HealthSense: Classification of Health-related Sensor Data through User-Assisted Machine Learning

Erich P. Stuntebeck[1] eps@gatech.edu
John S. Davis II[2] john@lingfling.com
Gregory D. Abowd[1] abowd@cc.gatech.edu
Marion Blount[3] mlblount@us.ibm.com

[1]School of Interactive Computing and GVU Center, Georgia Institute of Technology, Atlanta, Georgia
[2]LingFling, Inc., Arlington, Virginia
[3]IBM T.J. Watson Research Center, Hawthorne, New York

ABSTRACT

Remote patient monitoring generates much more data than health-care professionals are able to manually interpret. Automated detection of events of interest is therefore critical so that these points in the data can be marked for later review. However, for some important chronic health conditions, such as pain and depression, automated detection is only partially achievable. To assist with this problem we developed HealthSense, a framework for real-time tagging of health-related sensor data. HealthSense transmits sensor data from the patient to a server for analysis via machine learning techniques. The system uses patient input to assist with classification of interesting events (e.g., pain or itching). Due to variations between patients, sensors, and condition types, we presume that our initial classification is imperfect and accommodate this by incorporating user feedback into the machine learning process. This is done by occasionally asking the patient whether they are experiencing the condition being monitored. Their response is used to confirm or reject the classification made by the server and continually improve the accuracy of the classifier's decisions on what data is of interest to the health-care provider.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *indexing methods,* J.3 [**Life and Medical Sciences**]: Health, Medical information systems.

General Terms

Management, Measurement, Design, Experimentation.

Keywords

Healthcare, sensor data analysis, indexing.

1. INTRODUCTION

Recent evidence suggests that remote health monitoring is on the cusp of achieving widespread adoption. Low cost sensors, increased wireless broadband penetration, surging medical expenses and growing demand for tether-less medical solutions are resulting in monitoring scenarios that previously were only discussed in research publications. Example remote health monitoring scenarios include applying pulse oximeters to monitor blood oxygen levels, using glucometers for diabetic patients and detecting cardiac events with portable EKG monitors [2-3,17]. There are networked weight sensors for detecting weight loss and gain, portable EEG sensors for monitoring epilepsy and EMG sensors for detecting muscle dynamics. Additional sensors are on the horizon for application to a wide variety of conditions and will change the relationship between health-care provider and patient.

Even as new sensors are being developed, there are certain medical conditions for which automated detection is not feasible. In such cases, either there is no known way to directly detect the corresponding physiological condition or such detection is impractical from a cost or usability perspective. We refer to this category of health-care conditions as being directly undetectable. Two important conditions, pain and depression, fall in this category, and are both expected to have a growing impact on society as they are often associated with rising chronic health conditions. Chronic pain is correlated to increasing rates of obesity and by 2030 an estimated 67 million Americans are expected to suffer from doctor-diagnosed arthritis [14]. Depression often occurs when patients become aware of their own chronic health conditions; 15-25% of cancer patients suffer from depression once cancer has been discovered [12]. A recent WHO study found that 23% of patients with two or more chronic health conditions also suffered from depression [13].

Given the significance of the above statistics, we believe there is value in extending remote health monitoring techniques to accommodate directly undetectable conditions in the absence of direct sensing technology. We believe that biophysical contextual data collected from a patient suffering from a directly undetectable health condition may be correlated to events associated with the health condition based on patient feedback. Our approach is to hypothesize an initial model and collect data that we believe is loosely associated with the condition events of interest. As data is collected, we automatically detect condition events while recognizing that our decisions may be inaccurate due to imperfections in the initial model. Upon the occurrence of such events, we query the patient in near real-time to determine if our understanding of the existence of an event is consistent with the patient's own psychological awareness of the event. In so doing, we use machine learning to reinforce our understanding of the condition events and improve our model.

HotMobile'08, February 26-26, 2008, Napa Valley, CA, USA
Copyright 2008 ACM 978-1-60558-118-7/08/0002 … $5.00.

To illustrate our approach, we introduce the HealthSense framework for monitoring directly undetectable health conditions. The key design challenge and novelty of HealthSense is to integrate near real-time patient feedback in a machine learning loop involving a patient's own health data. As data is collected the system must quickly determine if the automated understanding of an event is consistent with the patient's awareness of the event. As will be discussed in Section II, this challenge distinguishes this work from other research on activity modeling and remote health monitoring. HealthSense consists of wearable sensors for collecting biophysical context, a patient-worn mobile hub for forwarding sensed data and collecting patient feedback and a server for processing sensed data and detecting events of interest. The HealthSense design is lightweight both as a client and server and emphasis is placed on ease of use over real-time processing. Standard machine learning algorithms are leveraged and patient feedback involves a simple yes/no question to minimize interruption of the patient's normal activities.

While the intended goal of HealthSense is detection of directly unobservable conditions such as pain and depression, organizing a study with patients experiencing either of these symptoms proved to be infeasible for this preliminary work. Instead, to test our approach we chose to monitor the readily available condition of itching. While obviously not a direct proxy for pain or depression, itching is not directly observable by any sensor and thus met our primary requirement. From a medical perspective, detection of itching has potential usefulness with respect to eczema (a condition that impacts 15 million Americans) as well as important childhood diseases such as chickenpox (in which it scratching is not advised) [15]. We thus set out to determine whether the HealthSense system could identify the action of a patient scratching.

In the remainder of this paper we present HealthSense along with results of our preliminary study involving itching. The paper is organized as follows. In Section II we present related work on remote health monitoring as well as activity detection. In Section III we present an architectural overview of HealthSense with a discussion of our design choices. In Section IV we discuss our use case and present preliminary results from a small user study. We discuss future directions and conclude the paper in Section V.

2. BACKGROUND

A large body of previous work exists in the space of remote health monitoring; however, to our knowledge, the problem of detecting directly undetectable health-related events has not been explored. Previous work has focused largely on real-time anomaly detection based on well-defined and pre-existing models [1-3,6]. An oft cited example is that of real-time detection of an irregular heart rhythm using an EKG sensor [3,6]. In contrast to the conditions monitored by HealthSense, an irregular heart rhythm is directly observable from EKG sensor data. Simple, well-known processing methods such as a peak-detector can be used to extract this information from the signal. Blount et al. go a bit further with their ability to detect anomalies based not only on well-known models, but also the patient's historical sensor readings [3]. Management of the large amounts of data accumulated through remote monitoring has been another common theme in the research. Extracting meaningful information from the various types of sensors a patient may be outfitted with and presenting this information in a format suitable for physicians, as well as the patient, is a non-trivial task. Cárdenas et al. explore the problem of data visualization for a patient outfitted with a variety of sensor types [4]. The patient's electronic diary is also correlated to their sensor readings.

In attempting to detect directly undetectable conditions, HealthSense also touches on the areas of activity and gesture recognition from on-body sensors. Although most previous work in this space has not been related to detecting a specific medical condition, HealthSense shares the goal of detecting a high-level, directly undetectable event from low-level sensor streams [8]. Westeyn et al.'s work on the detection of self-stimulatory behavior in individuals with autism using on-body accelerometers comes close to our goals for HealthSense, although they do not mention updating the system with online learning after deployment [11]. Feature extraction and selection is critical for applications involving machine learning and has been examined extensively in the activity recognition research. Lester et al. presented work in which a large number of features (over 600) were extracted for detecting physical activities such as walking, driving, sitting and standing [8,9]. Their work was valuable in prioritizing which features were most important for classification, but they did not consider scenarios in which training is insufficient and in situ feedback is required.

To acquire in situ feedback, HealthSense must query the patient regarding the accuracy of its decisions. Intille et al. describe the Experience Sampling Method for querying users on their current state periodically via a PDA [7,10]. Although they appear to use this data for subsequent off-line supervised learning, they do suggest that online learning is possible and desirable. HealthSense shares with this work the method of sampling data from the user, although rather than doing so periodically, we attempt to reduce the frequency of interruption using initial supervised learning and subsequent polling when the system believes it has observed the condition of interest. The MyExperience project is another work involving collection of smart-phone-based user feedback [5]. However, feedback is not used as part of a machine learning loop and is closer to the format of a small survey than the Boolean questions that HealthSense poses to the patient.

3. THE DESIGN OF HEALTHSENSE

The HealthSense framework consists of three tiers. The sensor tier is the simplest of the three and consists of wearable, wireless sensors that send data to the mobile hub. In this paper we focus on triaxial accelerometers (as our use case in Section IV discusses), although our system accommodates arbitrary sensor types. We experimented with two types of accelerometers: Bluetooth-based Witilt accelerometers and Zigbee-based SunSpots. Since our mobile hub supported Bluetooth links, the Witilt accelerometers were preferred as they could communicate directly to the mobile hub. The SunSpots on the other hand required a wearable SunSpot basestation connected to the mobile hub's USB port. We configured the sensors to deliver data at a sampling rate of 60Hz. Although the system is capable of much higher sampling rates, we found 60 Hz to be more than sufficient given the time domain characteristics we chose to monitor and the frequency domain analysis we performed.

A Nokia N800 PDA serves as our mobile hub (we also used the previous version – the Nokia 770). The Nokia is a good choice because it supports several communication protocols (Bluetooth, 802.11, USB) in a small form factor and has an open architecture running Linux. The mobile hub provides store and forward capabilities for the sensor data with a minimal amount of data processing. The sampled data arriving from the sensors consists of a vector value and a sequence number. Assuming constant rate sampling by the sensors, we could optionally treat the sequence number as a time stamp. Nevertheless, we chose to establish a global sensor clock with the mobile hub and synchronize the sensors by adding time stamps upon data arrival. As data is received from a particular sensor, the time stamps are added and samples are organized into an envelope (an event window) indicating their data type, the sensor ID, and an interval value. The interval value plays the role of a packet number and increases by one for each successive event window for a given sensor. We configured each event window to consist of 128 samples of data. In addition, the mobile hub formats raw sensor data as appropriate. In the case of accelerometer data, it converts raw values into gravitational values. Event windows are then sent to the server as XML.

The HealthSense server performs archiving, feature extraction, and statistical processing of the data it receives. The server is based on Apache Tomcat, embedded Apache Derby and Weka. HealthSense uses a servlet model for receiving data and all received data is stored in the Derby database. Data can be extracted from the database for playback and HealthSense is augmented with the Ptolemy PtPlot utility for displaying signal traces for review by the user [16]. Upon receiving data, features are extracted using utilities we wrote in Java. We extract several features from the data received including the mean, standard deviation, product-moment correlation, and frequency domain energy.

Once features have been calculated for a given event window, they are prepared for processing by the Weka engine. Weka is a Java-based machine learning utility that we embedded into our server. For each event window, a Weka Instance consisting of the calculated features is classified using a given Weka classification engine. The classification is based on training performed with features extracted from previously collected data. We assume that our classifier classifies received data into one of two categories representing the occurrence or non-occurrence of a directly undetectable condition. When positive classification occurs (e.g., when the classifier declares that a condition event of interest did occur), the HealthSense server queries the user to determine if the user agrees that the event occurred. The user is not queried after negative classifications.

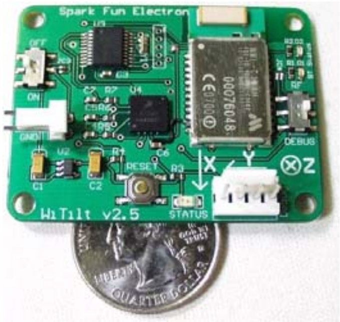

Figure 1. WiTilt Tri-Axial Bluetooth Accelerometer.

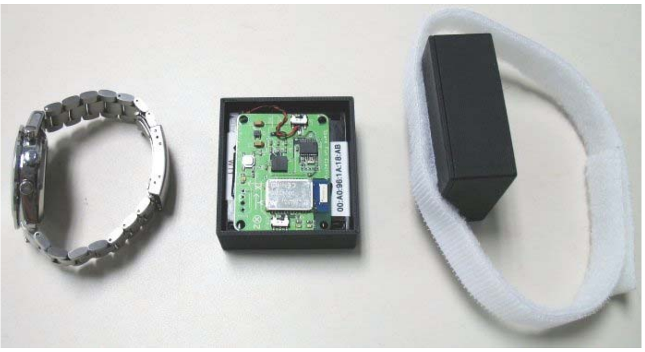

Figure 2. WiTilt accelerometer in case and case configured with wrist-strap compared to a wristwatch.

The user query process involves the server sending a short message to the mobile hub consisting of the event window interval associated with the classification. Reception of such a message by the mobile hub causes a GUI window to pop up with a yes/no question asking the user if the condition event occurred or not. Selecting yes or no on the GUI results in a message being sent back to the server so that the corresponding classification decision can be maintained or updated. Currently the pop up query persists on the GUI until the user responds or a subsequent query is received from the server. If a subsequent query is received, it overrides any previous queries for which the user has not responded so that the user response applies to the latest query received. In a future design, we will likely time out user queries so that they vanish if a user does not respond within a maximum time window. This will allow us to avoid stale queries in which the user is forced to remember a condition that may have occurred long in the past.

Capturing initial data for training our Weka classifiers was a tedious process. The primary challenge was in labeling our data based on manual classification in preparation for automated classification. Such labeling involved manually reviewing data signals with a plotting GUI, extracting relevant signal subsequences, determining features associated with the subsequences and then creating the Weka input files (i.e., ARFF files) based on the features to be used for training the classifier. Because of the manual review component of this process, training with even a small amount of data was laborious. It is our belief that tools for simplifying labeling and manual classification of arbitrary time series data would be a helpful contribution to the field of machine learning applied to pervasive computing.

4. USE CASE: DETECING AN ITCH

To test HealthSense, we focused our attention on the condition of itching. Most people respond to an itching sensation by scratching the region that itches, so our goal was to detect the occurrence of scratching. Itching can occur both as an acute medical condition (e.g., eczema or chickenpox) as well as a routine incident (e.g., scratching in response to an insect bite or the sensation caused by uncomfortable clothing). Our challenge was to differentiate medically relevant scratches (e.g., scratches at a location where the patient is experiencing an eczema outbreak) from routine daily scratching. Assuming the patient scratches both types of itches with a similar motion, this obviously requires a priori knowledge of the locations of medically relevant scratches.

The first decision we faced in applying HealthSense to scratch detection was determining the appropriate type of sensor for detecting a scratch. Accelerometers seemed a natural choice since a patient's response to experiencing an itch involves the motion of scratching. Thus, using two wrist-mounted tri-axial accelerometers (one per hand) we collected data from six users over a period of approximately twenty minutes per user and noted the times during which each user scratched. The noted segments were then manually extracted from the data, along with additional segments not representing a scratch, to provide HealthSense with initial training. Visual inspection of the accelerometer data gave us confidence that a scratch had identifiable characteristics that could be automatically detected.

After extracting windows of the appropriate size from the raw time-series data, various features were calculated. These included the frequency-domain energy of the signal in 1 Hz increments between 0 and 10 Hz, the correlation between the three accelerometer axes, the standard deviation of the time domain for each axis, and the RMS of each time-series axis. We found approximately 2.13 seconds of data sampled at 60 Hz (128 samples) to be sufficient for detecting scratching with a tri-axial accelerometer. The choice of window size over which to extract the desired features from the raw sensor data is an important consideration and will likely change with different types of sensors and different applications. Note that lengthening the window size lengthens the decision period of the system, and thus the amount of time before the system can query the patient for confirmation if necessary.

To get an idea of the potential classification accuracy of the system, we fed our data into the Weka machine learning toolkit and explored various classifiers commonly found in the activity recognition literature such as the C4.5 decision tree, naïve Bayes classifier, and the multilayer perceptron neural network. Using leave-one-out analysis of the data, we found that scratches could be identified with accuracy between 87% (naïve Bayes) and 91% (C4.5 decision tree). These results confirmed the feasibility of detecting scratching with a tri-axial accelerometer, the features we calculated, and the selected classifiers.

With the system now somewhat trained from this supervised learning procedure, we set out to quantify any improvement in classification accuracy afforded by the user-query feedback mechanism. As ground truth, we ran the system using only the initial training data for several minutes and observed the accuracy. The test was conducted while seated at a desk and performing standard activities for this setting such as using a mouse and keyboard. As would be expected, accuracy remained fairly constant throughout the test, ranging from 63% to 73% in each minute. Note that the classifier always has a binary output – either we observe the condition of interest or not. Next we ran the same test without scratching, however this time we used the user's feedback to update the classifiers after each decision was made. This time the accuracy in each minute rose from 62% in the first minute to a maximum of 93% in the third minute. This not only demonstrates the value of user feedback, but also how quickly the system can be made to improve upon its initial training. In a third test we incorporated scratching at a rate of once per minute in a random location. The scratch was always correctly identified and similar accuracies to the last test were observed. The first minute showed 62% accuracy and a maximum of 93% accuracy was obtained in the third minute. Finally, we set out to determine the system's performance in determining scratches at a particular location. Eczema patients, for example, do not necessarily experience eczema everywhere on their body and thus the system should be able to differentiate normal daily scratching from scratching at the site of the eczema. We tested whether the system could differentiate a scratch to the back of the neck versus elsewhere on the body. With training, the system improved in accuracy from 81% to a maximum of 100% in the third minute. It was clear during the course of testing that when scratching elsewhere the system would initially make an incorrect decision and identify that action as the condition of interest. However, with several responses from the user indicating the decision was incorrect, the system could then rule out that location at near 100% accuracy. All above results were obtained using a C4.5 decision tree. Initially we attempted to use a neural network as the classifier, however retraining it with each incoming sample proved to be too processor intensive for the system to operate in real-time.

5. FUTURE DIRECTIONS

The work presented in this paper serves as an initial indication of the usefulness of user-assisted feedback applied to data collection in remote health monitoring systems. As this work continues, we see several areas for improvement. First and foremost, we recognize that our feature pool can be expanded significantly. In general, it will not be known a priori which features are most useful. Nevertheless, the availability of a large number of features can be automatically pruned and prioritized based on the user feedback as certain features are discovered to be more important than others. In a similar fashion, additional sensors will likely be relevant depending on the condition being monitored. The issue of changes in orientation of the sensors while training the system has not yet been addressed. An additional accelerometer to indicate whether the patient is sitting or standing could be used as another feature for the classifier and may improve accuracy. Some of the additional sensors may not necessarily be worn on the patient's body and may involve contextual data about the patient's surroundings. For example, the presence of antagonistic individuals may have an impact on depression. Herein lies an opportunity for HealthSense: the ability to discover important correlations that otherwise would be difficult to detect.

A disadvantage of our current approach to receiving user feedback is that we reinforce our model against positive classifications but we do not account for the occurrence of negative classifications. In effect, we do not appropriately handle false negatives (i.e., if HealthSense fails to detect a scratch). One way to handle this is to allow the patient to voluntarily notify HealthSense when a scratch occurs that HealthSense fails to detect. The challenge in patient initiated declarations is that timing becomes critical since a patient's declaration must be precisely aligned with the data it corresponds to. An improved approach is for HealthSense to query the patient for validation of both positively and negatively classified results. Alignment is no longer an issue in this case since the query will be associated to a particular event window, assuming that the query rate is relatively slow (e.g., a patient-noticeable delay should occur between successive positive and negative queries). Unfortunately, queries about both positive and negative classifications risk the possibility of overwhelming the patient since every event window will contain either a positive or negative classification. Gauging the frequency with which it is appropriate to interrupt the user for feedback is actually an interesting problem in itself.

In this paper we presented HealthSense, a framework for automated detection of health-related events that can not be directly observed by any current sensor. HealthSense assumes that the occurrence of these conditions is correlated with events that can be observed. This correlation may not be obvious though, and thus we occasionally require feedback from the patient to validate the system's decisions. Although in this preliminary work we examined scratching, we believe this system's real promise lies in detecting conditions such as pain and depression. The system is designed to be flexible though, and can accommodate any type of condition and all varieties of sensors.

6. REFERENCES

[1] D. Apiletti, E. Baralis, G. Bruno, and T. Cerquitelli, "SAPhyRA: Stream Analysis for Physiological Risk Assessment," Twentieth IEEE International Symposium on Computer-Based Medical Systems, pp. 193 – 198, 2007.

[2] M. Blount, V. M. Batra, A. N. Capella, M. R. Ebling, W. F. Jerome, S. M. Martin, M. Nidd, M. R. Niemi, and S. P. Wright, Remote health-care monitoring using Personal Care Connect, IBM Systems Journal, Volume 46, Number 1, 2007.

[3] M. Blount, J. Davis, M. Ebling, J. Kim, K. Kim, K. Lee, A. Misra, S. Park, D. Sow, Y. Tak, .M. Wang, and K. Witting, "Century: Automated Aspects of Patient Care," Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 21 - 24, Daegu, Korea, 2007.

[4] A. Cárdenas, R. Pon, and R. Cameron, "Management of Streaming Body Sensor Data for Medical Information Systems," Proceedings of METMBS 2003, Las Vegas, NV, pp. 186-191, 2003.

[5] Froehlich, M. Chen, S. Consolvo, B. Harrison, & J. Landay, "MyExperience: A System for In Situ Tracing and Capturing of User Feedback on Mobile Phones," Proceedings of MobiSys 2007, San Juan, Puerto Rico, June 11-14, 2007.

[6] S. Intille, L. Bao, E. Tapia, and J. Rondoni, "Acquiring In Situ Training Data for Context-Aware Ubiquitous Computing Applications," Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp 1 - 8, Vienna, Austria, 2004.

[7] S. Intille, J. Rondoni, C. Kukla, I. Ancona, and L. Bao, "A Context-Aware Experience Sampling Tool," Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp 972 - 973, Ft. Lauderdale, Florida, 2003.

[8] J. Lester, T. Choudhury and B. Borriello, "A Practical Approach to Recognizing Physical Activities," Pervasive 2006, LNCS 3968, pp. 1 - 16, 2006.

[9] J. Lester, T. Choudhury, N. Kern, G. Borriello and B. Hannaford, "A Hybrid Discriminative/Generative Approach for Modeling Human Activities," Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, pp. 766 - 722, Edinburgh, Scotland, 2005.

[10] E. Tapia, S. Intille, and K. Larson, "Activity Recognition in the Home Using Simple and Ubiquitous Sensors," Pervasive 2004, LNCS 3001, pp. 158 – 175, 2004.

[11] T. Westeyn, K. Vadas, X. Bian, T. Starner, and G. Abowd, "Recognizing Mimicked Autistic Self-Stimulatory Behaviors Using HMMs," Proceedings of the Ninth IEEE International Symposium on Wearable Computers, pp. 165 - 167, 2005.

[12] http://www.cancer.gov/cancertopics/pdq/supportivecare

[13] http://www.medpagetoday.com/Psychiatry/Depression-/tb/6617

[14] http://www.cdc.gov/arthritis/data_statistics

[15] http://www.wrongdiagnosis.com/e/eczema/stats.htm

[16] http://ptolemy.berkeley.edu/java/ptplot/

[17] http://www.bodymedia.com