Using Many Simple Sensors for Automatic Monitoring in the Home

Daniel Wilson, Britton Wolfe, and Chris Atkeson

Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
{dan.wilson, cga}@cs.cmu.edu
bwolfe@andrew.cmu.edu

Abstract. About 1 in 5 Americans have some kind of disability, and 1 in 10 have a severe disability [16]. Instrumented environments can be used for automatic monitoring for people with physical and cognitive disabilities. We describe a system that uses a handful of motion detectors to predict light switch usage in an instrumented home. We use a naive Bayes learner with an automatically extracted feature set. We demonstrate results over two months of data collected from an instrumented house occupied by students. We explain why our methods are independent of this environment and describe specific applications to home health care.

1 Introduction

Automatic monitoring of people with cognitive disabilities can improve the accuracy of pharmacologic interventions, track illness progression, and lower caregiver stress levels [7]. Unfortunately, sensor choice and placement are severely restricted. For instance, Alzheimer's patients frequently strip themselves of clothing, including any wearable sensors [4]. Elderly individuals are also more sensitive to small changes in environment [5]. The challenge is to provide generalizable methods that can provide significant monitoring capability by exploiting low-cost, minimally invasive sensors.

Current industrial monitoring applications rely upon humans to manually monitor the status of motion detectors, bed-exit sensors, bed-wetting sensors, and staff calling buttons [23]. We use similar sensors because they report important information at minimal privacy, monetary, and computational cost. [22] has shown that movement patterns alone are an important indicator of cognitive function, depression, and social involvement among people with Alzheimer's disease living in nursing homes. Light switch usage is a significant indicator of movement. As a proof of concept experiment we use motion detectors and instrumented light switches to (1) predict when a light switch will be manipulated (in real-time), and (2) predict which light switch it will be.

2 Instrumented Environment

We have instrumented a two story house occupied by two students (including an author) with a variety of sensing equipment. See Figure 1 for details. This experiment uses data

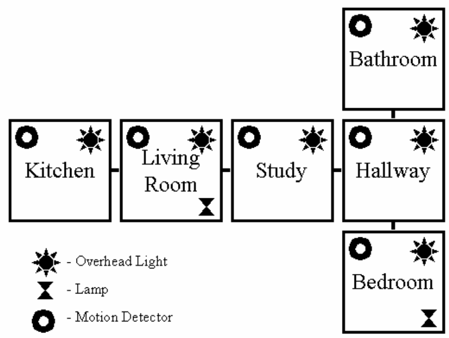

Fig. 1. Sensor placement.

gathered from motion detectors and light switch sensors. Motion detectors provide a binary indication of heat and movement (e.g., human presence) in an area. They are unobtrusive and widely used in security systems. The automated light switch [19] reports an on/off value when manipulated by a user and can also be manipulated remotely through an X10 protocol.

X10 is a communications language that allows devices to communicate via existing 110V electrical wiring. We use eight automated wall and lamp switches to monitor and control lighting levels throughout the house. Overhead lights controlled by wall switches are located in the kitchen, living room, study, hallway, bathroom, and bedroom. There is one lamp in the living room and another in the bedroom, both controlled by switches mounted nearby. There are also six wireless motion detectors located in the six rooms: the kitchen, living room, study, hallway, bathroom, and bedroom. The detectors are attached by velcro to the ceilings in order to maximize room coverage. The detectors require both heat and movement to trigger, and readings are logged as they occur. Sequentiality is not guaranteed, as multiple detectors may be triggered simultaneously by different people in different rooms.

3 Broad Feature Set

When a light switch is used we look for an explanation by examining the state of the environment moments before the event occurred. This information is described by a feature set, in which each feature describes some portion of the environment. This feature set is composed of a mix of sensor-specific information and high-level domain knowledge. We start from a large set and narrow it down to the most predictive (see Table 1). We initially considered the following set of features :

– 6 Motion detector counts. These six numbers represent the number of times each of the six motion detectors has been triggered in the last 15 seconds.

Table 1. Feature set after feature selection

Number	Feature Name
1	Current Hour
1	Last Motion Detector
8	Light Switch Count
16	Transition Count

- 8 Light switch counts. These features represent the number of times each of the eight light switches have been manipulated in the last 15 seconds.

- 36 Transition counts. These features represent the number of times each transition occurred in the last 15 seconds. We recognize a *transition* when two motion detectors are set off consecutively as a result of a person moving from one room to another, or staying in one room long enough to set off the same motion detector twice. There are six motion detectors resulting in 36 transitions. Note that only a subset of these are physically contiguous.

- 1 Current hour. The current hour at the time of the event.

- 1 Light or dark. Using sunrise and sunset data we can determine whether it was light or dark outside at the time of the event.

- 1 Last motion detector. This number can take on a value of zero through six, representing which motion detector, if any, was triggered just before the event occurred.

4 Automatic Feature Selection

Feature selection was done by cross-validation and through the use of an independence test. For the features that represent counts (motion detector counts, light switch counts, and transition counts), we calculated the average mutual information between the class of an event and the absence or presence of each feature in the event. Using these methods we narrowed our feature set to a small collection of the most predictive features (see Table 1) .

Our most important feature is the last motion detector triggered before a light switch was used. In a home automation scenario lights in a room are turned on automatically when motion and darkness are detected. This is a common scenario for the same reason that this feature was important to our learner: people tend to turn on the lights when they enter a dark room.

The two features, current hour and light or dark, share some common ground. There is a strong correlation between the amount of light outside and light switch usage. The two features perform similarly, but discretizing the time into hours is a sacrifice. A trade-off exists between discretization and accuracy. To compensate we use kernel density estimation to combine information from nearby time slots. This is similar to how Flexible Bayes treats continuous features [10]. However, a problem arises from the circular nature of our discrete values (0,..,23). Applying a straightforward Gaussian kernel fails to ”wrap around” to include hours that are far apart, yet actually close (e.g., zero and twenty three). Our solution is to imagine a circle (like a clock face) with unit area. We divide the circle into twenty four angles, and convert hours into angles, α_t, measured clockwise from high noon. A Von Mises kernel has unit area over the circle and keeps a continuous derivative. The function is as follows:

$$\frac{exp(b\ cos(\alpha_1 - \alpha_2))}{2\pi J_0(b)}\ . \tag{1}$$

J_0 is a first degree Bessel function of the first kind. The variable b represents the width of the kernel. We set b = 10, so that our kernel encompasses about six hours. This value was chosen by cross-validation. The smoothed hour feature outperforms the unsmoothed hour and the light or dark feature (see Results section).

We found that the motion detector counts were less predictive than transition counts, despite significant noise contained in the transition readings. The two feature types share information, so we chose to discard the motion detector counts.

Of the 36 transition readings the most predictive corresponded to transitions between motion detectors in physically connected rooms. These are the transitions that occur most often, so it is not surprising that they should outperform rare transitions. There are six rooms in the house and five are physically connected (see Figure 1). These ten transitions, plus the six transitions of one room to the same room, were scored as most predictive. The rest of the transitions were discarded.

5 Preprocessing

The transition features were valuable to our learner, yet they include significant noise. The reason lies in how the transitions are gathered from the data stream. Let the data stream $d = \{e_1, ...e_N\}$, where e is an *event* composed of the time of occurrence t and the identity of the activated motion detector $m \in M$, $e_i = \{t_i, m_i\}$. The list of events are stored as they happen (in chronological order). In our dataset there are over 70,000 of these events. There may be large gaps of time between event occurrences, or events may happen simultaneously. Many people may simultaneously set off motion detectors in various rooms, so events are not listed with any regard to physical connectivity. The question is how to extract meaningful room transitions from a stream of chronological events.

A basic method is to simply pull pairs of events chronologically, without regard to their physical or temporal nearness. Although these noisy results are still predictive, there is room for improvement. We devised a better method by using a tree search approach similar to the minimax algorithm. We call it the transition finding algorithm (TFA).

The goal of TFA is to remove unlikely or impossible room transitions. It accomplishes this by using a scoring function to evaluate the likelihood of a particular transition occurring. TFA makes one pass through the list of events ($O(n)$ running time).

For each event, TFA tries to determine if it is noise; and if not, what event it should be paired with. While an exhaustive search of all possible pairings for a given input list is computationally infeasible, a greedy approach to pairing events would often reject the best overall set of pairings because it was not locally optimal. TFA uses a search method similar to the minimax algorithm to establish a balance between the low computational complexity of greedy algorithms and the optimal results guaranteed by an exhaustive search.

For a specified depth d and breadth b, TFA picks the best b scoring transitions that start with the current event e_0 (the event we are trying to find a match for). For each of those transitions, TFA temporarily marks it as being paired and recursively finds an optimal sum of scores for the next $d-1$ events. The transition starting with e_0 that leads to the highest overall sum of scores (for the block of d events) is the one that is output for e_0 (this could be no transition, if e_0 is noise).

The scoring function f takes two events and returns a non-negative real number. For our trials, an event consists of an *event number* n (i.e., the number associated with the motion detector for this event) and an integer t for the day and time the motion detector was set off (measured in seconds). Given two events $e_x = (n_x, t_x)$ and $e_y = (n_y, t_y)$ where $t_x \leq t_y$,

– If n_y is not physically adjacent to n_x, $f(e_x, e_y) = 0$.
– Otherwise, if $n_x = n_y$, then $f(X, Y) = f_s(t_y - t_x)$.
– If neither of the above conditions hold, $f(e_x, e_y) = f_r(t_y - t_x)$.

The two scoring functions, f_s and f_r, provide separate metrics for transitions between the same room and between different rooms, respectively. After a taking a reading the motion detector waits ten seconds before taking another one. This means that transitions between the same room tend to be ten seconds apart, and other transitions occur much more quickly, i.e., 1 or 2 seconds. We chose each of the scoring functions to be an exponentially decreasing function so that the score would quickly decrease as the time between events increases.

The scoring function for same room transitions is as follows: For $d \geq 10$, $f_s(d) = e^{\frac{10-d}{3}}$. Otherwise, $f_s(d) = 0$. The lower bound for d is 10 because the motion detectors will not recycle faster than 10 seconds. So the same motion detector cannot be set off twice within 10 seconds.

For different room transitions: $f_d(d) = e^{\frac{2-d}{4}} \sqrt{\frac{d}{2}}$ is a normalized GAMMA(1.5, 4) distribution. It was chosen because it reaches its maximum at $d = 2$, a reasonable time for walking from one room to the next.

5.1 TFA example

Here is an example where TFA would outperform a greedy algorithm. Given an input list of events (a, b, c, d) and the likelihood function in Table 2, TFA (with a search depth of two and breadth of two) would perform as follows:

– Examine the scores for $a \rightarrow b = 0.3$, $a \rightarrow c = 1.0$, $a \rightarrow d = 0.1$, choosing the best two (maximum breadth).

Table 2. Scores for transitions $x \to y$

		y		
		b	c	d
x	a	0.3	1.0	0.1
	b	-	0.9	0.0

- Recursively look at b, given $a \to b$ is a transition:
  The maximum depth has been reached, so return 0.9, the best of $b \to c = 0.9$ and $b \to d = 0.0$.
- Add the $a \to b$ score to the recursive score $(0.3 + 0.9)$ to get the overall $a \to b$ score: 1.2
- Recursively look at b, given $a \to c$ is a transition:
  The maximum depth has been reached, so return 0.0, the best of $b \to d = 0.0$ ($b \to c$ is not available because c is used in $a \to c$).
- Add the $a \to c$ score to the recursive score $(1.0 + 0.0)$ to get the overall $a \to c$ score: 1.0
- The best overall score is 1.2 for $a \to b$, so that transition is output. (Note that a greedy algorithm would have output $a \to c$, leading to a lower overall score.)
- TFA then moves to event b and uses the same procedure to match it in a transition; and so on, through the end of the event list.

(An actual run of TFA uses a much longer event list and significantly larger breadth and depth, to achieve greater precision.)

6 Approach

6.1 Learner

We use a Bayesian learner to predict light switch usage patterns. Bayesian techniques have been employed in a variety of problem domains [15]. This approach assumes that a parametric model can describe light switch usage patterns. We use training data gathered from our sensors to find Bayes-optimal estimates of the model parameters. Classification on new instances is done by using Bayes' rule to calculate the posterior probability of each class and choosing the most likely one. We also use information from the posterior probabilities to predict whether a light switch is likely to be used at any given time. The learning task is simplified by use of the naive Bayes assumption. This assumes independence between attributes, allowing parameters for each attribute to be learned separately.

6.2 Probabilistic Framework

Our Bayesian learner uses a multinomial mixture model similar to the one described by [15]. The model is parameterized by θ and composed of mixture components $c_j \in \{C\}$

. We assume a one-to-one correspondence between classes and mixture model components. We use c_j to indicate the jth mixture component and the jth class. Therefore, an event e_i is described by using class priors to choose a class c_j. That mixture model is used to generate an event according to the learned parameters, with distribution $P(e_i|c_j)$. The likelihood of an event, e_i, is the sum of probabilities over all mixture components:

$$P(e_i|\theta) = \sum_{j=1}^{|C|} P(c_j|\theta) P(e_i|c_j;\theta)\ . \tag{2}$$

We use the maximum likelihood estimate to describe the class prior parameters:

$$P(c_j|\hat{\theta}) = \frac{\sum_{i=1}^{|E|} P(c_j|e_i)}{|E|}\ . \tag{3}$$

Define $P(c_j|e_i) \in \{0,1\}$ as given by the instance's class.

6.3 Model

The multinomial distribution naturally handles attributes that represent the frequency of predictive activities. An event is recorded each time a light switch is manipulated. We represent an event by a vector of activity counts, $e_i = \{x_1, ..., x_{|X|}\}$. These are provided by our transition count and light switch count features. Define x_{it} to be the number of times activity x_t occurs during event e_i. If a person walks from the living room to the dining room and then back, we see an event of the form $e_i = \{1, 1, ..., 0\}$, where the ones indicate the transition from living room to dining room, and from dining room to living room. With a large enough window of time the order in which events occurred is lost.

We are given a set of labeled training instances, $E = \{e_1, ..., e_{|E|}\}$. The probability of an event given each class is the multinomial distribution:

$$P(e_i|c_j;\theta) = \sum_{t=1}^{|X|} P(x_t|c_j;\theta)^{x_{it}}\ . \tag{4}$$

The parameters of each class are the probability of an attribute x_t in a class c_j. We use a Laplacian prior to prevent probabilities of zero or one:

$$P(x_t|c_j;\hat{\theta}_j) = \frac{1 + \sum_{i=1}^{|E|} x_{it} P(c_j|e_i)}{|X| + \sum_{k=1}^{|X|} \sum_{i=1}^{|E|} x_{ik} P(c_j|e_i)}\ . \tag{5}$$

7 Classification

7.1 Classification

There are around eighty five thousand seconds in a day, and on average twenty five light switch manipulations. The overwhelming number of non-events pose a problem to our probabilistic learner. Although most real-time data represents a non-event, we are interested in discriminating between eight classes of events (corresponding to the eight light switches). Hence, for now we ignore non-events and formulate a classification problem consisting only of event classes. Later analysis will be used to recognize an event from a non-event.

During the training phase we estimate model parameters from labeled examples. Given a test instance, we select the most likely classification by calculating the posterior probability of each class given the evidence of the test instance, and choose the class with the highest probability. This follows according to Bayes' rule:

$$P(c_j|e_i;\hat{\theta}) = \frac{P(c_j|\hat{\theta})P(e_i|c_j;\hat{\theta}_j)}{P(e_i|\hat{\theta})} \; . \tag{6}$$

7.2 Detection & Classification

We wish to (1) predict whether a light switch will be manipulated during some time window, and (2) predict which light switch it will be. The learner is presented with a decision (i.e., a test instance) every second. The learner does not know whether the test instance is an event or a non-event, although usually it is a non-event. We observe a correct prediction when the learner predicts a correct occurrence within a w second window of the actual occurrence. We report results with the time window w set to 30 seconds.

Earlier we avoided learning a class to describe non-events, so now we must take another approach. During initial training we calculated posterior probabilities for each class over many training instances. We can average those values to find a threshold for each class. Each time the learner receives a new instance we calculate the posterior probabilities of each class. If a class posterior probability rises above its respective threshold then the system logs that class of event as having occurred. If that event does not actually occur within the time window w of the prediction, a false positive occurs. If the system fails to predict an event then a false negative occurs. Finally, if an event does occur, but is different from the predicted event, then a false positive and a false negative occur.

7.3 Outlier Detection

Once a model of normal user behavior is learned we can detect abnormal patterns. For example, this could be used to warn when a patient has a marked decrease in activity level, which can be indicative of depression [22]. Our approach to outlier detection is to compare previously learned posterior classifications to actual occurrences. When an event occurs that has a low posterior, it connotes a surprise inversely proportional to the size of the posterior. By thresholding this value we can raise a red flag for very ”surprising” behavior.

8 Results

Two months of data were collected, corresponding to approximately 1500 light switch manipulations (around twenty five per day). Evaluation differs for the classification task vs. the detection plus classification task.

8.1 Classification

Table 3. Naive Bayes classification results

Dataset	Unsmoothed	Smoothed
Naive Bayes TFA	74.43% $\pm$ 0.016	77.62% $\pm$ 0.014
Naive Bayes	71.04% $\pm$ 0.024	75.57% $\pm$ 0.019
Home automation	34.62% $\pm$ 0.011	

The problem is to classify an instance as one of eight classes, with no possibility of a non-event. A low baseline strategy of choosing the most used light every time (bathroom), returns 26.53% accuracy. A better baseline mimics a home automation strategy. If it is dark outside this baseline randomly turns on all lights in the room in which motion was last detected. This home automation baseline reflects the state of the art, and returns an accuracy of only 34.62%. Our learner performed training and testing on a jacknifed dataset, testing on 10% and training on 90% of the dataset. The final mean accuracy was 75.57% with variance 0.019. See Table 3 for a summary of all experiments.

8.2 Detection & Classification

There is no simple baseline for evaluating detection and classification results. Instead, we look at false positive and false negative rates. The learner was trained on nearly two months' of event data and then given seven days of never before seen real-time test instances. There was one test instance per second for the seven days, summing to over six hundred thousand instances. Because the look-back window for collecting attributes is often short, the vast majority of test instances contained no data. These ”empty states” were thrown away, causing 6% of actual events to be ignored outright (false negatives). 81.17% of events were correctly predicted to occur, although only 66.25% of these were correctly classified. The learner executes a late or incorrect prediction about once every six and a half hours.

9 Related Work

Over the last several years much effort has been put into developing and employing sensor arrays applicable to a variety of domains, including camera networks for people tracking [3], [20], facilitation of group meetings [6], vibration sensors for accident detection [2], wearable sensors [14], and software simulated environments [13]. Unfortunately, most prior work does not suit the special needs of healthcare facilities.

Our work is akin to the Neural Network House [17], [18] in that we use distributed sensors to monitor and predict activity in a house. However, the neural networks employed by Mozer specialize in negotiating between energy conservation and comfort. Our system probabilistically learns normal patterns of activity for monitoring and prediction.

10 Discussion and Future Work

If the system were used to mimic human actions, then occupant behavior would be modified and presumably any learned patterns would be affected. Even if we do not affect occupant behavior, we cannot assume stationarity of our environment. We are currently exploring a reinforcement learning solution, using a variety of implicit and explicit user feedback for reward and punishment. By using the probabilistic framework described here as a starting point, a reinforcement learner could continuously adapt to user needs, with a much smaller learning curve.

This paper explores the first steps towards the larger goal of automatic monitoring with low-cost sensors. We are currently instrumenting the assistive intelligent environment with contact switches on doors and windows, vibration sensors on counter tops, pressure mats in couches and beds, as well as a variety of miscellaneous sensors (such as a toothbrush-use sensor). With more complex information we hope to detect and predict more complex behavior with higher accuracy.

11 Conclusion

We explore a low-cost, low-technology automatic monitoring application. We used motion detectors and light switch sensors to predict light switch manipulation in a house with two occupants. A probabilistic learning framework was described. We showed that our learner can accurately detect and predict light switch usage. Our results show significant improvement over a home automation inspired baseline, and we discussed future work that could lead to even greater improvement.

References

1. Abowd, G. and Mynatt, E. (2000). Charting Past, Present, and Future Research in Ubiquitous Computing. ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, Pages 29-58, March 2000.
2. Abowd, G., Atkeson, C., Bobick, A., Essa, I., MacIntyre, B., Mynatt, E., and Starner, T. (2000). Living Laboratories: The Future Computing Environments Group at the Georgia Institute of Technology. Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, Netherlands, April 1-6, 2000.
3. Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000). EasyLiving: Technologies for Intelligent Environments.

4. Burgio, L., Scilley, K., Hardin, M., Janosky, J., Bonino, P., Slater, S., and Engberg, R. (1994). Studying Disruptive Vocalization and Contextual Factors in the Nursing Home Using Computer-Assisted Real-Time Observation. Journal of Gerontology, Vol. 49, No. 5, Pages 230-239.
5. Burgio, L., Scilley, K., Hardin, M., Hsu, C. (2001). Temporal patterns of disruptive vocalization in elderly nursing home residents. International Journal of Geriatric Psychiatry. 16, 378-386.
6. Cohen, M. (1999). The future of human-computer interaction or how I learned to stop worrying and love my intelligent room. IEEE Intelligent Systems. March/April 1999.
7. Davis, L., Buckwalter, K., Burgio, L. (1997). Measuring Problem Behaviors in Dementia: Developing a Methodological Agenda. Adv. Nurs. Sci, 20(1),40-55.
8. Huang, A., Ling, B., Ponekanti, S., and Fox, A. (1999). Pervasive Computing: What Is It Good For? Communications of the ACM. 84-91. 5/1999.
9. Huberman, B. and Clearwater, S. (1995). A multi-agent system for controlling building environments. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95), pages 171-176.
10. John, G., and Langley, P. (1995) Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo.
11. Kortuem, G., Segall, Z., and Bauer, M. (1998) Contextaware, adaptive wearable computers as remote interfaces to 'intelligent' environments. In Second International Symposium on Wearable Computers, Pittsburgh, PA, October, pp. 58-65, IEEE Computer Society.
12. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. (2000). Multi-Camera MultiPerson Tracking for EasyLiving. 3rd IEEE International Workshop on Visual Surveillance, Dublin, Ireland, pp. 3-10.
13. Lesser, V., Atighetchi, M., Benyo, B., Horling, B., Raja, A., Vincent, R., Wagner, T., Xuan, P., and Zhang, S. (1999). A multi-agent system for intelligent environment control. In Proceedings of the Third International Conference on Autonomous Agents. ACM Press.
14. Mann, S. (1996). 'Smart Clothing': Wearable multimedia computing and 'personal imaging' to restore the technological balance between people and their environments. Proceedings of ACM Multimedia. ACM Press, pp. 163-174.
15. McCallum A. and K. Nigam. (1998). A comparison of event models for naive bayes text classification. AAAI-98 Workshop on Learning for Text Categorization.
16. McNeil, J. (1997). Disabilities affect one-fifth of all Americans: Proportion could increase in coming decades. U.S. Census Brief.
17. Mozer, M. C. (1999). An Intelligent Environment Must be Adaptive. IEEE Intelligent Systems and their Applications, 14(2), 11-13.
18. Mozer, M. C. (1998). The neural network house: An environment that adapts to its inhabitants. In Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments (pp. 110-114). Menlo, Park, CA: AAAI Press.
19. SmartHome. Networked home products using X10 technology. http://www.smarthome.com/.
20. Stillman, S., Tanawongsuwan, R., and Essa, I. (1999). A system for tracking and recognizing multiple people with multiple cameras. In Proceedings of Second International Conference on Audio-Visionbased Person Authentication, Washington, DC.
21. Trivedi, M., Kohsia, H., Mikic, I. (2000). Intelligent Environments and Active Camera Networks. IEEE Conf.
22. SMC VanHaitsma, K., Lawton, M.P., Kleban, M., Klapper, J., and Corn, J. (1997). Methodological Aspects of the Study of Streams of Behavior in Elders with Dementing Illness. Alzheimer Disease and Associated Disorders. Vol. 11, No. 4, pp. 228-238
23. Vigil Incorporated . The Vigil Integrated Care Management System . http://www.vigil-inc.com/.