

AUBE '01

12TH INTERNATIONAL
CONFERENCE ^{ON} AUTOMATIC
FIRE DETECTION

March 25 - 28, 2001
National Institute Of Standards and Technology
Gaithersburg, Maryland U.S.A.

PROCEEDINGS

Editors: Kellie Beall, William Grosshandler and Heinz Luck



NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

Hans-Christian Müller
Duisburg Gerhard-Mercator-University
Department of Communication Engineering
Bismarckstrasse 81, 47048 Duisburg

A New Approach to Fire Detection Algorithms based on the Hidden Markov Model

Abstract

In this paper the theory and some experimental results of a new approach to fire detection algorithms are discussed. The algorithm is based on a signal classification principle, which is widely used in the field of speech recognition. The sensor signals are transformed into appropriate finite time series. The transformation is done in a preprocessing step that extract a suitable sequence from the signals to be classified. The time series are modeled by a Hidden Markov Model (HMM). The classification of a recognized sequence is the final step with respect to the decision making process.

1 Introduction

Advances in semiconductor technology have influenced the development of fire detection algorithms to a high degree in recent years. The use of microprocessors in fire detection devices is nearly state-of-the-art. Improvement of detectivity has been achieved in modern fire detection systems independently of the design scheme with respect to central or distributed intelligence. Implementation of improved detection algorithms leads to reduced false alarm rates and good detection features in comparison with the classical threshold detector.

Modern fire detection algorithms are often based on a fuzzy logic or a neural network approach. They might include some sort of feature extraction methods applied to the sensor signals.

In the following we will introduce the concept of statistical models of sequential data with respect to the fire detection problem.

2 Hidden Markov Models

Let us consider an urn-and-ball system [1], [2]. We assume that there are N (large) glass urns in a room. Within each is a large quantity of colored balls. There are M distinct colors of the balls. The physical process to obtain observations is as follows. A genius is in the room, and, according to some random procedure, it chooses an initial urn. From this urn, a ball is chosen at random, and its color is recorded as the observation. The ball is then replaced in the urn from which it was selected. A new urn is the selected according to the random selection procedure associated with the current urn, and the ball selection process is repeated. The entire process generates a finite observation sequence of colors, which we would like to model as the observable output of an Hidden Markov Model (HMM).

Obviously, the simplest HMM that corresponds to the urn-and-ball process is one in which each state corresponds to a specific urn, and for which a (ball) color probability is defined for each state. The choice of urns is dictated by the state transition-matrix of the HMM.

Furthermore, it should be noted that the ball colors in each urn may be the same, and the distinction among various urns is in the way the collection of colored balls is composed. Therefore an isolated observation of a particular color ball does not immediately tell which urn it is drawn from.

2.1 Elements of an HMM

The above experiment consists of drawing balls from urns in some sequence. Only the sequence of balls is shown to us. An HMM for discrete symbol observation such as the urn-and-ball model is characterized by the following:

1. N , the number of states in the model. Although the states are hidden, for many practical applications there is often physical significance attached to the states of the model. Thus, in the urn-and-ball model, the states correspond to the urns. Generally, the states are interconnected in such a way that any state can be reached from any other state. However, other possible interconnections of states are often of interest. In the following individual states are labeled as $\{1, 2, \dots, N\}$ and the state at time t is denoted as q_t .
2. M , the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled. For the urn-and-ball experiment the observation symbols are the colors of the balls selected from

the urns. We denote the individual symbols as $V = \{v_1, v_2, \dots, v_M\}$.

3. T , the length of an observation sequence.
4. $O = (O_1 O_2 \dots O_T)$ the observation sequence, where O_t denotes the observation at time t .
5. The state transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = P(q_{t+1} = j | q_t = i), \quad 1 \leq i, j \leq N$$

defines the probability of being in state j at time $t + 1$ given that we were in state i at time t . For the special case where any state can be reached from any other state in a single step, we have $a_{ij} > 0$ for all i, j . For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs.

6. The observation symbol probability distribution $B = \{b_j(k)\}$,

$$b_j(k) = P(O_t = v_k | q_t = i), \quad 1 \leq k \leq M$$

defines the probability of observing the symbol $O_t = v_k$ at time t given that we are in state j .

7. The initial state distribution $\pi = \{\pi_i\}$, in which

$$\pi_i = P(q_1 = i), \quad 1 \leq i \leq N$$

defines the probability of being in state i at the beginning of the experiment (i.e., at $t = 1$).

It can be seen from the above that a complete specification of an HMM requires the specification of two model parameters, N and M , specification of observation symbols, and the specification of the three sets of probability measures A , B and π . For convenience, $\lambda = (A, B, \pi)$ will be used as a compact notation to denote an HMM.

Using the model, an observation sequence $O = (O_1 O_2 \dots O_T)$ is generated as follows: We start our experiment at time $t = 1$ by choosing one of the urns, according to the initial probability distribution π , then we choose a ball, the observation symbol from this urn. The state and the observation symbol at time $t = 1$ are denoted as q_1 and O_1 respectively. After this we choose an urn (may be the same or a different from the urn at $t = 1$) according to the transition probability distribution A and again select a ball (denoted as O_2) from this urn depending on the observation symbol probability $b_j(k)$ for that urn (state). The continuation of this procedure up to time $t = T$ generates the observation sequence $O = (O_1 O_2 \dots O_T)$.

2.2 The Three Problems for HMMs

Most applications of HMMs are finally reduced to solving three main problems. These are:

Problem 1: Given the observation sequence $O = (O_1 O_2 \dots O_T)$ and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, for a given model?

Problem 2: Given the observation sequence $O = (O_1 O_2 \dots O_T)$ and a model $\lambda = (A, B, \pi)$, how do we choose a corresponding state sequence $Q = (q_1 q_2 \dots q_T)$ such that $P(O, Q|\lambda)$, the joint probability of the observation sequence and the state sequence is maximized?

Problem 3: How do we adjust the HMM model parameter $\lambda = (A, B, \pi)$ so that $P(O|\lambda)$ or $P(O, Q|\lambda)$ is maximized?

Problem 1 is the evaluation problem, namely, given a model and a observation sequence, how do we compute the probability that the observation was produced by the model? Problem 1 can be viewed as one of scoring how well a given model matches a given observation sequence. For example, if we consider the fire detection problem the solution of problem 1 allows to make an alarm decision.

Problem 2 is the one in which we attempt to uncover the hidden part of the model. Typical uses might be to learn about the structure of the model, to find the optimal state sequence for a given observation sequence.

Problem 3 is the one in which we attempt to optimize the model parameters to best describe how a given observation sequence comes about. The observation sequences to adjust the model parameters are called training sequences because they are used to “train” the model. This problem is the crucial one for most applications of HMMs, because it allows us to create best models for real phenomena.

2.3 Types of the HMMs

One way to classify types of HMMs is by the structure of the transition matrix A . In the special case of ergodic or fully connected HMMs every state can be reached from every other state of the model in a single step.

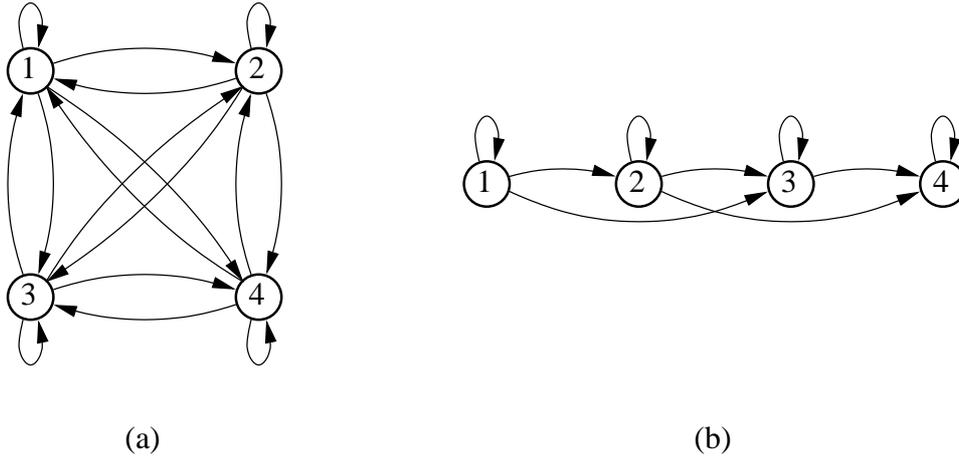


Figure 1: Two types of HMMs

As shown in Figure 1(a), for an $N = 4$ state model, we have

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad \text{with } a_{ij} > 0 \text{ for all } 1 < i, j < 4$$

For some applications, particularly the one discussed here, other types of HMMs have been found to account for observed properties of the signal being modeled better than the standart ergodic model. One such model is shown in Figure 1(b). This model is called a left–right–model because the underlying state sequence associated with the model has the property that the system states proceed from left to right as time increases. Clearly, the left–right–type of HMM has the desirable property that it can model signals whose properties change in time in a successive manner (e.g., increasing sensor signal values obtained form a scattering light smoke sensor). The transition coefficients have the property

$$a_{ij} = 0, \quad j < i$$

Hence, no transitions are allowed to states whose indices are lower than that of the current one. Clearly, the initial state distribution $\pi = \{\pi_i\}$ is given by

$$\pi_i = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases}$$

because the state sequence must begin in state 1. Often, with left–right–models, additional constraints are placed on the transition coefficients. A constraint of the form

$$a_{ij} = 0, \quad j > i + \Delta i$$

is often used to make sure that large changes in state indices do not occur. In particular, for the example of Figure 1(b), the value of Δi is 2.

3 Realization

A discussion of the solution of the three problems of HMMs will exceed the scope of the paper. The solution of problem 3, the synthesis or training problem, is not necessarily part of a detection algorithm. The parameters of an HMM can be viewed as parameters of the detection algorithm which have to be adjusted by a preceding training procedure.

The solution of problem 1, the analysis problem, will be discussed later from the computational effort point of view which is of some interest as far as the practical realization is concerned.

3.1 The parameters of the HMM

We are using a left–right–model with $N = 10$ states. The number of distinct observation symbols per state is $M = 15$. Here, the observation symbols represent certain sensor signal values. The length of an observation sequence equals $T = 12$. We use a sampling rate of 0.2 Hz. Hence, the observation length corresponds to a duration of 1 minute.

The initial state distribution $\pi = \{\pi_i\}$ is given according to the chosen left–right–model by a 10×1 vector with only one nonzero component, i.e. $\pi_1 = 1$. The remaining parameter, the transition probability distribution, represented by a 10×10 matrix, as well as the observation symbol probability distribution, represented by a 10×15 matrix, are adjusted by a couple of training sequences, i.e, we used 30 training sequences taken from test fires of type TF1, TF3, TF4, TF5 and TF7.

3.2 Solution of Problem 1

A straightforward way to determine $P(O|\lambda)$ is by enumerating every possible state sequence of length T . There are N^T of such state sequences.

The probability of the observation sequence $O = (O_1 O_2 \dots O_T)$ given the state sequence

$Q = (q_1 q_2 \dots q_T)$ and the model λ is

$$\begin{aligned} P(O|Q, \lambda) &= \prod_{t=1}^T P(O_t|q_t, \lambda) \\ &= b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T) \end{aligned}$$

The probability of such a state sequence Q can be written as

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

The probability that the O and Q occur simultaneously, is the product of the above two terms

$$P(O, Q|\lambda) = P(O|Q, \lambda)P(Q|\lambda)$$

The probability of O given the model λ is obtained by summing this joint probability over all possible state sequences Q . Hence, we have

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda)P(Q|\lambda)$$

From the last equation we see that the summand involves $2T - 1$ multiplications. Hence summation over all possible state sequences requires $(2T - 1)N^T$ multiplications and $N^T - 1$ additions. Even for small values, $N = 10$ and $T = 12$, this means approximately $2.3 \cdot 10^{13}$ multiplications. Clearly, a more efficient procedure is required to solve problem 1. Such a procedure exists and is called the forward procedure.

Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = i|\lambda)$$

i.e., the probability of the partial observation sequence, O_1, O_2, \dots, O_t , (until time t) and the state i at time t given the model λ . We can solve for $\alpha_t(i)$ inductively, as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad 1 \leq t \leq T-1, 1 \leq j \leq N$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

Again, let us examine the number of multiplications involved with this procedure. The initialization step involves N multiplications. The induction step requires N multiplications plus one for the out of bracket $b_j(O_{t+1})$ term. This has to be done for $1 \leq j \leq N$ and for $1 \leq t \leq T$, which amounts to $(N+1)N(T-1)$ multiplications in the induction step. The termination step requires no further multiplications. Hence the total number of multiplications is $N + N(N+1)(T-1)$. For $N = 10$ and $T = 12$ we need about 1200 computations for the forward procedure as compared to $2.3 \cdot 10^{13}$ required by the direct computation of $P(O|\lambda)$.

3.3 The Algorithm

The algorithm we used works as follows: At each discrete time instant t (e.g., multiple integer of 5 seconds) an observation sequence O is taken from the sensor signal including the current signal value as well as the 11 preceding signal values. Then the a posteriori probability $P(O|\lambda)$ is calculated according to the forward-procedure.

4 Experiments and Results

In this section we present some results taken from a computer simulation of the algorithm described above. The following plots show the output signal $s(t)$ of a scattering light smoke sensor and the a posteriori probability $P(O|\lambda)$ plotted as

$$P^*(O, \lambda) = \min(150, -\log[P(O|\lambda)])$$

The performance of the system in case of a testfire TF1 is in the following plot.

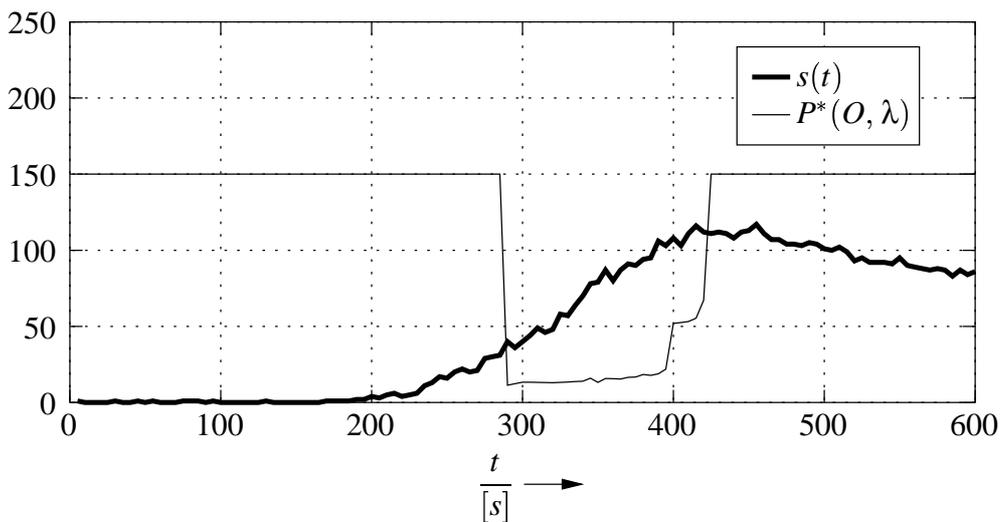


Figure 2: Sensor signal $s(t)$ and the logarithm of the a posteriori probability $P(O|\lambda)$ in case of testfire TF1

Obviously, $P(O|\lambda)$ is maximum (or equivalent: $P^*(O, \lambda)$ is minimum) about 100 seconds after the smoke sensor signal $s(t)$ leaves the steady-state value.

Figure 3 shows the performance of the system in case of a smoldering fire, i.e. testfire TF2. Here, the maximum of $P(O|\lambda)$ is reached 60 seconds after the smoke sensor signal $s(t)$ leaves the steady-state value.

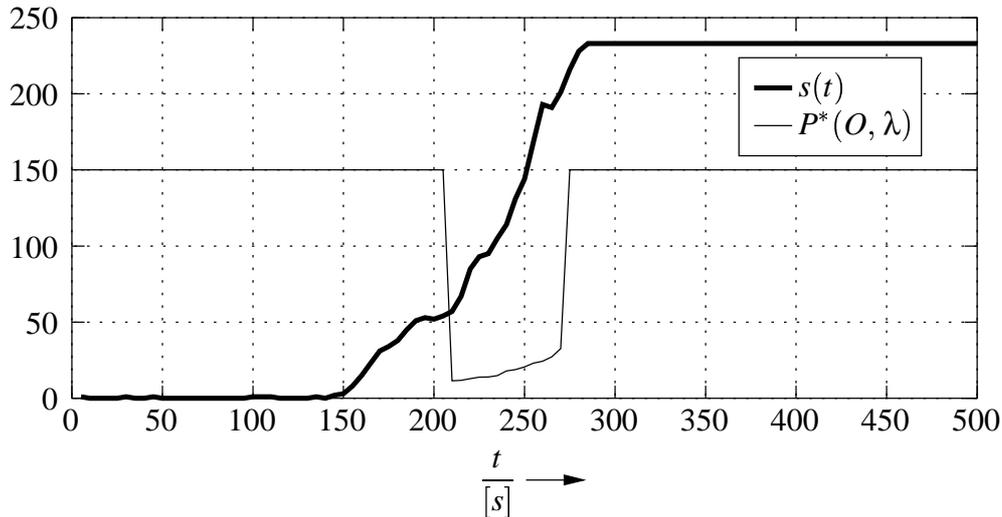


Figure 3: Sensor signal $s(t)$ and the logarithm of the a posteriori probability $P(O|\lambda)$ in case of testfire TF2

Figure 4 shows the performance of the system in case of a slowly developing fire, i.e. testfire TF7.

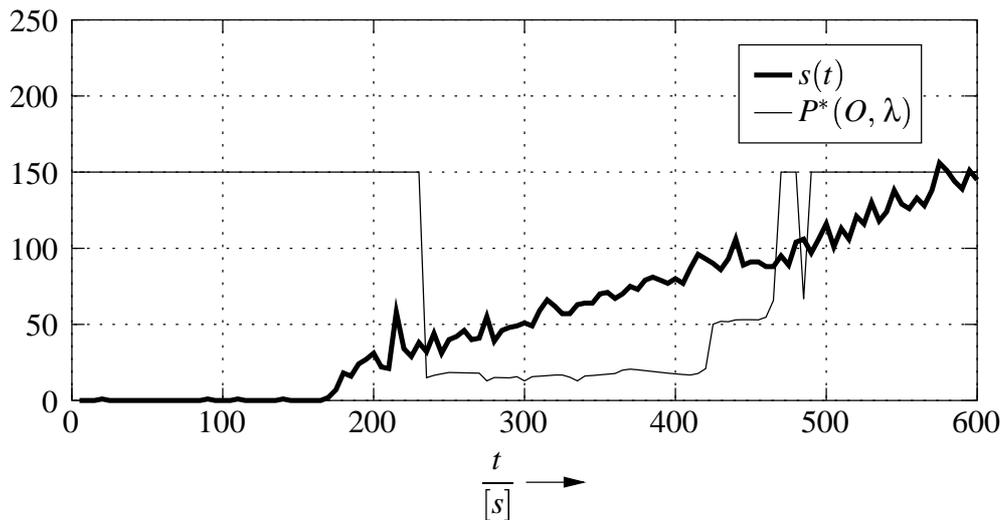


Figure 4: Sensor signal $s(t)$ and the logarithm of the a posteriori probability $P(O|\lambda)$ in case of testfire TF7

Again, the maximum of $P(O|\lambda)$ is reached 60 seconds after the smoke sensor signal $s(t)$ leaves the steady-state value.

Finally, in the last simulation presented here we used an artificial burst signal. Signals like this are most unlikely to occur in real fire situations, but may, for example, occur due to electromagnetic influences.

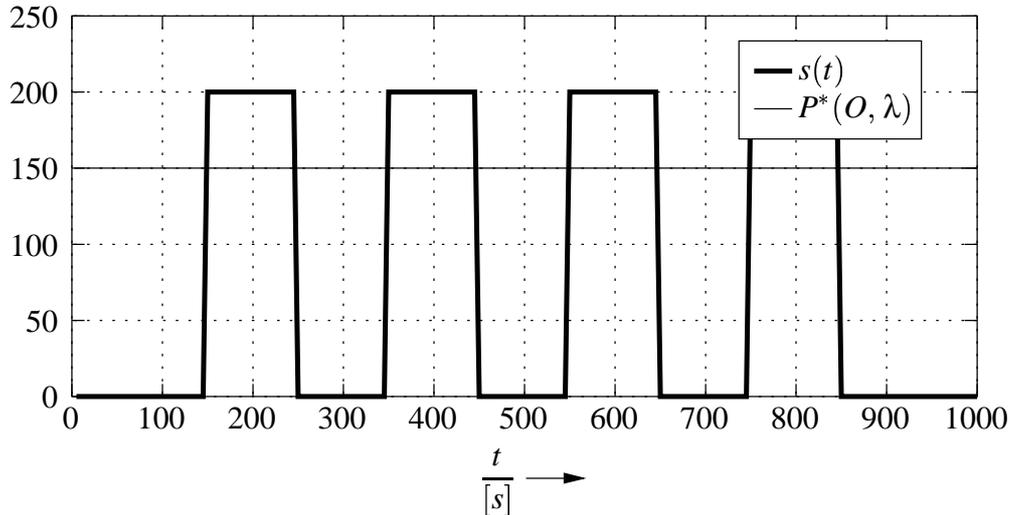


Figure 5: An artificial burst sensor signal $s(t)$ and the logarithm of the a posteriori probability $P(O|\lambda)$

Obviously, the maximum value of $P(O|\lambda)$ (the minimum value of $-\log(P(O|\lambda))$) is much smaller (greater) as compared to the last three examples. Note, a classical threshold detector will reach its alarm condition in case of the shown artificial burst signal.

5 Conclusion

In this paper we have derived a new approach for fire detection algorithms. The key element of our approach, which appears to be quite useful, is a Hidden-Markov-Model. To be more precise, we used a left-right-model. Simulation studies has been presented which show the ability to improve detection features in comparison with the classical threshold detector.

References

- [1] Dugad, Rakesh and Desai, U. B.; *A tutorial on hidden markov models*; Indian Institute of Technology, Bombay Powai; 1996
- [2] Rabiner, Lawrence R; *Fundamentals of speech recognition*; Prentice Hall; 1993