data with 100% accuracy. The sensors used in this study are known to record values at $\pm 20\%$ of the actual blood glucose level [13]. Physicians implicitly smooth the original sequence of values recorded by the CGM sensor. Figure 2 shows an example plot of CGM data on which a physician has explicitly marked the actual graph used for making diagnostic decisions. We investigated

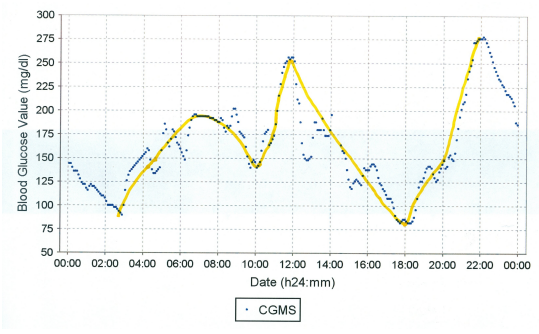


Fig. 2. An example of how a physician smooths a daily plot of CGM data.

a number of smoothing techniques such as simple moving averages, exponential moving averages, polynomial regression with L2 regularization, low pass discrete Fourier transform filter, and cubic spline smoothing. Since smoothing the CGM data using cubic splines was identified by the physicians to best correspond with their implicit smoothing process, we chose to use this method to preprocess the entire dataset.

A. Smoothing CGM Graphs with Cubic Splines

Cubic spline smoothing is a regularized version of cubic spline interpolation. Given a set of n points $\{(x_i, y_i)\}$, the objective of cubic spline interpolation is to connect adjacent points using simple cubic functions S_i, as follows:

$$S_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i, \forall x \in [x_i, x_{i+1}]$$

While this type of interpolation produces smooth curves, random noise from the CGM data is still preserved in the resulting spline. In cubic spline smoothing, the spline $S = \{S_i\}$ is allowed to deviate from the original data points, at the same time requiring that the spline and its first and second derivative remain continuous at all points $\{(x_i, y_i)\}$. In our approach, the spline is determined by minimizing the objective function shown in Equation 1 below:

$$L = \sum_{i=1}^{n} \frac{w_i}{Z}(S_i(x_i) - y_i)^2 + \frac{\lambda}{x_n - x_1}\int_{x_1}^{x_n} |S''(x)|^2 dx \quad (1)$$

where w_i/Z is a normalized weight associated with the i'th point and $Z = \sum w_i$ is the normalization constant. When smoothing the CGM plot, the physicians draw a curve that passes through the *significant* (i.e. least noisy) local optima. To replicate this behavior, a subset of local optima is identified automatically by examining a window of 90 minutes before and after each CGM point. If the point is the maximum or minimum value out of all points in the window, then that point is considered to be a significant local optima. Correspondingly, each w_i is defined as:

$$w_i = \begin{cases} C, & \text{if } (x_i, y_i) \text{ is a significant local optima} \\ 1, & \text{otherwise} \end{cases}$$

A local optima weight of $C = 1000$ was found to obtain the best smoothing behavior on the development data.

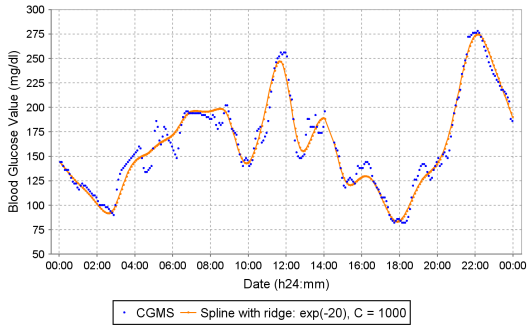


Fig. 3. Cubic spline smoothing with $\lambda = e^{-20}$ and $C = 1000$.

Minimizing the first term of Equation 1 leads to a curve that is close to the original data points, and especially close to the significant local optima because of the larger weights.

The second term corresponds to the average curvature of the spline, and its minimization is meant to effectively smooth away the noise in the original graph. The parameter λ controls the trade-off between the two terms. When $\lambda = 0$, the result is an interpolating spline that passes through every data point, with a volatile concave-convex behavior. At the other extreme, a large value for λ results in a straight line. To avoid issues with missing data points and different starting times of the first CGM reading, we normalize the total curvature of the spline by the length of the graph. Because the number of local optima is not constant for every plot, the weights w_i are also normalized. An example of this type of smoothing is shown in Figure 3.

V. Feature Engineering

The original set of domain specific features described in [**?**] contains MAGE, distance traveled and excursion frequency. To increase the discriminative performance of the model, we augment the feature set with a rich set of pattern recognition features. Table I contains a summary of the features investigated in this work.

TABLE I
INVESTIGATED FEATURES.

Feature	Description
$MAGE$	Mean Amplitude of Glycemic Excursion
EF	Excursion Frequency
DT	Distance Traveled
σ	Standard Deviation
AUC	Area Under the Curve
RR	Roundness Ratio
BE	Bending Energy
ϵ	Eccentricity
DC_i	Direction Codes, for $1 \leq i \leq 3$
FF_i	Amplitudes of low DFT frequencies for $1 \leq i \leq 24$
μ_{pq}	2-Dimensional central moments of order $p+q \leq 3$

A. Excessive Glycemic Variability Detection Features

1) Standard Deviation: Given a CGM graph represented as a sequence of points $\{(x_i, y_i)\}$, this feature is computed as the sample standard deviation over the set of measurements $\{y_i\}$. The intuition is that an excessively variable day will have a higher standard deviation than an acceptable day.

2) Area Under the Curve: This feature is computed as the total area between the CGM graph and a horizontal line corresponding to the minimum blood glucose level measured for that day. Figure 4 shows a blood glucose plot in which the shaded region is used to compute the area under the curve (AUC). The intuition is that a larger area correlates with increased glycemic variability.

3) Central Image Moments: Image moments are computed on the pixel intensities of a given image and can be used to derive useful properties of the image such as the total intensity, centroid, orientation, and moment of inertia. To compute the image moments of a CGM graph, we use the 2-dimensional region shown in Figure 4. If we use C to denote this region and

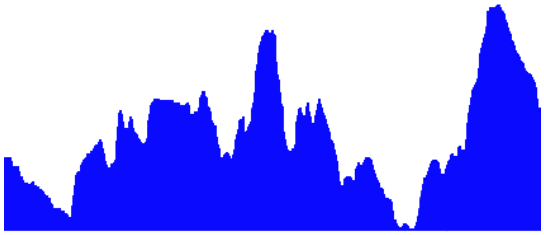


Fig. 4. Area under the curve for the CGM plot shown in Figures 2 and 3.

$f(x, y)$ for the binary intensity values at any pixel position, then the image moments are calculated as follows:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{2}$$

For m_{00}, this equation computes the total number of points in the object. If the entire image contains $N \times M$ pixels, then the $N \times M$ moments m_{pq} of order $p + q$ uniquely determine the image. The lower order moments can therefore be used to summarize the image. Since glycemic variability does not change when the CGM region is translated, we will be using instead the lower order *central moments*, modified versions of image moments that are translation invariant.

Using m_{00} along with m_{10} and m_{01}, the center of mass of each axis can be computed, which then gives the centroid of the shape $(\bar{x}, \bar{y})$:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

The central image moments are then computed as follows:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{3}$$

As features for variability detection we use central moments of order up to 3, i.e. μ_{11}, μ_{20}, μ_{02}, μ_{21}, μ_{12}, μ_{30}, and μ_{03}. Moment μ_{00} is excluded, since it is equivalent to the already considered AUC feature.

4) Eccentricity: Eccentricity can be thought of as the ratio between the maximum and minimum distance from the boundary of the object to its centroid [14]. Eccentricity conveys how much the shape of an object deviates from being circular, or equivalently, how elongated the object is. It is computed as in Equation 4 below, using central image moments:

$$\epsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}}{\mu_{00}} \tag{4}$$

The intuition behind using this feature is that an acceptable day is expected to be more elongated than an excessively variable day, and consequently will have a higher eccentricity.

5) Discrete Fourier Transform: Using the Discrete Fourier Transform (DFT), the sequence $\{y_i\}$ of n CGM measurements in the time domain is transformed into a sequence $\{Y_k\}$ of n complex sinusoidal components in the frequency domain, as shown in Equation 5 below, where $e^{-\frac{2\pi j}{n}}$ is the n-th complex

root of unity:

$$Y_k = \sum_{i=0}^{n-1} y_i e^{-\frac{2\pi j}{n}ki}, \text{where } k = 0, 1...n-1 \quad (5)$$

The corresponding time points $\{x_i\}$ are sampled every 5 minutes for an entire day, resulting in $n = 288$ samples $\{y_i\}$. The complex numbers $\{Y_k\}$ represent the magnitude and the phase of the sinusoidal components of the sampled input function $\{y_i\}$. A particular Y_k corresponds to a sinusoidal component with frequency k/n cycles per day. Like the image moments, the DFT uniquely determines the original signal; therefore, we can use the lower frequencies to summarize the CGM graph. Since very rapid fluctuations are indicative of noise, we use as features the magnitudes of the first 24 components, i.e. $\{||Y_k||, 1 \leq k \leq 24\}$. By ignoring the component Y_0 (a real number), we make the DFT feature set translation invariant.

6) Roundness Ratio: This feature is a ratio between the perimeter of the CGM graph and its area. If $\{p_i = (x_i, y_i)\}$ is a sequence of n CGM points, then the perimeter and the roundness ratio are defined as in Equation 6 below:

$$RR = \frac{P^2}{4\pi\mu_{00}}, \text{where } P = \sum_{i=1}^{n-1} \|p_{i+1} - p_i\| \quad (6)$$

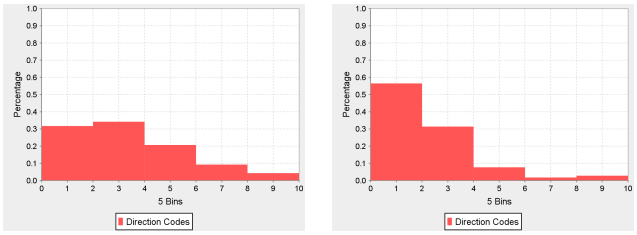
In the general case of 2D objects, this feature will take the value of 1 for a perfect circle, and larger values as the objects deviate more from a circular shape. If an acceptable day resembles a rectangle and an excessively variable day resembles a similar rectangle that is much more jagged (a similar area with a larger perimeter), then the roundness ratio of the excessively variable day will be much larger than that of the acceptable day. Due to its dependence on the perimeter, smoothing CGM data is expected to enhance the discriminative power of this feature.

7) Bending Energy: Bending energy computes the average curvature of the CGM graph $\{(x_i, y_i)\}$, as shown in Equation 7 below, in which P refers to the perimeter.

$$BE = \frac{1}{P}\sum_{i=1}^{n-2}(\theta_{i+1} - \theta_i)^2, \text{where } \theta_i = \arctan\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (7)$$

Due to larger and more frequent fluctuations in blood glucose levels, an excessively variable day should have a higher bending energy than a day with acceptable variability. Smoothing CGM data is expected to improve the discriminative power of this feature because the angles between consecutive points become less sensitive to random noise.

8) Direction Codes: A direction code (DC) is the absolute difference between the values of two consecutive blood glucose readings. Consequently, a CGM plot with n blood glucose measurements has $n - 1$ direction codes. Direction codes for the entire day are placed into bins, depending on their value. A bin b_i is defined by a minimum DC value low_i and a maximum DC value $high_i$, i.e. $b_i = [low_i, high_i)$. If c_i is the total number of direction codes falling into bin b_i, then the corresponding direction code feature is defined as $DC_i = c_i/(n-1)$. The bins used to define the DC features for this application are $b_1 = [0, 3)$, $b_2 = [3, 6)$, and $b_3 = [6, 9)$. We arrived at this particular set of bins by analyzing histograms of direction codes on the CGM data.



(a) Excessively variable day (b) Acceptable day

Fig. 5. Direction code histograms on smoothed data.

Figure 5(a) shows the histogram of direction codes on the smoothed CGM data for the excessively variable day presented in Figure 2. Figure 5(b) shows a histogram for a different day with acceptable variability. The width of each bin in these two histograms is 2 mg/dl. When comparing the two figures, there is an obvious shift of distribution into the first bin, indicating fewer 5-minute blood glucose spikes on the acceptable day. This behavior was consistent across different days of data. A similar, albeit less pronounced difference, was also observed in the histograms of direction codes on raw CGM data when using a bin width of 5 mg/dl.

Based on the analysis of these histograms, a bin width of 3 mg/dl was chosen. This width would only introduce three DC features, adding finer granularity to the raw data histograms and capturing most of the distribution variation in the smoothed data histograms. Using the three bins $b_1 = [0, 3)$, $b_2 = [3, 6)$, and $b_3 = [6, 9)$ means that direction codes with value greater than 9 are ignored (these direction codes are still counted in the total number $n - 1$).

B. Feature Selection

Automatic feature selection was performed on the development dataset for both raw and smoothed CGM data. We investigated filter methods based on the Pearson's Correlation Coefficient (PCC) and the t-test, and the wrapper approach using greedy backward elimination.

1) Filter Methods: The proposed features in Table I were ranked using Welch's t-test (unequal variance two sample t-test). Table II shows the features ranked based on their p-value, using bold for features whose p-value is less than a threshold that was determined based on performance on the development data. Features which did not meet this criteria were filtered out. Filtering features using the PCC did not result in better performance on the development data.

2) Wrapper Methods: Greedy backward elimination was performed 10 times on the development data set, using all but one of the 10-folds as training data. The features which were most common from the results of wrapper selections across the 10 folds were chosen. The features selected using the raw

and the smooth CGM data are shown in bold in the last two columns of Table II.

TABLE II
FEATURE SELECTION USING t-TEST AND BACKWARD ELIMINATION.

T-test Filter (p-value)		Backward Elimination	
Raw	Smooth	Raw	Smooth
AUC (2×10^{-11})	$\mathbf{DC_1}$ (2×10^{-9})	$\mathbf{DC_1}$	**DT**
EF (3×10^{-10})	**DT** (4×10^{-8})	$\mathbf{FF_1}$ $\mathbf{FF_{12}}$	**RR**
DT(3×10^{-9})	**AUC** (5×10^{-8})	**EF**	$\mathbf{DC_3}$
$\boldsymbol{\sigma}$ ($5\times\mathbf{10^{-9}}$)	$\mathbf{DC_3}$ (1×10^{-7})	$\boldsymbol{\mu_{20}}$	**MAGE**
DC_1 (1×10^{-8})	**EF** (6×10^{-7})	ϵ	$\boldsymbol{\sigma}$
DC_3 (4×10^{-8})	$\boldsymbol{\sigma}$ ($7\times\mathbf{10^{-7}}$)	σ	AUC
$MAGE$ (1×10^{-7})	μ_{pq} (2×10^{-5})	DT	EF
μ_{pq} (2×10^{-7})	DC_2 (3×10^{-5})	$MAGE$	ϵ
DC_2 (1×10^{-3})	$MAGE$ (4×10^{-5})	RR	DC_1
ϵ (2×10^{-3})	FF_i (8×10^{-5})	DC_3	DC_2
FF_i (3×10^{-2})	RR (9×10^{-5})	BE	BE
RR (0.12)	BE (2×10^{-3})	AUC	FF_i
BE (0.30)	ϵ (5×10^{-3})	DC_2	μ_{pq}

The four sets of selected features shown in Table II are quite different from each other. No feature appears in all four sets. Direction Codes, Excursion Frequency, Standard Deviation, and Distance Traveled were the only features that appeared in three of the four feature sets. The only features that were not selected in any of the four sets are Eccentricity and Bending Energy. The fact that the four feature sets are very different indicates that the features overlap in terms of the CGM plot information they encode. In the backward elimination setting, the Fourier features were selected only when using raw data. This behavior is consistent with our expectation that smoothing eliminates some of the random noise from the CGM data.

VI. EXPERIMENTAL EVALUATION

In preliminary work, the best performance on excessive glycemic variability detection was obtained using a naïve Bayes learning algorithm. We now believe that the naïve Bayes model is not the most appropriate algorithm for this problem. Naïve Bayes assumes that the features are independent of each other given the class label, which is not the case for our extended set of features. EF and DT, for example, are not independent, since each excursion greater than 75 mg/dl represents a large distance traveled. This motivated us to explore Multilayer Perceptrons (MP) [15] and Support Vector Machines (SVM) [16], [17], two learning algorithms that can seamlessly accommodate overlapping features. MPs that are implemented as a backpropagation network with enough hidden nodes can approximate any decision surface [18]. Similarly, an SVM with a Gaussian kernel is a very flexible learning model, as it can approximate highly non-linear decision boundaries.

The MP and SVM parameters are tuned using grid search on a separate development dataset – the same data that is also used for feature selection. There are 262 examples in the entire glycemic variability dataset, 187 positive and 75 negative. The development dataset is created from 52 randomly chosen examples, 37 positive and 15 negative. The remaining data is used for training and evaluating the models, using 10-fold cross validation. The distribution of the output label in the development set is similar to the label distribution in each of the 10 folds. The same setting (same folds, and same development data) is used for evaluating all the systems. The overall evaluation process is illustrated in Figure 6 below.

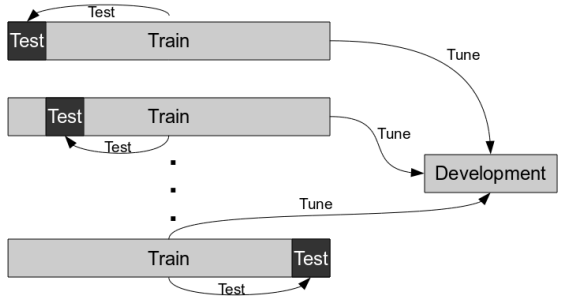


Fig. 6. Illustration of the evaluation setting.

The MP uses the development set to find optimal values for the learning rate and momentum, parameters that control the speed at which the MP corrects itself. The SVM model uses the development set to find the best kernel, and then it re-uses the development set to find optimal parameters for the capacity parameter and the kernel parameters. In all tuning experiments, the Gaussian kernel obtained the best performance. The width of the Gaussian kernel was then optimized on the same development data. We used the Weka implementation [19] for the naïve Bayes model and the Multilayer Perceptron. We used LibSVM [20] for the SVM implementation.

A. Results and Discussion

We report performance using accuracy, sensitivity, and specificity, three error metrics that are commonly used for analyzing the performance of classifiers in the medical domain.

We evaluated the model developed in the preliminary study by training a naïve Bayes classifier on raw data using only the three original features: MAGE, Excursion Frequency, and Distance Traveled. The 10-fold cross validation results are shown in Table III as NB Raw.

TABLE III
PRELIMINARY RESULTS USING 10-FOLD CROSS VALIDATION.

Model	Accuracy	Sensitivity	Specificity
NB Raw	87.1%	78.3%	90.6%

Table IV shows the results of the 10-fold cross validation evaluation for each of the three learning models, using the features automatically selected by the t-test filter. This is shown for both raw and smoothed CGM data. Table V shows the performance of the three learning models when the features are automatically selected using greedy backward elimination. Accuracy, sensitivity, and specificity were reported using the

default probability threshold of 0.5 for the NB model and the trained decision boundary for the SVM and MP models.

TABLE IV
RESULTS OF 10-FOLD EVALUATION USING t-TEST FILTERING OF FEATURES.

Model	Accuracy	Sensitivity	Specificity
NB Raw	87.1%	81.6%	89.3%
NB Smooth	91.9%	91.6%	92.0%
MP Raw	90.0%	83.3%	92.6%
MP Smooth	91.4%	85.0%	94.0%
SVM Raw	89.5%	78.3%	93.3%
SVM Smooth	**92.8%**	88.3%	94.6%

TABLE V
RESULTS OF 10-FOLD EVALUATION USING GREEDY BACKWARD ELIMINATION OF FEATURES.

Model	Accuracy	Sensitivity	Specificity
NB Raw	91.9%	88.3%	93.3%
NB Smooth	89.5%	85.0%	91.3%
MP Raw	91.4%	85.0%	94.0%
MP Smooth	**93.8%**	86.6%	96.6%
SVM Raw	92.8%	85.0%	96.0%
SVM Smooth	91.4%	80.0%	96.0%

When using automatic feature selection, the best accuracy of 93.8% is obtained by the MP model trained on smooth data, with a feature set selected through greedy backward elimination. Smoothing the data increases the performance consistently when using feature sets selected through the t-test. Smoothing also improves the accuracy of the best performing system in the greedy backward elimination setting. A one sided, paired t-test was performed to investigate the significance in improvement between the two best systems shown in bold in Tables IV, V, and the previous NB system shown in Table III. The improvements are significant at levels $p < 0.01$. A more detailed analysis of the statistical significance of the results is presented in the Appendix.

Figure 7 shows the receiver operating characteristic (ROC) curves computed for: the previous NB system (Table III), the current best accuracy system (Table V), and the best accuracy system using a t-test filter (Table IV). To compute the ROC curves, a probability threshold was varied against the probabilities computed by the NB model on the test instances. For the SVM and MP models, the ROC points where obtained by comparing the value of the decision function against a variable threshold.

One interesting result is that, in terms of area under the ROC curve, the best system is the SVM trained on smooth data with filtered features, and not the Multilayer Perceptron with greedy backward elimination that obtained the best accuracy.

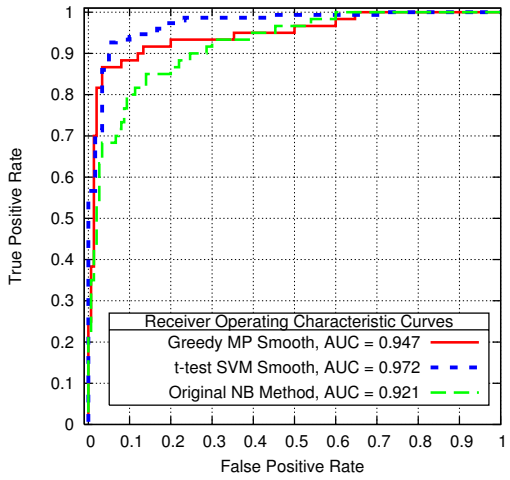


Fig. 7. Comparison of ROC curves for the best classifiers.

VII. CONCLUSIONS AND FUTURE WORK

We have shown that excessive glycemic variability can be automatically detected with high accuracy by training classification models on a small dataset of CGM graphs manually annotated by two diabetes experts. The new models exploit a rich set of pattern recognition features and are trained using two high performance learning algorithms. Experimental results show that smoothing the CGM data leads to significant improvements in performance, and consequently, the current best models substantially outperform a previously proposed Naïve Bayes model. Furthermore, our results indicate that top performance can be obtained using domain independent pattern recognition features to augment and/or replace domain specific metrics.

A constraint that we faced during development of the current models was the relatively small size of the glycemic variability dataset. In order to obtain statistically significant results from the 10-fold cross validation, we had to limit the size of the development set, which may have led to suboptimal parameters and imperfect feature selection. Collecting more annotated CGM plots from diabetes experts is therefore a high priority for future work.

We also plan to change the binary annotation scheme to a "5-star" ordinal scheme wherein physicians would annotate CGM plots using 5 ordered labels, from least variable (1 star) to excessively variable (5 stars). This new annotation scheme would alleviate the problem of disagreement between annotators, with minimal additional effort for the annotators. The new annotations have the potential to further improve accuracy when used in an ordinal regression setting [**?**].

Once refined, this work could be applied as a routine clinical screen to identify at risk patients. Currently, the requisite CGM data may be obtained in one of two ways. Real-time CGM may be integrated with the insulin pumps worn by patients with type 1 diabetes. Retrospective CGM may be used diagnostically for patients with any type of diabetes, for from 3 to 7 days at a time. If a patient were to use CGM

for the week preceding an office visit, automated excessive glycemic variability detection could inform clinical diabetes management.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support from Medtronic, and from Ohio University through the Russ College Biomedical Engineering Fund, the Heritage College of Osteopathic Medicine Research and Scholarly Affairs Committee, the Research Challenge Program, and the Appalachian Rural Health Institute Diabetes Research Initiative. We would also like to thank the participating patients, research nurses, and graduate research assistants for their contributions. This work was partially supported by grant IIS-1117489 from the NSF.

APPENDIX

Tables VI through VIII show the accuracy results previously reported in Section VI-A, together with their standard deviation computed over the 10 folds. Additionally, the tables show the confidence intervals associated with confidence levels of 95% and 99%.

TABLE VI
NB PRELIMINARY RESULTS.

Model	Accuracy	σ	95%	99%
NB Raw	87.1%	5.0%	±3.1%	±4.1%

TABLE VII
t-TEST FILTERING.

Model	Accuracy	σ	95%	99%
NB Raw	87.1%	3.9%	±2.4%	±3.1%
NB Smooth	91.9%	3.9%	±2.4%	±3.1%
MP Raw	90.0%	4.7%	±2.9%	±3.8%
MP Smooth	91.4%	6.6%	±4.1%	±5.4%
SVM Raw	89.5%	5.6%	±3.4%	±4.5%
SVM Smooth	92.8%	6.4%	±3.9%	±5.2%

TABLE VIII
GREEDY BACKWARD ELIMINATION.

Model	Accuracy	σ	95%	99%
NB Raw	91.9%	5.0%	±3.1%	±4.1%
NB Smooth	89.5%	4.9%	±3.0%	±4.0%
MP Raw	91.4%	4.3%	±2.7%	±3.5%
MP Smooth	93.8%	5.9%	±3.6%	±4.8%
SVM Raw	92.8%	5.1%	±3.1%	±4.1%
SVM Smooth	91.4%	5.8%	±3.6%	±4.7%

REFERENCES

[1] World Health Organization, "Diabetes," 2011, http://www.who.int/mediacentre/factsheets/fs312/en/index.html, accessed April, 2011.
[2] Diabetes Control and Complications Trial Research Group, "The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus," *New England Journal of Medicine*, vol. 329, no. 14, pp. 977–986, 1993.
[3] A. Ceriello and M. A. Ihnat, "'Glycaemic variability': A new therapeutic challenge in diabetes and the critical care setting," *Diabetic Medicine*, vol. 27, no. 8, pp. 862–867, 2010.
[4] I. B. Hirsch and M. Brownlee, "Should minimal blood glucose variability become the gold standard of glycemic control?" *Journal of Diabetes and its Complications*, vol. 19, no. 3, pp. 178–181, 2005.
[5] E. S. Kilpatrick, A. S. Rigby, and S. L. Atkin, "For debate. Glucose variability and diabetes complication risk: We need to know the answer," *Diabetic Medicine*, vol. 27, no. 8, pp. 868–871, 2010.
[6] ——, "The effect of glucose variability on the risk of microvascular complications in type 1 diabetes." *Diabetes Care*, vol. 29, no. 7, pp. 1486–1490, 2006.
[7] L. Monnier and C. Colette, "Glycemic variability," *Diabetes Care*, vol. 31, no. s2, pp. S–150–S–154, 2008.
[8] L. Monnier, E. Mas, C. Ginet, F. Michel, L. Villon, J. Cristol, and C. Colette, "Activation of oxidative stress by acute glucose fluctuations compared with sustained chronic hyperglycemia in patients with type 2 diabetes," *Journal of the American Medical Association*, vol. 295, no. 14, pp. 1681–1687, 2006.
[9] D. Rodbard, "Interpretation of continuous glucose monitoring data: Glycemic variability and quality of glycemic control," *Diabetes Technology & Therapeutics*, vol. 11, no. s1, pp. S–55–S–67, 2009.
[10] F. J. Service, G. D. Molnar, J. W. Rosevear, E. Ackerman, L. C. Gatewood, and W. Taylor, "Mean amplitude of glycemic excursions, a measure of diabetic instability," *Diabetes*, vol. 19, no. 9, pp. 644–655, 1970.
[11] B. W. Bequette, "Continuous glucose monitoring: Real-time algorithms for calibration, filtering, and alarms," *Diabetes Science and Technology*, vol. 4, no. 2, pp. 404–418, 2010.
[12] G. Sparacino, A. Facchinetti, A. Maran, and C. Cobelli, "Continuous glucose monitoring time series and hypo/hyperglycemia prevention: Requirements, methods, open problems," *Current Diabetes Reviews*, vol. 4, pp. 181–192, 2008.
[13] J. Mastrototaro, J. Shin, A. Marcus, and G. Sulur, "The accuracy and efficacy of real-time continuous glucose monitoring sensor in patients with type 1 diabetes," *Diabetes Technology & Therapeutics*, vol. 10, no. 5, pp. 385–390, 2008.
[14] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Amsterdam: Elsevier, 2009.
[15] C. M. Bishop, *Neural Networks for Pattern Recognition.* New York: Oxford University Press, 1995.
[16] B. Schölkopf and A. J. Smola, *Learning with kernels - support vector machines, regularization, optimization and beyond.* Cambridge, MA: MIT Press, 2002.
[17] V. N. Vapnik, *The Nature of Statistical Learning Theory.* Berlin: Springer-Verlag, 1995.
[18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
[19] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with Java implementations," in *Proceedings ICONIP/ANZIIS/ANNES99: Future Directions for Intelligent Systems and Information Sciences*, 1999, pp. 192–196.
[20] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, accessed April, 2011.