Continuous Multimodal Authentication Using Dynamic Bayesian Networks

Justin Muncaster and Matthew Turk
Computer Science Department, University of of California, Santa Barbara
{jmunk, mturk}@cs.ucsb.edu

Abstract

In this paper we address the issue of post-login user verification using biometrics. We propose a general framework for continuously monitoring a user and his or her characteristics throughout the session in order to provide continuous verification of identity. We present a multimodal approach using dynamic Bayesian networks to account for classification uncertainty and to encode the system's dynamic model. The system supports an extendable number of modalities that can be integrated at both the score-level and the decision-level. The system can also be extended to incorporate important contextual information. We provide a simple way for an operator to tailor the dynamic model using known domain knowledge. Initial tests with the system using face recognition and keystroke dynamics are promising, and we expect performance to improve as more modalities are incorporated.

1. Introduction

Most biometrics for computer systems work by authenticating a user during login and then operate under the assumption that the system is secure from that point forward. This type of verification is lacking because it fails to address a security risk that may arise after the system has been accessed. Turk and Altinok [1] initially addressed the shortcomings of a "one-shot" approach by identifying two assumptions of such systems: (1) a user can be verified at a single point in time, and (2) the user remains constant for the duration of the session. These assumptions are likely to be invalid in many contexts. For example, the number of required biometrics to verify identity at login may be impractical in a low-security environment, and the user may change after login in a hostile environment. Security under the above assumptions cannot adequately defend against the practice of "hijacking," where an attacker targets a post-authenticated user. A continuous approach of establishing identity is desirable in these circumstances so that a system can be monitored for the duration of the session.

Although such a system will likely require strong, active biometrics to initially authenticate a user, we do not wish to simply ask a user to constantly submit to an intrusive biometric measurement. To achieve continuous verification a system should make heavy use of passive biometrics [2] that do not require a user to actively provide a measurement. Passive biometrics, such as face and keystrokes, vary in their ability to identify a user, and their performance frequently depends on contextual information. A face recognizer, for example, may not be able to provide an accurate classification if lighting conditions are not suitable. Keystroke analysis can provide many measurements; however, the accuracy of the classifier may be relatively poor. Because of these tradeoffs in accuracy, multimodal techniques [3,4] have gained popularity due to their ability to exploit the benefits of one biometric while mitigating the inaccuracies of another.

Different techniques of multimodal integration have been investigated thus far. In [2], Jain identified that biometric integration can occur on the feature level, the score level, and the decision level. In feature level integration all of the initial features from measurements are grouped together into a single feature vector for classification. Although the most information is available at this point, feature-level integration suffers from the so-called curse of dimensionality. Additionally, the features of some measurements may not always be available. Score-level integration does integration later in the pipeline when a classifier has output some "match" score for a particular biometric. Jain [5] has investigated many different techniques for integrating different scores. Finally, integration can occur at the decision level when many biometric classifiers make a hard decision on whether the biometric matched the user or not. At this point the least information is available, but inferences can still be drawn using voting techniques.

Turk and Altinok [1] attempted to achieve a continuous, score-level multimodal system by classifying based on a weighted sum of scores from each modality. The weighting factor was chosen in such a way that would capture the reliability of the modality, and was decreased monotonically with the

time since the last measurement. This allowed the system to account for the uncertainty that arises due to the age of the measurements.

Recently, Zhang *et al*. [6] developed a continuous multimodal biometrics system using a hidden Markov model (HMM). In this work, the user's identity was the sole hidden variable that would "emit" biometric measurements. The authors were successful in integrating results from a fingerprint biometric classifier with a face classifier and developed a model that intuitively separated the uncertainty that arose from the dynamic model from the uncertainty that arose due to the sensor model.

In this paper we take an approach similar to [6], but we use a Dynamic Bayesian Network (DBN) to achieve continuous multimodal biometrics. The advantage of a dynamic Bayesian network is its ability to account for many hidden variables, as opposed to the single hidden variable that an HMM typically models. By modeling more hidden variables, DBNs are capable of modeling important contextual information. In addition, our approach is indifferent to whether integration occurs at the score-level or at the decision-level. Our approach will treat sources of evidence as black boxes from which we can receive either a score or a decision.

First, we will discuss DBNs and their ability to model multivariate probability distributions efficiently. Next, we will talk about how the parameters of a DBN can be set using expert knowledge and learning techniques. Next we will define *classifiers*, our source of information for integration. Finally we will introduce our preliminary experiment involving a simple DBN to perform decision-level integration.

2. Bayesian Networks

2.1. Definition

Bayesian networks [7] provide a compact way to encode multivariable probability distributions. A Bayesian Network $B = \langle G, \Theta, X_V \rangle$ is a directed acyclic graph G, a parameter set Θ, and a set of random variables X_V. The graph $G = (V, E)$ has *n* vertices, given as $V = \{v_1, ..., v_n\}$. There is a one-to-one mapping between vertices in V and random variables in X_V. For this paper we will assume the corresponding random variables are discrete and are given by $X_V = \{X_1, ..., X_n\}$ and that the mapping is done in the obvious way.

An edge $(u, v) \in E$ in a Bayesian network denotes a direct relationship between X_u and X_v. The absence of such an edge is captured mathematically as conditional independence between the respective random variables. Many conditional independencies are implied by the structure of G. These have been exploited by various researchers [7,8,9] to make interesting computations tractable. Formally, a variable X_u is conditionally independent of X_v given X_w if u is *d-separated* from v by w. For a complete explanation of d-separation, see [7].

Before we define the parameter set, we must introduce some notation. Given a set of m vertices $U = \{u_1, ..., u_m\}$ we will denote the joint random variable corresponding to all of the vertices as $X_U = X_{u_1} \times ... \times X_{u_m}$. Given any random variable we will denote the probability $P(X_U = x_U)$ as $P(x_U)$ when the context is clear.

To finally specify the Bayesian network we require the parameter set $\Theta = \{\theta_v\}^{v \in V}$. These parameters are given by a set of conditional probability distributions.

$$\theta_v = P(x_v \mid x_{pa(v)}) \tag{1}$$

In (1), $pa(v)$ denotes the parents of *v*, or the set of vertices such that for each $u \in pa(v)$ there is a directed edge from u to v. In other words, for each variable X_v that is directly influenced by variables $X_{pa(v)}$, we require the probability of the state of X_v given the possible states of the direct influences. If X_v and $X_{pa(v)}$ are discrete, as we are assuming for this paper, we refer to (1) as a conditional probability table (CPT).

Given the parameter set and the conditional independencies implied by d-separation in G, we can completely specify the joint probability over X_V as

$$P(x_V) = \prod_{v \in V} P(x_v \mid x_{pa(v)}) = \prod_{v \in V} \theta_v .$$

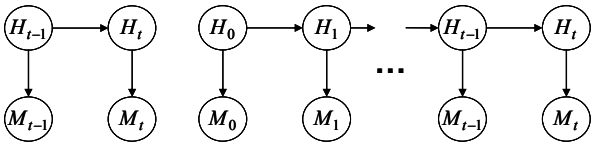
Probabilistic inference refers to the computation of marginal probabilities of particular variables given some evidence. That is, once we have some evidential observations $X_\xi = x_\xi$ we would like to compute the marginal probability of each random variable given the evidence, or $P(x \mid x_\xi)$, $x \in X_V$. This computation can be computed efficiently on tree-structured networks using Pearl's celebrated belief propagation algorithm [7]. Cooper [10] has shown that for graphs that are multiply connected inference is NP-hard in general; however general methods have been developed by researchers [8,9] that exploit conditional independency and have practical computational complexity on a large number of problems.

For notational simplicity, for the rest of this paper we will refer to vertices v and random variables X_v interchangeably by giving them the same name and relying on context to disambiguate.

2.2. Dynamic Bayesian Networks

A Bayesian network that models time-varying phenomena is called a Dynamic Bayesian Network (DBN) [17]. In a DBN, some variables are stochastic and change over time. To specify a DBN, one creates two time-slices of a traditional Bayesian network and connects time-varying nodes with an edge. As time passes the network is "unrolled" to extend the model.

A Hidden Markov Model (HMM) is a special case of a DBN in which there is a single time-varying hidden variable and a single measurement variable per time-slice. Respectively, the random variables are $X = \{H_{t-1}, M_{t-1}, H_t, M_t\}$. Figure 1(a) depicts two slices of this HMM and Figure 1(b) shows the unrolled HMM.



(a) (b)

Figure 1. (a) Two time-slices of an HMM, a special case of a DBN. (b) The unrolled HMM.

To compute $P(H_t \mid M_t, M_{t-1}..., M_0)$, the marginal probability of the hidden state given all of the measurements, one can apply a recursive formula that relies on the Markov assumption, which asserts $P(H_t \mid H_{t-1},..., H_0) = P(H_t \mid H_{t-1})$. When viewed as dynamic Bayesian network, the Markov assumption comes out of the conditional independencies implied by the network. Specifically, since H_t is d-separated from $H_{t-2},..., H_0$ by H_{t-1}, the Markov assumption arises naturally out of the DBN structure.

The key difference between a DBN and an HMM is the fact that a DBN can model many hidden variables and the dependencies (or lack thereof) between them. This is useful because we can model many time-varying phenomena and the interactions between variables. Additionally, we have all of the benefits of Bayesian networks and thus we can efficiently account for the effects of contextual information. An example of a DBN that could be used for biometric authentication is shown in Figure 2.

Inference in a DBN is intractable in general but practical in many useful cases. Murphy [11] has examined many exact and approximate algorithms for probabilistic inference in a DBN.

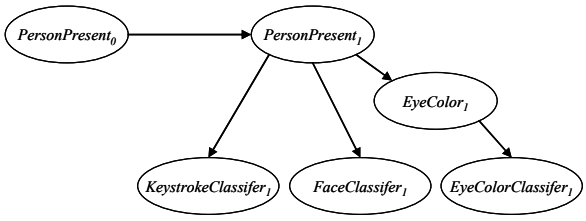


Figure 2. Example DBN for biometric authentication. The states of *PersonPresent* are {*genuine, imposter, absent*}.

3. Setting Model Parameters

3.1. Transition matrix for time-varying variables

Assume that we have a variable X_t that varies with time, and that X_t only depends on its previous state, X_{t-1}. Let there be k states in both X_{t-1} and X_t which are denoted by the integers from $1...k$. Due to the edge from X_{t-1} to X_t, we need to specify the state transition matrix $P(X_t = j \mid X_{t-1} = i)$ given some time interval Δt. This transition matrix will be a square matrix, where entry $P_{ij} = P(X_t = j \mid X_{t-1} = i)$.

Assessing this information directly can be difficult. For example, suppose we were to ask a domain expert, "What is the probability that in the next 5 seconds the user is still present? What is the probability that in the next 5 seconds the genuine user leaves the desk and an imposter is in his place?" The fact that there is a specific time interval involved it makes it very difficult to elicit an accurate answer to the above questions. We offer a more systematic approach to setting parameters.

We start by assuming that the processes by which X_t changes states is a Poisson process. This asserts that over the time interval between X_{t-1} and X_t, the probability of changing states is constant. Under this assumption it can be shown that given some time Δt the probability that a change from state i occurs will follow the exponential cumulative distribution with rate parameter λ_i, giving

$$P_{switch_i} = P(X_t \neq X_{t-1} \mid X_{t-1} = i) = 1 - e^{-\lambda_i \Delta t} \qquad (2)$$

And, consequently, the probability of staying in state i is

$$P_{stay_i} = 1 - P_{switch_i} = e^{-\lambda_i \Delta t} \qquad (3)$$

One important thing to note here is that we assume the probability of a state change is *constant* over the entire time interval Δt. Given a large time interval we do not expect this to be the case so we will insist that Δt is small enough to roughly satisfy our assumption. A consequence of this is that we cannot wait until a new measurement arrives to evaluate (2) and do the Bayesian updating because the amount of time between measurements could be large. Thus, we will choose an update rate that is small and repeatedly do updates.

This does *not* mean, however, that we must take *measurements* at a very fast rate because we may be limited by classification speed and the availability of measurements. We choose a small update rate so that we can transition hidden variables through the dynamic model more accurately than if we waited a long time to do probabilistic inference.

An important question lies in how to ascertain a rate parameter, λ_i, using expert knowledge relevant to the application. One piece of information that an expert may know is the average time that a variable may stay in a state i, which we will denote $\hat{\mu}_i$. We know that the maximum-likelihood estimator for the rate parameter of an exponential distribution is $\lambda_i = \hat{\mu}_i^{-1}$, where $\hat{\mu}_i$ is the average time spent in state i. By substituting this into (2) we can estimate P_{switch_i} and P_{stay_i}.

Finally, we are ready to estimate P_{ij}. Since the state must either *stay* in the same state or *switch* to a new state, using total probability and the chain rule we have,

$$\begin{aligned} P_{ij} = {} & P_{switch_i} \cdot P\left(X_t = j \mid X_{t-1} = i, switch\right) \\ & + P_{stay_i} \cdot P\left(X_t = j \mid X_{t-1} = i, stay\right) \end{aligned} \tag{4}$$

Notice that the event *stay* occurs when $i = j$, corresponding to the diagonal elements in the transition matrix. When this happens the first term in (4) goes to zero and the expression reduces to P_{stay_i}.

A *switch* occurs when $i \neq j$ and corresponds to the off-diagonal elements. When this happens the second term in (4) goes to zero and the expression becomes $P_{switch_i} \cdot P\left(X_t = j \mid X_{t-1} = i, switch\right)$. Thus when we switch states we require the probability of going to each other state, independent of time.

$$P_{i \to j|sw} = P\left(X_t = j \mid X_{t-1} = i, i \neq j\right)$$

This gives our final transition matrix, after substituting a particular Δt into the appropriate expressions. By using expert knowledge to estimate P_{switch_i} and $P_{i \to j|sw}$, we have effectively dealt with the problem of estimating probabilities over specific time intervals.

$$P_{ij} = \begin{cases} P_{stay_i} & i = j \\ P_{switch_i} \cdot P_{i \to j|sw} & i \neq j \end{cases} \tag{5}$$

3.1. Other model parameters

Estimating model parameters from expert knowledge in most other cases is straightforward. For hidden variables, such as *EyeColor*, the information is known beforehand for the genuine user. When the parameters are not known, such as an imposter's eye color, the unknown parameter can be assessed using prior information (say, the distribution of the general population's eye color) as well as learned [12] as data arrives by using techniques like expectation-maximization.

4. Classifiers

In the context of a DBN, the result of a classifier is just another random variable. Input data goes into a "black box" and out comes a result. We assume that given the data available, the output of this box depends only on a small subset of random variables in our DBN. In biometric applications the output of a classifier on certain data will usually depend only on the identity of the user, however this need not always be the case. If we are able to classify other pieces of information, e.g., a soft biometric like eye color, then we can integrate that information into our probabilistic model as well.

We will treat all input data to a classifier symbolically as y. We will define a classifier $c(y; X)$ as a function that when given appropriate input data will emit a classification over some range that depends solely on the value of a random variable, X. If the range of c is discrete (e.g., $\{genuine, imposter\}$), we will call the classifier a *decision-level classifier*. If the range of c is continuous (e.g., a score between [0,1]) we will call the classifier a *score-level classifier*.

A requirement of c is that when given input data y we are able to estimate $P(c(y) \mid X = x)$. For decision level classifiers this will be a set of multinomial distributions and being discrete it will behave just like any other variable in a Bayesian network. To estimate the model parameters $P(c(y) \mid X = x)$ we can look at training data and use a standard maximum-likelihood estimation technique.

For example, suppose we have a face-recognizer $FaceClassifier(y; PersonPresent)$ that has two possible outcomes which depend on who the present person is (either *genuine* or *imposter*). The $FaceClassifier$ will just identify that the person present is genuine as either '+', or "-". To estimate $P(FaceClassifier(y) | PersonPresent)$, we must iterate through all of the cases where the user is a genuine user and count the frequency of correct classification and false-negatives, giving $P(PersonPresent(y) | genuine)$. Additionally, we must go through all of our training data for non-genuine users and count the correct classifications and false-positives, giving $P(PersonPresent(y) | imposter)$. This gives the conditional probability table for $FaceClassifier$.

If we have a score-level classifier then the output of the classifier is a real number. It is usually some similarity measure to whatever is being recognized like, for example, the genuine user. Although the output of a classifier is continuous, we are able to incorporate the likelihoods from evaluating $P(c(y) | x)$ using standard inference algorithms. Thus, we need to be able to estimate $P(c(y) | x)$.

Estimating a one-dimensional density is a problem in itself and we do not address it in this paper. In [6], the authors provided this probability for a face-detector by using histograms of the biometric scores in question. Jain attempted to estimate the same distribution in [5] using Parzen-windows [13]. The important finding here is that both decision-level classifiers and score-level classifiers are dealt with in the same probabilistic framework, regardless of how this density is estimated.

5. Preliminary results

5.1. The model

For our preliminary experiments, we chose to use a simple model that integrates face recognition with keystroke dynamics. We chose these biometrics because both of them are passive and very different in nature. The accuracy of face recognition is highly dependent on the imaging conditions; keyboard dynamics requires data collected over a much longer period of time. In general, the accuracy of these modalities may be low enough so that long-term integration is useful. We use a traditional login procedure to establish identity early on and then fall back on continuous integration of weaker biometrics.

The Bayesian network used is shown in Figure 3. In this preliminary work, we are not yet utilizing the full potential of a DBN model. In particular, we do not have important pieces of contextual information that could be relevant to how the user operates. Possible pieces of context could include time of day, security alert levels, and other environmental conditions that may affect certain classifiers.

Because many variables were omitted, our network is very similar to the HMM given in [6]. A key difference, however, lies in the user model where we choose to explicitly model the absence of a user. By doing this and using expert knowledge, we expect to have a more accurate model of the dynamics of the system. One would expect, for example, that in a hijacking attempt a malicious user would attempt to seize the system when the genuine user leaves it unattended. By keeping track of the possibility that a user left the system for a brief time, our prior probability of an attack will be appropriately adjusted.

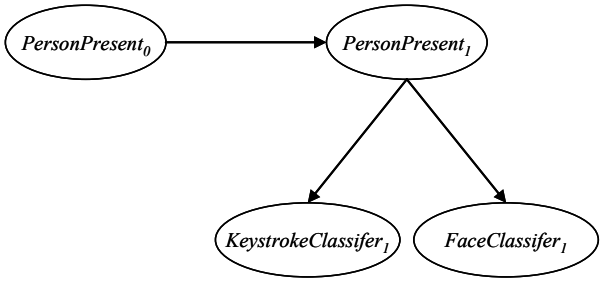


Figure 3. The DBN for our initial tests.

We chose to model a user using a secure computer for a period of 30 minutes. Our two sources of evidence were pictures taken of the user and the typing habits of the user. Pictures were taken of the user every 40 seconds on average, with some mild perturbation of the time. The measurements were taken far apart so that we can reasonably assume their independence. Keystrokes were monitored and a classifier was used to emit classifications when sufficient data was available.

We assumed the following parameters to the dynamic model, where "gen", "imp", and "abs" denote genuine, imposter, and absent, respectively:

$$\begin{array}{lll} \hat{\mu}_{gen} = 30\,\text{min} & P_{gen \to imp|sw} = 0.01 & P_{gen \to abs|sw} = 0.99 \\ \hat{\mu}_{imp} = 10\,\text{min} & P_{imp \to gen|sw} = 0.00 & P_{imp \to abs|sw} = 1.00 \\ \hat{\mu}_{abs} = 15\,\text{min} & P_{abs \to gen|sw} = 0.50 & P_{abs \to imp|sw} = 0.50 \end{array}$$

This model asserts that it is very unlikely that an imposter forcibly removes the genuine user and takes over ($P_{gen \to imp|sw}$). It also says that if an imposter is going to switch states, it is certainly going to be the case that the imposter got up and left. The reason for this is that if we have a case where we go from

imposter to *genuine*, then we have already caught the hijacker and there is no need to do further inference. Finally, when no user is present and a new user is going to use the system, we consider ourselves to be a total state of uncertainty with respect to who the user is. Depending on the threat to the system, this probability may be adjusted by a system operator.

Due to the ability to tune parameters we are able to create a system that is tolerant of classification errors if there is continuity with regard to the person present but sensitive to new classification results if there is block of time where the system was absent of a user.

We acquired testing data by creating a set of "virtual users" whereby we combined keystroke data from one person with image data of another. This technique is justified since these two modalities can safely be assumed independent. The image data was acquired from [14] and the keystroke data was acquired from users in previous experiments.

To test the complete system, a total of 15 images and 30 minutes of typing data from the genuine user were set aside and were never seen in the training process. Additionally, we chose another individual who not in any the training set to be the imposter and 15 images and 30 minutes of typing data were set aside for this person as well.

5.2. The classifiers

To do face recognition we chose to use the basic *eigenface* approach described by Turk and Pentland [15]. We assumed that we have a face detector available that could find a face and segment it properly; hence all of our recognition was done with face images.

We trained the system on 10 individuals, each with 5 images. A particular individual was labeled as the genuine user and the remaining 9 were labeled as imposters. Training and performance evaluation was done by leaving one image out of the data set, training the classifier on the remaining 4*10 images, and then testing the left-out images. Recognition was done by projecting the test image onto the mean image using the first 10 principal components, and then choosing the nearest-neighbor with respect to the Euclidean distance of the weights of the components. The classifier would output "+" if the nearest-neighbor was the genuine user and "-" otherwise.

Since the emphasis of this paper was not on face recognition we did not desire to have a state-of-the-art face recognizer. Unfortunately, despite our classification approach the data we used was so well controlled that our initial results for face recognition were too good, making multimodal integration unnecessary. Thus, to simulate a weaker classifier we simply corrupted our data with added artificial noise. In a more controlled experiment we would obviously prefer to use a more realistic data set to assess classifier performance.

Our keystroke classifier was done using a support vector machine (SVM) and used keyhold latencies as well as time between keypresses as features. The approach, as well as the training of the system, is described in [16]. Like the face classifier, the classifier outputs "+" if the user is classified as *genuine* and "-" if the user is classified as *imposter*.

The classification accuracy of the two biometrics on the training data is shown in first two rows and columns of the CPTs in Table I. Because we explicitly model the possibility that a user is *absent*, we must consider this case in the conditional probability table for each classifier. To do this we added a new state to each classifier, "$\varnothing$", which denotes when there is no classification. Because of its semantic meaning, a classifier will never actually output "$\varnothing$". Likewise, if no user is present a classifier will never output "+" or "-". This asserts the following:

$$P(FaceClassifer = \varnothing \mid PersonPresent = gen) = 0$$
$$P(FaceClassifer = \varnothing \mid PersonPresent = imp) = 0$$
$$P(FaceClassifer = + \mid PersonPresent = abs) = 0$$
$$P(FaceClassifer = - \mid PersonPresent = abs) = 0$$

And due to total probability,

$$P(FaceClassifer = \varnothing \mid PersonPresent = absent) = 1$$

This has the desired affect of updating our model with the information that a user is definitely present when a classification arrives, which is consistent with our intuition (if a keystroke classification occurs, *someone* must have pressed the key). If this assumption is false (e.g., the face classifier sometimes classifies the background as a face) then we would have to update the above probabilities appropriately.

	Face Classifier			Keystroke Classifier		
	+	−	∅	+	−	∅
genuine	0.71	0.29	0	0.47	0.53	0
imposter	0.35	0.65	0	0.24	0.76	0
absent	0	0	1	0	0	1

Table I. The conditional probability tables for two classifiers.

Note that we will *not* use the absence of a measurement as an evidential finding, because that would lead to the incorrect conclusion that no user is present if no measurement occurred. Clearly, this is not true and we would rather not give any evidence and let

the dynamic model account for the possibility that no user is present.

5.3. Results

To test the viability of our approach we considered two possible scenarios. In the first scenario, we simply have a genuine user who uses the system as usual. We would like our system to maintain a high degree of belief that the user is genuine despite the possibility of errant classifications.

In our second scenario we chose to place our virtual users such that a genuine user initially logs into a system and uses it for 10 minutes, at which point he leaves it unattended. At 20 minutes into the scenario, an imposter comes in and attempts to use the system for the remaining time. We would like to be able to recognize this hijacking.

The state of belief of the current user with respect to time in the first test scenario is shown in figure 4. An important thing to notice is the system is robust to classification errors. Although both biometrics make mistakes, the dynamic model is able to mitigate the mistakes by providing a strong prior that the user is indeed genuine. Because it is unlikely for the person to just "flip" from one person to another, the probability of an imposter remains low despite classification errors.

Figure 5 shows the results of the mock hijacking scenario. In this scenario the genuine user logs into the system and uses it for a short time. At this time the genuine user leaves the system unattended and a hijacker enters.

Notice how the beliefs that the user is absent increase during the time that the system is not receiving any measurement. This then affects the prior probability that an imposter can enter the system. When the new measurements provide support for this hypothesis, our system is more sensitive to the negative classifications and we are able to integrate information and detect the hijacking. It must be noted that indeed there were no classification errors for the face classifier in this case, however we chose to show this example to illustrate the sensitivity to negative measurements given an uncertain prior. More rigorous work needs be done to assess robustness to classification errors in this situation, however our experience with the system is that errors are overcome in time.

6. Conclusion

We have proposed the use of dynamic Bayesian networks to do continuous temporal integration of biometric measurements. We showed how to create a model and estimate its parameters using expert knowledge. Our technique is independent of the classification technique and can integrate on both the decision level and the score level, although our tests only integrated decision-level classifications. Our preliminary results show promise and warrant future work to develop a more controlled experiment, investigate the use of additional modalities, and utilize contextual information to test and hopefully improve the robustness of our model.

7. Acknowledgement

Partial support provided by NSF IGERT Grant# DGE-0221713. Justin Muncaster would like to thank Toyon Research Corp. for permission to use pieces of code for this research and Bhaskar Rao for work on the keystroke dynamics classifier.

8. References

[1] Altinok and M. Turk, "Temporal integration for continuous multimodal biometrics," In Multimodal User Authentication, '03, Santa Barbara, CA, Dec 11-12, 2003.

[2] Jain, R. Bolle, and S. Pankanti, "Introduction to Biometrics," Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, 1999.

[3] Ross and A. Jain, "Multimodal biometrics: an overview," European Signal Processing Conference, September 2004.

[4] L. Hong and A. Jain, "Multimodal Biometrics", Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, 1999.

[5] K. Jain, K. Nandakumar and A. Ross, "Score Normalization in Multimodal Biometric Systems", Pattern Recognition, Vol. 38, No. 12, pp. 2270-2285, 2005.

[6] S. Zhang, et al., "Continuous verification using multimodal biometrics," ICBA 2006.

[7] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann Publishers, 1988.

[8] S. Lauritzen and D. Spiegehalter, "Local computations with probabilities on graphical structures and their applications to expert systems," Journal of Royal Statistical Society, 1988.

[9] N. Zhang and D. Poole, "A simple approach to Bayesian network computations," Proceedings of the Tenth Biennial Canadian Artificial Intelligence Conference, 1994.

[10] G. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," Artificial Intelligence, 1990.

[11] K. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," Ph.D. Thesis, Dept. of Comp. Sci, UC Berkeley, 2002.

[12] D. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. Thirteenth Conference on Uncertainty in Artificial Intelligence, Providence, RI, pp. 80-89, August 1997.

[13] E. Parzen, "On estimation of a probability density function and mode," Ann. Math. Stat. 33, 1962.

[14] Dr. Libor Spacek Computer Vision Research, http://cswww.essex.ac.uk/mv/allfaces/faces94.html, 1994.

[15] M. Turk and A. Pentland, "Face Recognition using eigenfaces", Proc. IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, 1991.

[16] B. Rao, "Continuous Keystroke Biometric System," M.S. thesis, Media Arts and Technology, UCSB, 2005.
[17] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. Computational Intelligence, (5):142-150, 1989.

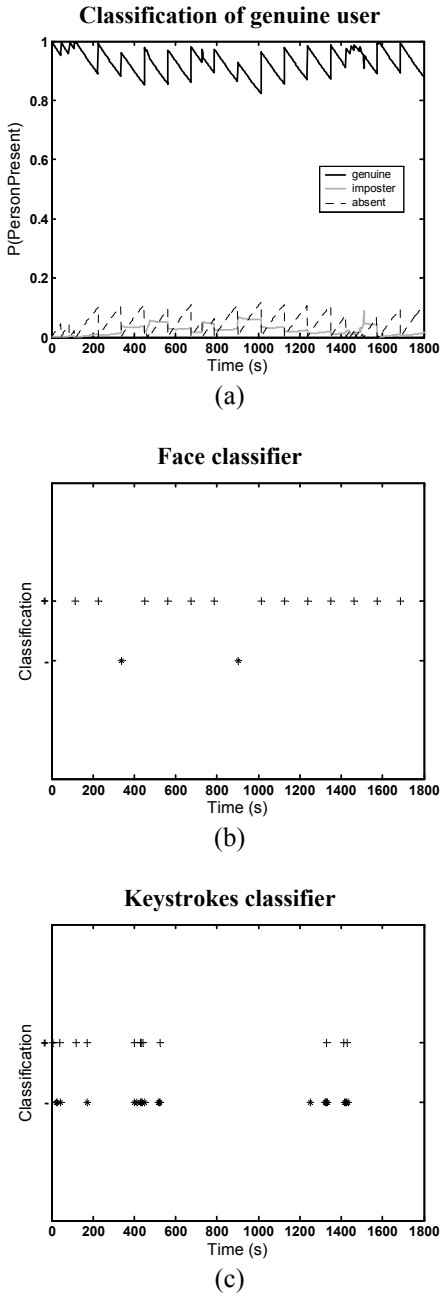


Figure 4. The normal scenario. The user is present for the entire time. (a) State of belief of who the current user is with respect to time. (b) The output of the face classifier. (c) The output of the keystoke classifier.

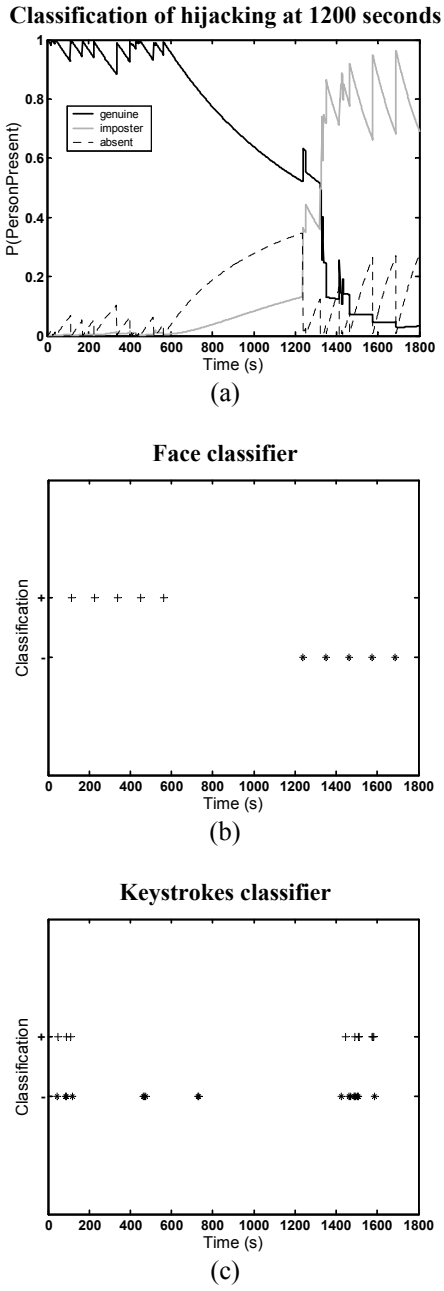


Figure 5. The hijacking scenario. At 600 seconds the user leaves and at 1200 seconds the system is hijacked. (a) State of belief of who the current user is with respect to time. (b) The output of the face classifier. (c) The output of the keystoke classifier.