Making use of partial knowledge about hidden states in HMMs : an approach based on belief functions.

Emmanuel Ramasso, Thierry Denoeux

Making use of partial knowledge about hidden states in HMMs: an approach based on belief functions

Emmanuel Ramasso, Thierry Denoeux

***Abstract*—This paper addresses the problem of parameter estimation and state prediction in Hidden Markov Models (HMMs) based on observed outputs and partial knowledge of hidden states expressed in the belief function framework. The usual HMM model is recovered when the belief functions are vacuous. Parameters are learnt using the Evidential Expectation-Maximization algorithm, a recently introduced variant of the Expectation-Maximization algorithm for maximum likelihood estimation based on uncertain data. The inference problem, i.e., finding the most probable sequence of states based on observed outputs and partial knowledge of states, is also addressed. Experimental results demonstrate that partial information about hidden states, when available, may substantially improve the estimation and prediction performances.**

***Index Terms*—Hidden Markov Models, Dempster-Shafer Theory, Evidence Theory, Evidential Expectation-Maximisation (E^2M) algorithm, Uncertain data, Soft labels, Partially supervised learning.**

I. Introduction

Hidden Markov Models (HMMs) are powerful tools for sequential data modeling and analysis. For several decades, many complex applications have been successfully addressed using HMMs, such as word sequence discovery in speech recordings [20], motion sequence recognition in videos [30], gene finding in DNA sequences [16], prognosis of ball bearing degradation [11], [21] or financial time series forecasting [5].

A HMM is a simple dynamic Bayesian network composed of observed random variables (outputs) X_t and latent discrete random variables (hidden states) Y_t, where t is a discrete time index [20] (Figure 1). The sequence of states $Y_1, Y_2, \ldots$ is a Markov chain and the distribution of the output X_t at time t, as well as the distribution of X_t conditional on all X_u, only depend on Y_t. We note that this simple model has recently been extended to "pairwise" [18] and "triplet" Markov chains [19]. However, only the basic HMM will be considered in this paper.

In the standard setting, the outputs are observed until some time T while the states remain hidden. The model parameters (i.e., the probability distribution of Y_1, the state transition probabilities and the parameters of the conditional probability distributions of X_t given Y_t, referred to as emission probabilities) can then be estimated using an iterative procedure called the Baum-Welch algorithm [1], [20], which is a particular instance of the Expectation-Maximization (EM) algorithm.

Emmanuel Ramasso is with the FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM, Automatic Control and Micro-Mechatronic Systems Department, 24 rue Alain Savary, F-25000 Besançon, France

Thierry Denoeux is with the Université de Technologie de Compiègne, Heudiasyc, UMR CNRS 7253, Centre de Recherches de Royallieu, BP 20529, F-60205 Compiègne Cedex, France

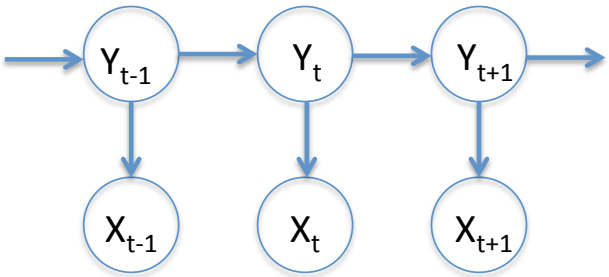


Fig. 1. Graphical representation of a Hidden Markov Model.

In this paper, we consider a different situation in which the states are not completely hidden but are *partially observed*. Partial observations of hidden states may be available in a wide range of applications. For instance, in speech recognition, partial information on words or phonemes may be available from the analysis of lip motion. In behavior analysis, video sequences may be labeled with some imprecision or uncertainty. In machine diagnosis and prognosis applications, experts may express probability judgements on the machine condition at different time steps, etc.

Here, partial knowledge about hidden states will be assumed to be described using the the Dempster-Shafer theory of belief functions [26], a formal framework for representing and reasoning with uncertain information. This theory combines logical and probabilistic approaches to uncertainty and includes the set-membership and probabilistic frameworks as special cases. In particular, it allows the representation of weak knowledge up to complete ignorance: the usual HMM model will thus be recovered as a special case.

In this context, we will solve the two classical problems related to HMMs, i.e.,

1) Estimating the model parameters based on observations of outputs and partial information on states (learning) and
2) Finding the most likely sequence of states, given the observed outputs and partial information on states (inference).

The latter problem will be solved by a variant of the Viterbi algorithm, while the former will be addressed using a methodology for statistical inference based on uncertain observations

first introduced in [7] in the special case of Gaussian mixture models and exposed in a very general setting in [10]. As HMMs can be seen as generalizations of mixture models [3], the results presented in this paper somehow extend those presented in [7], with more mathematical intricacies due to the sequential nature of the model. The main features of this approach are:

1) The representation of uncertain observations using belief functions;
2) The definition of a *generalized likelihood criterion* that can be interpreted in terms of degree of conflict between the statistical model and the observations.
3) An extension of the EM algorithm, called the *Evidential EM (E^{2}M) algorithm*, which under very general conditions converges to a local maximum of this criterion.

The rest of the paper is organized as follows. Section II presents the necessary background on belief functions and the E^{2}M algorithm. The core of our contribution is described in Section III and Section IV reports experimental results. Section V concludes the paper.

II. Background on belief functions

This section recalls the necessary background notions on the Dempster-Shafer theory of belief functions (Subsection II-A) and its application to statistical estimation using the E^{2}M algorithm (Subsection II-B).

A. Basic concepts

Let Y be a variable taking values in a finite domain Ω, called the *frame of discernment*. Uncertain information about Y may be represented by a *mass function* m on Ω, defined as a function from the powerset of Ω, denoted by 2^Ω, to the interval $[0,1]$, such that

$$\sum_{A\subseteq\Omega} m(A) = 1. \tag{1}$$

Function m is said to be *normalized* if $m(\emptyset) = 0$, a condition that will be assumed in the rest of this paper. Any subset A of Ω such that $m(A) > 0$ is called a *focal element* of m. Two special cases are of interest:

1) If m has a single focal element A, it is said to be *logical* and denoted as m_A. Such a mass function encodes a piece of evidence that tells us that $Y \in A$, and nothing else. There is a one-to-one correspondence between subsets A of Ω and logical mass functions m_A: logical mass functions are thus equivalent to sets.
2) If all focal elements of m are singletons, then m is said to be *Bayesian*. There is a one-to-one correspondence between probability distributions $p : \Omega \to [0,1]$ and Bayesian mass functions m such that $m(\{\omega\}) = p(\omega)$, for all $\omega \in \Omega$: Bayesian mass functions are thus equivalent to probability distributions.

To each normalized mass function m, we may associate belief and plausibility functions from 2^Ω to $[0,1]$ defined as follows:

$$Bel(A) = \sum_{B\subseteq A} m(B) \tag{2a}$$

$$Pl(A) = \sum_{B\cap A\neq\emptyset} m(B), \tag{2b}$$

for all $A \subseteq \Omega$. These two functions are linked by the relation $Pl(A) = 1 - Bel(\overline{A})$, for all $A \subseteq \Omega$. Each quantity $Bel(A)$ may be interpreted as the degree to which the evidence *supports* A, while $Pl(A)$ can be interpreted as the degree to which the evidence *does not refute* A. The following inequalities always hold: $Bel(A) \leq Pl(A)$, for all $A \subseteq \Omega$. If m is Bayesian, then function Bel is equal to Pl and is a probability measure. The function $pl : \Omega \to [0,1]$ such that $pl(\omega) = Pl(\{\omega\})$ is called the *contour function* associated to m.

Let m_1 and m_2 be two mass functions induced by independent items of evidence. Their *degree of conflict* [26] is defined by

$$\kappa = \sum_{B\cap C=\emptyset} m_1(B)m_2(C). \tag{3}$$

If $\kappa < 1$, m_1 and m_2 are not totally conflicting and they can be combined using Dempster's rule [26] to form a new mass function defined as:

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B\cap C=A} m_1(B)m_2(C) \tag{4}$$

for all $A \subseteq \Omega$, $A \neq \emptyset$ and $(m_1 \oplus m_2)(\emptyset) = 0$. Dempster's rule is commutative, associative, and it admits as neutral element the *vacuous* mass function defined as $m(\Omega) = 1$.

Let us now assume that m_1 is Bayesian. Its contour function is a probability distribution p_1 defined by $p_1(\omega) = m_1(\{\omega\})$ for all $\omega \in \Omega$. Combining m_1 with an arbitrary mass function m_2 with contour function pl_2 yields a *Bayesian mass function* $m_1 \oplus m_2$ with contour function $p_1 \oplus pl_2$ defined by

$$(p_1 \oplus pl_2)(\omega) = \frac{p_1(\omega)pl_2(\omega)}{\sum_{\omega'\in\Omega} p_1(\omega')pl_2(\omega')}. \tag{5}$$

(We note that, without ambiguity, the same symbol $\oplus$ is used for mass functions and contour functions). The degree of conflict between p_1 and pl_2 is

$$\kappa = 1 - \sum_{\omega'\in\Omega} p_1(\omega')pl_2(\omega'). \tag{6}$$

It is equal to one minus the mathematical expectation of pl_2 with respect to p_1. Finally, we may also note that, if m_2 is logical and such that $m_2(A) = 1$, then $p_1 \oplus pl_2$ is the probability distribution obtained by conditioning p_1 with respect to A.

B. E^{2}M algorithm

Let $\mathbf{Z}$ be a discrete random vector taking values in $\Omega_{\mathbf{Z}}$, with probability mass function $p_{\mathbf{Z}}(\cdot;\boldsymbol{\theta})$ depending on an unknown parameter $\boldsymbol{\theta} \in \Theta$. Let $\mathbf{z}$ denote a realization of $\mathbf{Z}$, referred to as the *complete data*. If $\mathbf{z}$ was perfectly observed, then the likelihood function given $\mathbf{z}$ would be defined as the function

from Θ to $[0,1]$ such that:

$$L(\boldsymbol{\theta};\mathbf{z}) = p_{\mathbf{Z}}(\mathbf{z};\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in \Theta. \tag{7}$$

Let us now assume that $\mathbf{z}$ is not precisely observed, but it is known for sure that $\mathbf{z} \in A$ for some $A \subseteq \Omega_{\mathbf{Z}}$. The likelihood function given such *imprecise data* is now:

$$L(\boldsymbol{\theta};A) = p_{\mathbf{Z}}(A;\boldsymbol{\theta}) = \sum_{\mathbf{z}\in A} p_{\mathbf{Z}}(\mathbf{z};\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in \Theta. \tag{8}$$

More generally, our knowledge of $\mathbf{z}$ may be not only imprecise, but also *uncertain*; it can then be described by a mass function m on $\Omega_{\mathbf{Z}}$ with focal elements $A_1,\ldots,A_r$ and corresponding masses $m(A_1),\ldots,m(A_r)$. In [10] it was proposed to extend the likelihood function (8) given such uncertain data by computing the weighted sum of the terms $L(\boldsymbol{\theta};A_i)$ with coefficients $m(A_i)$, which leads to the following expression:

$$L(\boldsymbol{\theta};m) = \sum_{i=1}^{r} m(A_i) L(\boldsymbol{\theta};A_i). \tag{9}$$

Using (8) and exchanging the order of summations over i and $\mathbf{z}$, we get

$$\begin{aligned} L(\boldsymbol{\theta};m) &= \sum_{\mathbf{z}\in\Omega_{\mathbf{Z}}} p_{\mathbf{Z}}(\mathbf{z};\boldsymbol{\theta}) \sum_{A_i \ni \mathbf{z}} m(A_i) && \text{(10a)} \\ &= \sum_{\mathbf{z}\in\Omega_{\mathbf{Z}}} p_{\mathbf{Z}}(\mathbf{z};\boldsymbol{\theta}) pl(\mathbf{z}). && \text{(10b)} \end{aligned}$$

The likelihood $L(\boldsymbol{\theta};m)$ thus only depends on m through its associated contour function pl. For this reason, we may write indifferently $L(\boldsymbol{\theta};m)$ or $L(\boldsymbol{\theta};pl)$. By comparing (10) with (6), we can see that $L(\boldsymbol{\theta};m)$ equals *one minus the degree of conflict* between $p_{\mathbf{Z}}(\cdot;\boldsymbol{\theta})$ and m. Consequently, maximizing $L(\boldsymbol{\theta};m)$ with respect to $\boldsymbol{\theta}$ amounts to *minimizing the conflict* between the parametric model and the uncertain observations. We may also observe from (10) that $L(\boldsymbol{\theta};pl)$ can be alternatively defined as the mathematical expectation of $pl(\mathbf{Z})$, given $\boldsymbol{\theta}$:

$$L(\boldsymbol{\theta};pl) = \mathbb{E}_{\boldsymbol{\theta}}\left[pl(\mathbf{Z})\right]. \tag{11}$$

To maximize the likelihood function $L(\boldsymbol{\theta};pl)$ given uncertain data pl, it was proposed in [9], [10] to adapt the EM algorithm [8] as follows.

In the E-step, the conditional expectation of $\log L(\boldsymbol{\theta};\mathbf{Z})$ considered in the standard EM algorithm is now replaced by the expectation with respect to $p_{\mathbf{Z}}(\cdot;\boldsymbol{\theta}^{(q)}) \oplus pl$, denoted as $p_{\mathbf{Z}}(\cdot|pl;\boldsymbol{\theta}^{(q)})$, where $\boldsymbol{\theta}^{(q)}$ is the current fit of parameter $\boldsymbol{\theta}$ at iteration q. We may remark that conditional expectation is recovered in the special case where m is a logical mass function. Using (5), the probability mass function $p_{\mathbf{Z}}(\cdot|pl;\boldsymbol{\theta}^{(q)})$ has the following expression:

$$p_{\mathbf{Z}}(\mathbf{z}|pl;\boldsymbol{\theta}^{(q)}) = \frac{p_{\mathbf{Z}}(\mathbf{z};\boldsymbol{\theta}^{(q)}) pl(\mathbf{z})}{L(\boldsymbol{\theta}^{(q)};pl)}, \tag{12}$$

where $L(\boldsymbol{\theta}^{(q)};pl)$ is given by (10). At iteration q, the following function is thus computed:

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}) &= \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log(L(\boldsymbol{\theta};\mathbf{Z}))|pl\right] && \text{(13a)} \\ &= \frac{\displaystyle\sum_{\mathbf{z}\in\Omega_{\mathbf{Z}}} \log(L(\boldsymbol{\theta};\mathbf{z})) p_{\mathbf{X}}(\mathbf{z};\boldsymbol{\theta}^{(q)}) pl(\mathbf{z})}{L(\boldsymbol{\theta}^{(q)};pl)} && \text{(13b)} \end{aligned}$$

The M-step is unchanged and requires the maximization of $Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)})$ with respect to $\boldsymbol{\theta}$. The E^{2}M algorithm alternately repeats the E- and M-steps above until the increase of observed-data likelihood becomes smaller than some threshold.

As shown in [10], the E^{2}M algorithm inherits the monotonicity property of the EM algorithm, which under broad conditions ensures convergence to a local maximum of $L(\boldsymbol{\theta};pl)$. This algorithm has been applied to mixture models with partial information on class labels [7] and/or uncertain attributes [10] and to partially supervised Independent Factor Analysis [6].

III. Partially Hidden Markov Models

In this section, we consider the HMM model introduced in Section I and we assume that partial knowledge of hidden states Y_t is available in the form of mass functions m_t for each $t \in \{1,\ldots,T\}$. The resulting model can be called a *Partially Hidden Markov Model* (PHMM). The notations will first be introduced in Subsection III-A. The learning and inference problems will then be tackled in Subsections III-B and III-C, respectively. Finally, a comparison between the model introduced in this section and related work will be performed in Subsection III-D.

A. Model and notations

A HMM can be described by the following parameters:

- Prior probabilities $\boldsymbol{\Pi} = \{\pi_1,\ldots,\pi_k,\ldots,\pi_K\}$, where $\pi_k = P(Y_1 = k)$ is the probability that the system was in state k at $t=1$ and K is the number of states;
- Transition probabilities $\boldsymbol{A} = [a_{k\ell}]$, where
$$a_{k\ell} = P(Y_t = \ell | Y_{t-1} = k), \quad (k,\ell) \in \{1,\ldots,K\}^2 \tag{14}$$
is the probability for the system to be in state ℓ at time t given that it was in state k at $t-1$, with $\sum_\ell a_{k\ell} = 1$;
- Parameters $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_j,\ldots,\boldsymbol{\phi}_K\}$ of the emission probability distributions in each state:
$$p_k(x_t;\boldsymbol{\phi}_k) = p(x_t|Y_t = k;\boldsymbol{\phi}_k), \quad k \in \{1,\ldots,K\}. \tag{15}$$

All these parameters can be arranged in a vector $\boldsymbol{\theta} = \{\mathbf{A},\boldsymbol{\Pi},\boldsymbol{\Phi}\}$.

Let $\mathbf{x} = (x_1,\ldots,x_T)$ denote the observed output sequence and $\mathbf{y} = (y_1,\ldots,y_T)$ the corresponding sequence of hidden states. To express the different probability distributions as functions of the parameters, let Y_{tk} denote the binary variable that equals 1 if the system was in state k at time t and 0 otherwise. With this notation, we have

$$p(y_1;\boldsymbol{\Pi}) = \prod_{k=1}^{K} \pi_k^{y_{tk}}, \tag{16a}$$

$$p(y_t|y_{t-1};\mathbf{A}) = \prod_{k=1}^{K}\prod_{\ell=1}^{K} a_{k\ell}^{y_{(t-1,k)}y_{t\ell}} \tag{16b}$$

and

$$p(x_t|y_t;\mathbf{\Phi}) = \prod_{k=1}^{K} p_k(x_t;\boldsymbol{\phi}_k)^{y_{tk}}. \tag{16c}$$

The likelihood function given the complete data $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ is thus

$$L(\boldsymbol{\theta};\mathbf{z}) = p(\mathbf{z};\boldsymbol{\theta}) = p(y_1;\mathbf{\Pi})\left(\prod_{t=2}^{T} p(y_t|y_{t-1};\mathbf{A})\right)\prod_{t=1}^{T} p(x_t|y_t;\mathbf{\Phi}) = \left(\prod_{k=1}^{K}\pi_k^{y_{1k}}\right)\left(\prod_{t=2}^{T}\prod_{k,\ell} a_{k\ell}^{y_{(t-1,k)}y_{t\ell}}\right)\left(\prod_{t=1}^{T}\prod_{k=1}^{K} p_k(x_t;\boldsymbol{\phi}_k)^{y_{tk}}\right). \tag{17}$$

In this paper, we assume that partial knowledge about the state y_t at each time t is available in the form of a mass function m_t on Ω. The observations thus consist in the output sequence $x_1, \ldots, x_T$ as in the usual HMM model and a sequence of mass functions $m_1, \ldots, m_T$ with corresponding contour functions $pl_1, \ldots, pl_T$, referred to as *uncertain (soft)* labels [7]. Combining these T mass functions using Dempster's rule yields a mass function on the product space Ω^T with contour function

$$pl(\mathbf{y}) = \prod_{t=1}^{T} pl(y_t). \tag{18}$$

Since $\mathbf{x}$ is precisely observed, we have $pl(\mathbf{x}', \mathbf{y}) = pl(\mathbf{y})$ if $\mathbf{x}' = \mathbf{x}$ and $pl(\mathbf{x}', \mathbf{y}) = 0$ otherwise, for all $(\mathbf{x}', \mathbf{y})$. The generalized likelihood function (10) then has the following expression:

$$L(\boldsymbol{\theta};\mathbf{x},pl) = \sum_{\mathbf{y}} L(\boldsymbol{\theta};\mathbf{x},\mathbf{y})pl(\mathbf{y}). \tag{19}$$

As suggested in [9], there is a formal analogy between the above model and the following probabilistic model. Consider a HMM whose output at each time t is a pair (X_t, U_t), where U_t is a Bernoulli random variable such that $P(U_t = 1|Y_t = k) = pl_{tk}$ and

$$p(x_t, U_t = 1|Y_t = k) = p(x_t|Y_t = k)P(U_t = 1|Y_t = k) = p_k(x_t)pl_{tk}, \tag{20}$$

for each $k \in \{1, \ldots, K\}$. Let $\mathbf{U} = (U_1, \ldots, U_T)$ and $\mathbf{u} = (1, \ldots, 1)$. The conditional probability of observing $\mathbf{U} = \mathbf{u}$ given that the system is in state k is

$$P(\mathbf{U} = \mathbf{u}|Y_t = k) = \prod_{t=1}^{T} P(U_t = 1|Y_t = k) = \prod_{t=1}^{T} pl_{tk}, \tag{21}$$

for each $k \in \{1, \ldots, K\}$. The likelihood function after observing $\mathbf{X} = \mathbf{x}$ and $\mathbf{U} = \mathbf{u}$ is

$$\begin{aligned} L(\boldsymbol{\theta};\mathbf{x},\mathbf{u}) &= p(\mathbf{x},\mathbf{u};\boldsymbol{\theta}) && \text{(22a)}\\ &= \sum_{\mathbf{y}} p(\mathbf{x},\mathbf{u}|\mathbf{y})p(\mathbf{y}) && \text{(22b)}\\ &= \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{u}|\mathbf{y})p(\mathbf{y}) && \text{(22c)}\\ &= \sum_{\mathbf{y}} p(\mathbf{x},\mathbf{y})pl(\mathbf{y}), && \text{(22d)} \end{aligned}$$

which is equal to $L(\boldsymbol{\theta};\mathbf{x},pl)$ from (19). This result shows that this artificial probabilistic model (with fictitious variables U_t taking value 1) is *formally* equivalent to the one considered here. This purely formal analogy will be instrumental in proving the results presented in the two following subsections.

B. Learning

The problem considered in this section is to estimate (learn) parameter $\boldsymbol{\theta}$, given the output sequence $\mathbf{x}$ and fixed uncertain labels $pl_1, \ldots, pl_T$, by maximizing the generalized likelihood function (19).

In order to implement the E-step if the E^{2}M recalled in Subsection II-B, we need to compute the expectation of the complete data log-likelihood with respect to the probability distribution $p(\mathbf{z}|\mathbf{x},pl;\boldsymbol{\theta}^{(q)})$ obtained by combining $p(\mathbf{z};\boldsymbol{\theta}^{(q)})$ with $pl(\mathbf{z})$ using Dempster's rule or, equivalently, by combining $p(\mathbf{z};\boldsymbol{\theta}^{(q)})$ with $pl(\mathbf{y})$ and conditioning on $\mathbf{x}$. By taking the logarithm of (17), we get

$$\log L(\boldsymbol{\theta};\mathbf{z}) = \sum_{k=1}^{K} y_{1k}\log\pi_j + \sum_{t=2}^{T}\sum_{k,\ell} y_{t-1,k}y_{t\ell}\log a_{ij} + \sum_{t=1}^{T}\sum_{k=1}^{K} y_{tk}\log p_k(x_t;\boldsymbol{\phi}_k). \tag{23}$$

Hence,

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}) = \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[L(\boldsymbol{\theta};\mathbf{Z})|\mathbf{x},pl\right] = \sum_{k=1}^{K}\gamma_{1k}^{(q)}\log\pi_j + \sum_{t=2}^{T}\sum_{k,\ell}\xi_{t-1,t,k,\ell}^{(q)}\log a_{ij} + \sum_{t=1}^{T}\sum_{k=1}^{K}\gamma_{tk}^{(q)}\log p_k(x_t;\boldsymbol{\phi}_k), \tag{24}$$

with $\gamma_{tk}^{(q)} = \mathbb{E}_{\boldsymbol{\theta}^{(q)}}[Y_{t,k}|\mathbf{x},pl]$ and $\xi_{t-1,t,k,\ell}^{(q)} = \mathbb{E}_{\boldsymbol{\theta}^{(q)}}(Y_{t-1,k}Y_{t\ell}|\mathbf{x},pl)$.

To compute $\gamma_{tk}^{(q)}$ and $\xi_{t-1,t,k,\ell}^{(q)}$, we can follow the same line of reasoning as for standard HMMs [20][3, Chapter 13]. The following proposition is proved in Appendix:

Proposition 1: We have

$$\gamma_{tk}^{(q)} = \frac{\alpha_{tk}^{(q)}\beta_{tk}^{(q)}}{L(\boldsymbol{\theta}^{(q)};\mathbf{x},pl)}, \tag{25}$$

$$\xi_{t-1,t,k,\ell}^{(q)} = \frac{\alpha_{t-1,k}^{(q)}\ p_\ell(x_t;\boldsymbol{\phi}_\ell^{(q)})pl_{t\ell}\ a_{k\ell}^{(q)}\ \beta_{t\ell}^{(q)}}{L(\boldsymbol{\theta}^{(q)};\mathbf{x},pl)} \tag{26}$$

and

$$L(\boldsymbol{\theta}; \mathbf{x}, pl) = \sum_{k=1}^{K} \alpha_{Tk}. \tag{27}$$

where the variables $\alpha_{tk}^{(q)}$ and $\beta_{tk}^{(q)}$ can be computed recursively as follows:

$$\alpha_{1k}^{(q)} = \pi_k^{(q)}\ pl_{1k}\ p_k(x_1; \boldsymbol{\phi}^{(q)}), \tag{28a}$$

$$\alpha_{t,k}^{(q)} = p_k(x_t; \boldsymbol{\phi}^{(q)}) pl_{tk} \sum_{\ell} \alpha_{t-1,\ell}^{(q)}\ a_{\ell k}^{(q)}, \tag{28b}$$

for $t = 2, \ldots, T$ and

$$\beta_{Tk}^{(q)} = 1, \tag{29a}$$

$$\beta_{t,k}^{(q)} = \sum_{\ell} \beta_{t+1,\ell}^{(q)}\ p_\ell(\mathbf{x}_{t+1}; \boldsymbol{\phi}^{(q)})\ pl_{t+1,\ell}\ a_{k\ell}^{(q)} \tag{29b}$$

for $t = T-1, \ldots, 1$. □

The M-step of the E^{2}M algorithm is similar to that of the EM algorithm in the standard case. Maximization of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ with respect to $\boldsymbol{\Pi}$ and $\mathbf{A}$ is achieved using appropriate Lagrange multipliers, which leads to:

$$\pi_k^{(q+1)} = \gamma_{1k}^{(q)} \tag{30a}$$

$$a_{k\ell}^{(q+1)} = \frac{\sum_{t=2}^{T} \xi_{t-1,t,k,\ell}^{(q)}}{\sum_{t=2}^{T} \sum_{\ell'=1}^{K} \xi_{t-1,t,k,\ell'}^{(q)}}. \tag{30b}$$

Update equations resulting from the maximization of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ with respect to $\boldsymbol{\Phi}$ depend on the form of the emission probability distributions. For instance, in the case of Gaussian emission densities, we have $p_k(x_t; \boldsymbol{\phi}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$ and the update equations are [20]:

$$\boldsymbol{\mu}_k^{(q+1)} = \frac{\sum_{t=1}^{T} \gamma_{tk}^{(q)}\ x_t}{\sum_{t=1}^{T} \gamma_{tk}^{(q)}}, \tag{31a}$$

$$\boldsymbol{\Sigma}_k^{(q+1)} = \frac{\sum_{t=1}^{T} \gamma_{tk}^{(q)} (x_t - \boldsymbol{\mu}_k^{(q+1)})(x_t - \boldsymbol{\mu}_k^{(q+1)})'}{\sum_{t=1}^{T} \gamma_{tk}^{(q)}}. \tag{31b}$$

We can remark that the consideration of partial knowledge on hidden states does not result in any increase in the complexity of the learning algorithm. Equations (28a)-(29b) correspond to a variant of the so-called forward-backward algorithm [20][3, Chapter 13], whose computational complexity scales like $O(K^2T)$, and updating the parameters through Equations (30)-(31) can be performed in $O(KT)$ operations, so that the overall complexity of one iteration of the E^{2}M algorithm is $O(K^2T)$. However, the number of iterations needed by the E^{2}M algorithm to achieve convergence can be expected to be influenced by the supplied knowledge on hidden states, faster convergence being achieved when more informative and accurate labels are provided. This phenomenon will be demonstrated experimentally in Subsection IV-B.

Several issues need to be addressed to make the algorithm work in practice. As in the usual forward-backward algorithm, the terms $\alpha_{t,k}^{(q)}$ and $\beta_{t,k}^{(q)}$ have to be rescaled to prevent them from converging exponentially to zero. The means and covariances of the Gaussian distributions can be initialized using a clustering procedure such as the K-means algorithm. Alternatively, we may pick K points randomly in $\{x_1, \ldots, x_T\}$ to initialize the means and use the whole dataset to initialize the covariances. Prior and transition probabilities can be estimated using uncertain labels using a process similar to that described in [25], [22]:

$$\pi_k^{(0)} \propto pl_1(k) \tag{32a}$$

$$a_{k\ell}^{(0)} \propto \sum_{t=2}^{T} pl_{t-1}(k) \cdot pl_t(\ell). \tag{32b}$$

If several training sequences are available, the results are simply averaged as done with usual HMMs [20].

C. Inference

The inference process as considered here consists in finding the most likely state sequence $(y_1^*, \ldots, y_T^*)$ given observed outputs $(x_1, \ldots, x_T)$ and partial knowledge about states, encoded as contour functions $pl_1, \ldots, pl_T$. This problem is important problem in many applications in which the states have a well-defined meaning such as speech [20], image [19], video [30] or signal [21] segmentation.

In the standard HMM model, the Viterbi algorithm makes it possible to retrieve the most probable sequence of hidden states given observations in TK^2 operations instead of K^T for greedy search [28], [12]. Thanks to the formal analogy with a probabilistic model as explained in Section III-A, the Viterbi algorithm can be directly applied in the case where partial knowledge about hidden state is available.

Let $\delta_t(k; \boldsymbol{\theta})$ denote the highest probability of a sequence $(\mathbf{x}_{1:t}, \mathbf{u}_{1:t}, \mathbf{y}_{1:t})$ up to time t and ending in state k:

$$\delta_t(k; \boldsymbol{\theta}) = \max_{\mathbf{y}_{1:t-1}} p(\mathbf{x}_{1:t}, \mathbf{u}_{1:t}, \mathbf{y}_{1:t-1}, y_t = k; \boldsymbol{\theta}). \tag{33}$$

These probabilities can be iteratively computed by:

$$\begin{aligned} \delta_t(k; \boldsymbol{\theta}) = \max_{\ell} [\delta_{t-1}(\ell; \boldsymbol{\theta}) P(Y_t = k | Y_{t-1} = \ell)] \\ p(x_t, U_t = 1 | Y_t = k) = \\ \max_{\ell} [\delta_{t-1}(\ell; \boldsymbol{\theta})\ a_{\ell k}]\, p_k(x_t; \phi_k) pl_{tk}, \end{aligned} \tag{34}$$

for $t = 2, \ldots, T$, starting from $\delta_1(k; \boldsymbol{\theta}) = \pi_k p_k(x_1; \phi_k) pl_{1k}$. The highest probability for the complete sequence is then

$$P^* = \max_k \delta_T(k; \boldsymbol{\theta}). \tag{35}$$

By keeping track of the argument maximizing the expression in (34):

$$\psi_t(k) = \arg\max_{\ell} [\delta_{t-1}(\ell; \boldsymbol{\theta}) a_{\ell k}] \tag{36}$$

for each t and k, the best state sequence can be retrieved by

backtracking as follows:

$$y^*_{t-1} = \psi_t(y^*_t), \quad t = T, \ldots, 2. \tag{37}$$

Note that similar equations were obtained in [25] for a different model called Evidential HMM, using a different process based on conditioning.

D. Related work

Before presenting numerical experiments with the model introduced above, it is interesting to compare it with some previous extensions of HMMs in the belief function framework.

In [13] and [2], the authors extend, respectively, hidden Markov chains and hidden Markov fields by allowing the output vectors X_t (corresponding to sensor measurements) to have a conditional probability distribution $p(x_t|Y_t \in A)$, for each $A \subseteq \Omega$. This extension provides a way to model partial sensor information. For instance, in a remote sensing application, Ω might have two elements: "forest" and "water", and X_t might represent the information from an optical sensor. The conditional density $p(x_t|Y_t \in \Omega)$ might then model the distribution of the sensor data in spots hidden by clouds. The authors then build a mass function $m_{\mathbf{x}}$ on the product space Ω^T induced by the sensor measurements $\mathbf{x} = (x_1, \ldots, x_T)$ and combine it with the Markov probability distribution $p(\mathbf{y})$ of $\mathbf{Y}$ using Dempster's rule. They show that the result is a Markov probability distribution, which allows them to use classical segmentation methods.

In [15], the authors propose to model a nonstationary Markov chain $(Y_1, \ldots, Y_T)$, with ill-known distribution, by an evidential Markov chain, defined as a mass function m_0 on Ω^T such that $m(A) = 0$ if $A \notin (2^\Omega)^T$ and

$$m_0(A_1 \times \ldots \times A_T) = m_0(A_1)m_0(A_2|A_1) \ldots m_0(A_T|A_{T-1}) \tag{38}$$

for all $(A_1, \ldots, A_T) \in (2^\Omega)^T$ (see also [27]). They show that the combined mass function $m_0 \oplus m_{\mathbf{x}}$, where $m_{\mathbf{x}}$ is the Bayesian mass function induced by sensor measurements, is the conditional distribution $p(\mathbf{y}|\mathbf{x})$ defined by $p(\mathbf{x}, \mathbf{y})$, where $p(\mathbf{x}, \mathbf{y})$ is the marginal distribution of a triplet Markov chain [17]. Hence, $p(\mathbf{y}|\mathbf{x})$ is computable in time linear in the number of observations. Furthermore, the authors propose a variant of the EM algorithm for estimating the parameters of the (stationary) evidential Markov chain and of the emission probability distributions.

The two above models are combined in [4], where the authors propose to model jointly the nonstationarity of the state sequence by an evidential Markov chain, and the imprecision of sensor measurements by conditional probability distributions $p(x_t|Y_t \in A)$ for each $A \subseteq \Omega$. Once again, they show that hidden states can be restored in linear time with respect to T, and they provide an algorithm for estimating the model parameters. In [19], the author considers even more general models consisting of pairwise Markov chains in which the hidden state sequence is modeled by an evidential Markov chain and sensors provide evidential information.

By comparing this previous work with the contribution presented in this paper, it is clear that they pursue different goals: in [2], [15], [19], [4], the authors extend the HMM to model situations in which we have *less* information that would be required to use the standard HMM (due to partial sensor information and/or nonstationarity of the hidden state sequence). In contrast, in our approach, we consider a standard HMM model (seen as a data generation mechanism), which is supplemented by belief functions that encode partial knowledge of hidden states, collected after the data have been generated. We thus handle situations in which we have *more* information than is usually assumed when using HMMs.

From a mathematical point of view, and adopting the terminology of Ref. [19], the inference algorithm presented in Subsection III-C can be seen as the Dempster's combination of the Markov distribution $p(\mathbf{y})$, a non Markovian mass function defined by (18) and a Bayesian mass function induced by the emission probability distribution $p(\mathbf{x}|\mathbf{y})$. It would be interesting to study more precisely the formal relationship between this model and the very general models introduced in [19]. The use of partial information, such as considered in this paper, in extensions of HMMs such as pairwise or triplet Markov chains, or even in the more general models introduced in [19], is also an interesting perspective. These research topics go beyond the scope of this paper and are left for further research.

IV. Experiments

In this section, the benefits of using partial knowledge on hidden states using the approach describe above are first demonstrated with simulated data in Subsection IV-A. Experimental results with engine condition data are then reported in Subsection IV-B.

A. Simulated data

We consider in this subsection data generated using a HMM with three states and three-dimensional Gaussian emission probability distributions $p_k(x_t; \boldsymbol{\phi}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$. The parameters were fixed as follows:

$$\boldsymbol{\Pi} = (1/3, 1/3, 1, 3)', \quad \mathbf{A} = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{pmatrix},$$

$$\boldsymbol{\mu}_1 = (2, 0, 0)', \quad \boldsymbol{\mu}_2 = (0, 2, 0)', \quad \boldsymbol{\mu}_3 = (0, 0, 2)',$$

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = I.$$

Three different experiments were carried out with this model to study the influence of soft label imprecision, labeling error, and partial information on states when segmenting a new output sequence.

1) Influence of label imprecision: To study how the imprecision of knowledge on hidden states influences the performances of the learning procedure described in Subection III-B, we proceeded as follows. A learning sequence $(\mathbf{x}, \mathbf{y})$ of length T was generated using the above model. Uncertain labels were then generated as follows:

$$pl_{tk} = \begin{cases} 1 & \text{if } y_t = k, \\ \nu & \text{otherwise,} \end{cases} \tag{39}$$

where ν is a nonspecificity coefficient, which quantifies the imprecision of the contour function pl_t. The value $\nu = 1$ corresponds to the classical HMM model, in which we have no information on hidden states, whereas the value $\nu = 0$ corresponds to the supervised learning situation, in which the states are observed precisely. The model was trained using observed outputs $\mathbf{x}$ and uncertain labels $pl_1, \ldots, pl_T$ as explained in Subection III-B. The E^{2}M algorithm was run 10 times with random initial values of the parameters, and the best solution according to the observed-data likelihood was retained.

To assess the quality of learning, we used a test dataset of 1000 observations from the same distribution. The most probable state sequence was computed using the Viterbi algorithm, assuming no prior knowledge of hidden states in the test sequence. The difference between the true and predicted state sequences was assessed using the adjusted Rand index (ARI) [14]. We recall that this commonly used clustering performance measure is a corrected-for-chance version of the Rand index, which equals 0 on average for a random partition, and 1 when comparing two identical partitions.

The whole experiment (training and test data generation, learning) was repeated 30 times. The results are shown in Figure 2 for $T = 100$ and $T = 300$. We can see that the quality of the results degrades gracefully from the fully supervised ($\nu = 0$) to the fully unsupervised ($\nu = 1$) case. When a longer sequence is used for training ($T = 300$), the influence of partial knowledge of hidden states is less important. However, even very imprecise labels ($\nu = 0.9$) can still improve the robustness of the results, as can be seen from the smaller dispersion of ARI values.

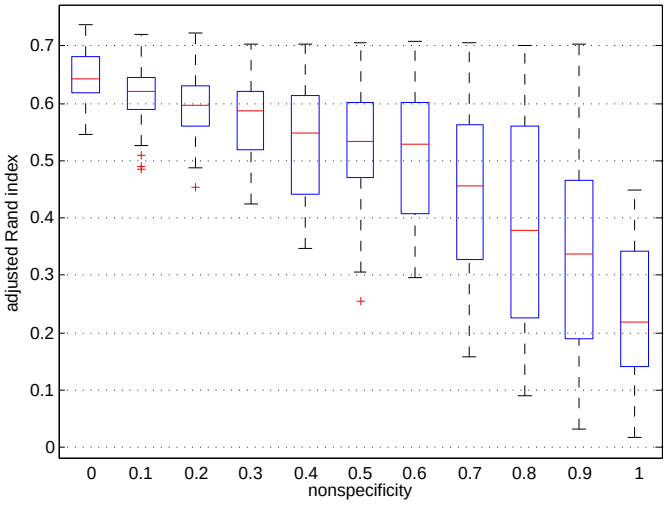
2) Influence of labeling error: In the previous experiment, information on hidden states was assumed to be always exact, i.e., the true state had the largest plausibility value. To simulate the more realistic situation in which information on states may be wrong, we proceeded as proposed in [7] and [10]. At each time step t, an error probability q_t was drawn randomly from a beta distribution with mean ρ and standard deviation 0.2. With probability q_t, the state y_t was then replaced by a completely random value $\widetilde{y}_k$ (with a uniform distribution over Ω). The plausibilities pl_{tk} were then determined as

$$pl_{tk} = P(y_t = k|\widetilde{y}_t) = \begin{cases} q_t/K + 1 - q_t & \text{if } \widetilde{y}_t = k, \\ q_t/K & \text{otherwise.} \end{cases} \quad (40)$$

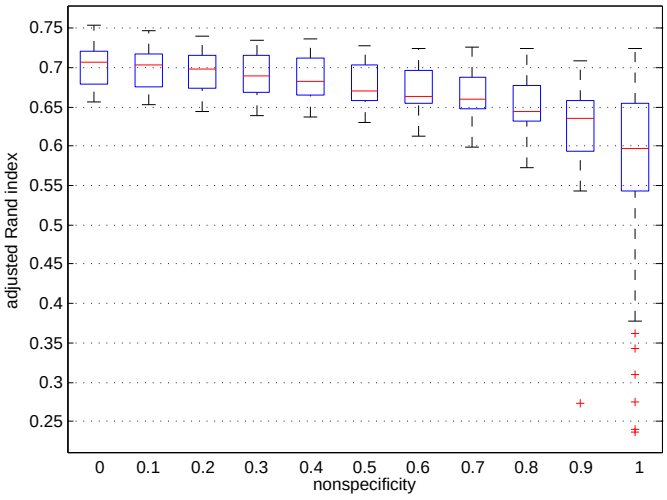
We can remark than the uncertain labels generated in this way are all the more imprecise that the error probability is high: in particular, we have $pl_{tk} = y_{tk}$ when $q_t = 0$ and $pl_{tk} = 1/K$ for all k when $q_t = 1$.

Training and test data sets were generated as in the previous section, and results were evaluated in the same way. For each randomly generated data set, the E^{2}M algorithm was run with uncertain labels pl_{ik}, noisy labels $\widetilde{y}_{ik}$ and no information on states.

Figure 3 shows the ARI as a function of mean error ρ for $T = 100$ (left) and $T = 300$ (right). As expected, a degradation of the segmentation quality is observed when the mean error probability ρ increases, with the ARI tending to a value close to zero as ρ tends to 1 when noisy labels are used for training. More importantly, Figure 3 shows that the use of partial information on states in the form of uncertain labels allows us to reach better segmentation results than those obtained using noisy labels. In particular, results never get worse than those obtained in the unsupervised case. These results show that our method is able to exploit additional information on observation uncertainty, when such information is available.



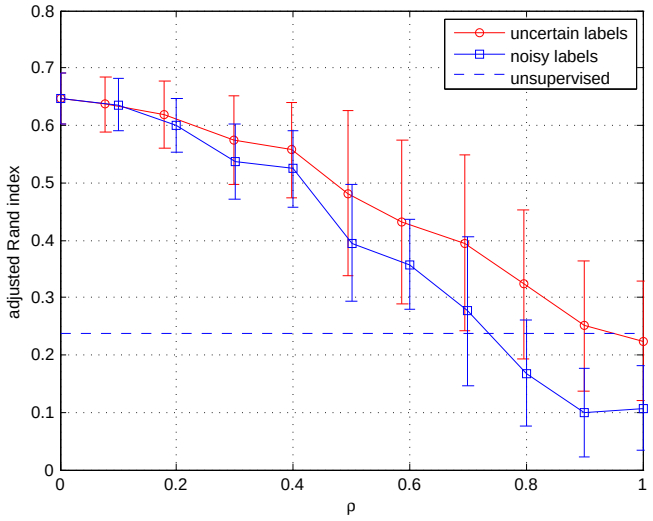
(a) $T = 100$



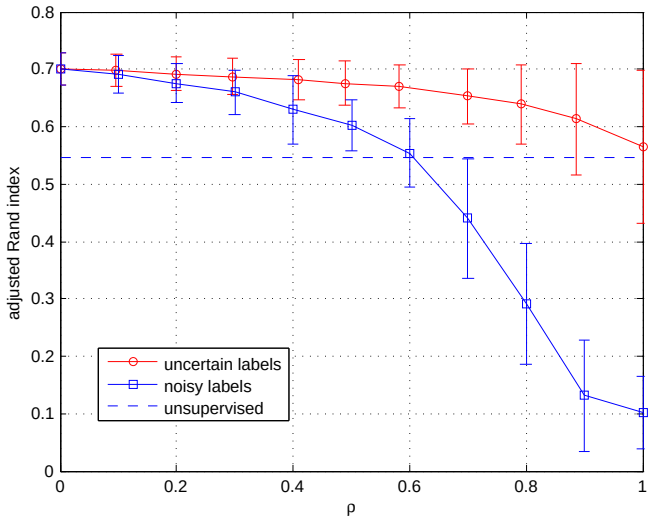
(b) $T = 300$

Fig. 2. Boxplots of the adjusted Rand index as a function of the nonspecificity coefficient ν over 30 repetitions, for learning datasets of $T = 100$ (left) and $T = 300$ (right) observations.

3) Influence of partial knowledge of state in the recognition phase: As shown in Subsection III-C, the Viterbi algorithm can be adapted to find the most likely state sequence for new data, based on observed outputs and partial observation of states provided by uncertain labels. To assess the influence of partial knowledge on states *in the test sequence*, we carried out the following experiment. Parameters were estimated using a sequence of $T = 300$ observations with no information of states, and uncertain labels were generated for the test



(a) $T = 100$



(b) $T = 300$

Fig. 3. Average values (plus and minus one standard deviation) of the adjusted Rand index over the 30 repetitions, as a function of the mean error probability ρ for learning datasets of $T = 100$ (left) and $T = 300$ (right) observations.

sequence of 1000 observations, with random labeling noise simulated as explained previously. The modified Viterbi algorithm described in Subsection III-C was used to segment the test sequence. Again, the whole process was repeated 30 times.

The results are shown in Figure 4. When $\rho = 0$, the test labels are known with no error and the ARI equals one. As the mean error probability ρ tends to one, the ARI between true and noisy labels tends to zero. However, using the observed output sequence and the uncertain labels, the Viterbi algorithm successfully exploit partially reliable information on states to compute meaningful partitions of the test sequence, with ARI values not exceeding those obtained with no information of test labels.

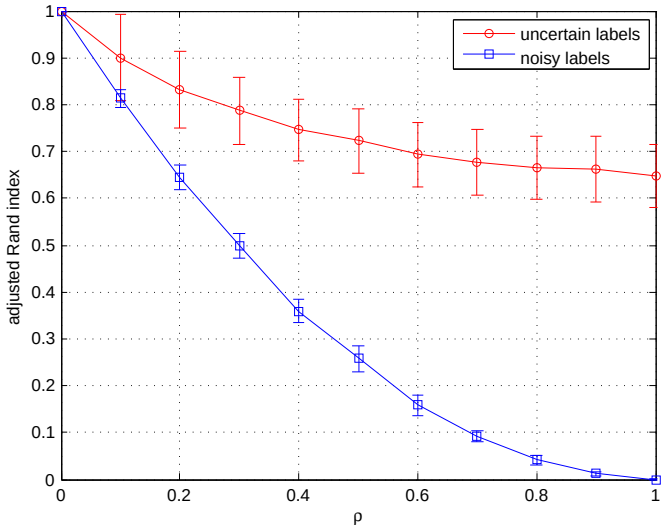


Fig. 4. Average values (plus and minus one standard deviation) of the adjusted Rand index over the 30 repetitions, as a function of the mean error probability ρ for test labels.

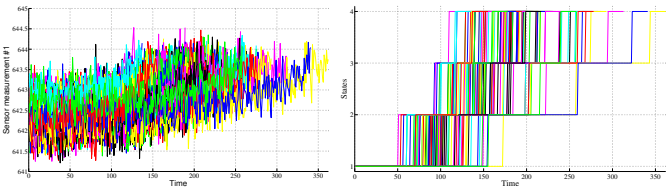
B. *Machine condition data*

As mentioned in Section I, uncertain information about states is typically available a posteriori in machine supervision applications, where experts may express probabilistic judgements about the machine condition at different times. To demonstrate the ability of our method to exploit partial information on states in this kind of applications, we used realistic machine condition data generated by an engine degradation simulator. The dataset, the experimental settings and the obtained results are described below.

1) Data description: A turbofan engine degradation simulator was designed at the NASA Prognostics Center of Excellence [24]. Several operating conditions (such as altitude or temperature) and fault modes were considered to cover a wide variety of situations. The simulation model was built using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) developed at the NASA Army Research Laboratory. By modifying 13 health parameters in C-MAPSS, the user can simulate the effects of faults and deterioration in any of the engine's five rotating components, including fan, LPC (Low-Pressure Compressor), HPC (High-Pressure Compressor), HPT (High-Pressure Turbine), and LPT (Low-Pressure Turbine) [29].

A dataset created using this simulator was first proposed to the 2008 Prognostics and Health Management (PHM) Data Challenge competition; the data was only described as run-to-failure time series with 21 dimensions, including temperature, pressure, and speed at various points, from multiple instances of an unspecified engineering system [29]. Data from the same engine model were collected by running the simulation several times under different flight conditions. No failure mode information was provided. NASA provided four data sets generated from four independent experiments with different settings such that only instances in the same data set can be considered from identical systems.

For each run of the simulation, the engine experienced



(a) The first feature for each time-series. (b) States for the each time-series.

Fig. 5. Features and states for the 100 time-series in the dataset.

complete run-to-failure operations, i.e., starting from brand new (with different degrees of initial wear and manufacturing variation), developing faults over a number of flights from one location to another, and finally reaching the failure condition measured by a set of predefined criteria. Depending on various factors, the amount and rate of damage accumulation for each engine instance are different, causing variable engine life.

2) Experimental settings: Only the first training dataset composed of 100 time series was used in this experient. Features 7, 9 and 16 were considered, which are among those shown in [29] to be the most relevant. Each time series in this dataset was manually segmented into four states [23]: *normal*, *transient*, *degrading* and *broken* modes. These "true" labels[1] were used to assess the performances of our method in segmenting the data, based on incomplete and partially reliable prior information on states.

The first feature for the complete dataset and the corresponding true states are represented in Figures 5(a) and 5(b), respectively. These figures show that the modeling of these time-series is difficult partly because of the great variability of possible durations in each state, which makes the detection of the functioning state quite difficult.

The performances of the PHMM learning algorithm was studied as a function of the quantity and quality of the partial information on states. The quantity of information was tuned by varying the proportion N of labeled data between 0 % (corresponding to unsupervised learning) and 100% (corresponding to fully supervised learning), by 25 % increments. The quality of labels was set by simulating labeling error as in Section IV-A2 (noisy labels). The emission probabilities in each state were assumed to be Gaussian and the parameters were estimated using ten-fold cross-validation.

3) Results: Figure 6 shows the distribution of the ARI (over 15 runs of the algorithm with different random labels) for different proportions N of labeled data and mean labeling error probabilities ρ. As in the previous experiment, we can observe that the performances are only mildly affected by labeling error. The median value of the ARI also does not depend much on the proportion of labeled data; however, the number of outliers (trials with very low ARI values) is much larger for small N: labeled data thus improve the robustness of the learning algorithm to initial conditions.

Moreover, learning time increases with noise level, as reported in Figure 7, which shows the number of iterations

[1]Available at http://www.femto-st.fr/~emmanuel.ramasso/PEPS_INSIS_2011_PHM_by_belief_functions.html.

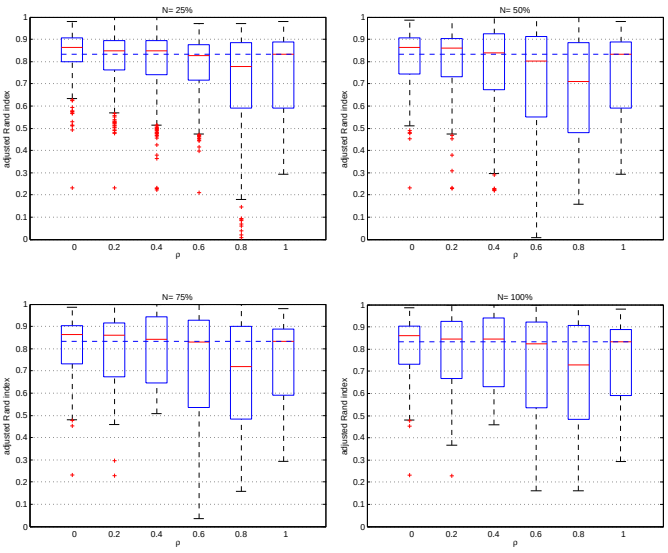


Fig. 6. Adaptive Rand Index as a function of labeling error ρ for different proportions N of labeled data. The dotted line corresponds to the unsupervised case.

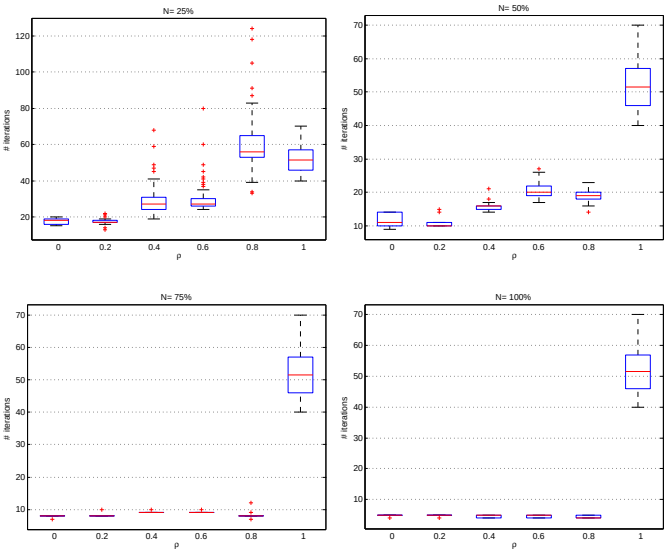


Fig. 7. Number of iterations as a function of labeling error ρ for different proportions N of labeled data.

of the E^{2}M algorithm for different proportions N of labeled data and mean labeling error probabilities ρ. As in Figure 6, the unsupervised case corresponds to $\rho = 1$ for any value of N. Interestingly, labeling 75 % of the data with a mean error rate up to 80 % allows us to reach convergence in less than 10 iterations, which is five times less than the number of iterations required in the unsupervised case. With 25 % of labeled data and a mean error rate of 60%, the gain in training time is still around 50 %. These results show that even very imprecise and uncertain information of states may drastically reduce the training time for HMMs.

V. Conclusion

In classical statistics and data analysis, observations are usually assumed to be precise and perfectly reliable. Latent variable models such a HMMs include both observed and unobserved (latent) variables. In some applications, however, a human expert, an unreliable sensor or an indirect measurement device may provide imprecise and/or partially reliable information on some of the variables. We then need to represent such partial information and exploit it for statistical inference.

In this paper, this problem has been addressed in the particular case of HMMs. Partial knowledge of hidden states has been assumed to be available and represented by belief functions. The E^{2}M algorithm, a variant of the EM algorithm for evidential data, has been particularized for this model, resulting in modified Baum-Welch update equations for parameter learning. The problem of finding the most probable state sequence based on observed outputs and partial information on states has also been solved using the variant of the Viterbi algorithm.

The proposed approach was validated using both simulated data and realistic engine condition data generated by the C-MAPSS simulation software developed by NASA. The use of partial information on states was shown to allow for improved performances and faster convergence of the learning algorithm in time series segmentation tasks. In particular, the performances were shown to improve gradually when the quantity and quality of data increase.

The proposed approach is very general and can be extended to any continuous and/or discrete latent variable models, including more general forms of dynamic Bayesian networks such as, in particular, pairwise or triplet Markov chains. More work is also needed to develop rigorous elicitation procedures allowing us to capture expert opinions in the form of belief functions. First steps in this direction have been reported in [6].

Ackowledgment

This work has been supported by a PEPS-INSIS-2011 grant from the French National Center for Scientific Research (CNRS) under the administrative authority of the French Ministry of Research. It was carried out in the framework of the Laboratories of Excellence MS2T and ACTION, which were funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (references ANR-11-IDEX-0004-02 and ANR-11-LABX-01-01). We thank the anonymous referees for their helpful comments.

References

[1] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, 41:164–171, 1970.

[2] A. Bendjebbour, Y. Delignon, L. Fouque, V. Samson, and W. Pieczynski. Multisensor images segmentation using Dempster-Shafer fusion in Markov fields context. *IEEE Trans. on Geoscience and Remote Sensing*, 39(8):1789–1798, 2001.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] M. Y. Boudaren, E. Monfrini, W. Pieczynski, and A. Assani. Dempster-shafer fusion of multisensor signals in nonstationary Markovian context. *EURASIP Journal on Advances in Signal Processing*, 2012(134), 2012.

[5] Yi-Chung Cheng and Sheng-Tun Li. Fuzzy time series forecasting with a probabilistic smoothing hidden markov model. *IEEE Transactions on Fuzzy Systems*, 20(2):291–304, 2012.

[6] Z. L. Cherfi, L. Oukhellou, E. Côme, T. Denoeux, and P. Aknin. Partially supervised independent factor analysis using soft labels elicited from multiple experts: Application to railway track circuit diagnosis. *Soft Computing*, 16(5):741–754, 2012.

[7] E. Côme, L. Oukhellou, T. Denoeux, and P. Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, 42(3):334–348, 2009.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–38, 1977.

[9] T. Denoeux. Maximum likelihood estimation from fuzzy data using the fuzzy EM algorithm. *Fuzzy Sets and Systems*, 18(1):72–91, 2011.

[10] T. Denoeux. Maximum likelihood estimation from uncertain data in the belief function framework. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):119–130, 2013.

[11] M. Dong and D. He. A segmental hidden semi-markov model (HSMM) -based diagnostics and prognostics framework and methodology. *Mech. Syst. Signal Processing*, 21:2248–2266, 2007.

[12] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[13] L. Fouque, A. Appriou, and W. Pieczynski. An evidential Markovian model for data fusion and unsupervised image classification. In *Proceedings of 3rd International Conference on Information Fusion (FUSION 2000)*, volume 1, pages TuB4–25 – TuB4–31, Paris, France, 2000.

[14] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–ñ218, 1985.

[15] P. Lanchantin and W. Pieczynski. Unsupervised restoration of hidden non stationary Markov chain using evidential priors. *IEEE Transactions on Signal processing*, 53(8):3091–3098, 2005.

[16] K. P. Murphy. *Dynamic Bayesian Networks: Representation, inference and learning*. PhD thesis, UC Berkeley, 2002.

[17] W. Pieczynski. Chaînes de Markov triplet, triplet Markov chains. *Comptes Rendus de l'Académie des Sciences – Mathématique, Série I*, 335(3):275–278, 2002.

[18] W. Pieczynski. Pairwise Markov chains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):634–639, 2003.

[19] W. Pieczynski. Multisensor triplet Markov chains and theory of evidence. *International Journal of Approximate Reasoning*, 45:1–16, 2007.

[20] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77:257–285, 1989.

[21] E. Ramasso. Contribution of belief functions to Hidden Markov Models. In *IEEE Workshop on Machine Learning and Signal Processing*, pages 1–6, Grenoble, France, October 2009.

[22] E. Ramasso, M. Rombaut, and D. Pellerin. State filtering and change detection using TBM conflict - application to human action recognition in athletics videos. *IEEE Trans. on Circuits and Systems for Video Technology*, 17(7):944–949, 2007.

[23] E. Ramasso, M. Rombaut, and N. Zerhouni. Joint prediction of observations and states in time-series based on belief functions. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 43(1):37–50, 2013.

[24] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *Int. Conf. on Prognostics and Health Management*, pages 1–9, Denver, CO, USA, 2008.

[25] L. Serir, E. Ramasso, and N. Zerhouni. Time-sliced temporal evidential networks: the case of evidential HMM with application to dynamical system analysis. In *IEEE Int. Conf. on Prognostics and Health Management*, pages 1–10, Denver, CO, USA, 2011.

[26] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, N.J., 1976.

[27] H. Soubaras. On evidential Markov chains. In L. Maddalena et al., editor, *Proceedings of IPMU'08*, pages 386–393, Torremolinos, Spain, 2008.

[28] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, 1967.

[29] T. Wang. *Trajectory Similarity Based Prediction for Remaining Useful Life Estimation*. PhD thesis, University of Cincinnati, 2010.

[30] A. Wilson and A. Bobick. Hidden Markov models for modeling and recognizing gesture under variation. *Int. Jour. of Pattern Recognition and Artificial Intelligence*, 15(1):123–160, 2001.

APPENDIX

Our proof of Proposition 1 is based on the formally equivalent probabilistic model described in Subsection III-A. Omitting $\boldsymbol{\theta}^{(q)}$ to simplify the notations, we have

$$\gamma_{tk}^{(q)} = p(Y_t = k|\mathbf{x}, pl) = p(Y_t = k|\mathbf{x}, \mathbf{u}) = \frac{p(\mathbf{x}, \mathbf{u}|Y_t = k)p(Y_t = k)}{p(\mathbf{x}, \mathbf{u})}, \quad (41)$$

where, as before, $\mathbf{u}$ denotes a realization of $\mathbf{U}$ assumed to be a vector of 1's. From (22a)-(22d), we can see that the denominator in the previous expression is $L(\boldsymbol{\theta}^{(q)}; \mathbf{x}, pl)$. Making use of conditional independence properties as well as the product rule of probability, we obtain for the numerator of (25):

$$\begin{aligned} &p(\mathbf{x}, \mathbf{u}|Y_t = k)p(Y_t = k) \\ &= p(\mathbf{x}_{1:t}, \mathbf{u}_{1:t}|Y_t = k)p(\mathbf{x}_{t+1:T}, \mathbf{u}_{t+1:T}|Y_t = k)p(Y_t = k) \\ &\quad = p(\mathbf{x}_{1:t}, \mathbf{u}_{1:t}, Y_t = k)p(\mathbf{x}_{t+1:T}, \mathbf{u}_{t+1:T}|Y_t = k) \\ &\quad\quad = \alpha_{tk}^{(q)} \beta_{tk}^{(q)}, \end{aligned} \quad (42)$$

where we use the notation $\mathbf{x}_{1:t} = (x_1, \ldots, x_t)$ and a similar notation for $\mathbf{u}_{1:t}$, and $\alpha_{tk}^{(q)}$ and $\beta_{tk}^{(q)}$ are defined as

$$\alpha_{tk}^{(q)} = p(\mathbf{x}_{1:t}, \mathbf{u}_{1:t}, Y_t = k; \theta^{(q)}) \quad (43)$$

and

$$\beta_{tk}^{(q)} = p(\mathbf{x}_{t+1:T}, \mathbf{u}_{t+1:T}|Y_t = k; \theta^{(q)}). \quad (44)$$

These variables can be computed using the forward-backward [20][3, Chapter 13]. Using the same line of reasoning as followed in [3, p.620], it can be shown that

$$\alpha_{1k}^{(q)} = p(\mathbf{x}_1, U_1 = 1, Y_1 = k; \boldsymbol{\phi}^{(q)}) = \pi_k^{(q)} \, pl_{1k} \, p_t(x_1; \boldsymbol{\phi}^{(q)}), \quad (45a)$$

and

$$\begin{aligned} \alpha_{t,k}^{(q)} &= p(x_t, U_t = 1|Y_t = k) \\ &\quad \sum_{\ell} p(\mathbf{x}_{1:t-1}, \mathbf{u}_{1:t-1}, Y_{t-1} = \ell)p(Y_t = k|Y_{t-1} = \ell) \\ &\quad\quad = p_k(x_t; \boldsymbol{\phi}^{(q)}) pl_{tk} \sum_{\ell} \alpha_{t-1,\ell}^{(q)} \, a_{\ell k}^{(q)}, \end{aligned} \quad (45b)$$

for $t = 2, \ldots, T$.

Using (41) and (42) with $t = T$, we have

$$p(Y_T = k|\mathbf{x}, \mathbf{u}) = \frac{p(\mathbf{x}, \mathbf{u}, Y_t = k)\beta_{Tk}^{(q)}}{p(\mathbf{x}, \mathbf{u})}, \quad (46)$$

which implies that $\beta_{Tk}^{(q)} = 1$. Recursion equations for $\beta_{t,k}^{(q)}$ can be obtained as

$$\begin{aligned} \beta_{t,k}^{(q)} &= \sum_{\ell} p(\mathbf{x}_{t+2:T}, \mathbf{u}_{t+2:T}|Y_{t+1} = \ell) \\ &\quad p(\mathbf{x}_{t+1}, U_{t+1} = 1|Y_{t+1} = \ell)p(Y_{t+1} = \ell|Y_t = k) \\ &\quad\quad = \sum_{\ell} \beta_{t+1,\ell}^{(q)} \, p_\ell(\mathbf{x}_{t+1}; \boldsymbol{\phi}^{(q)}) \, pl_{t+1,\ell} \, a_{k\ell}^{(q)}. \end{aligned} \quad (47)$$

Now,

$$\begin{aligned} \xi_{t-1,t,k,\ell}^{(q)} &= p(Y_{t-1} = k, Y_t = \ell|\mathbf{x}, pl) && (48a) \\ &= p(Y_{t-1} = k, Y_t = \ell|\mathbf{x}, \mathbf{U} = \mathbf{u}) && (48b) \\ &= \frac{p(\mathbf{x}, \mathbf{u}|Y_{t-1} = k, Y_t = \ell)p(Y_{t-1} = k, Y_t = \ell)}{p(\mathbf{x}, \mathbf{u})} && (48c) \\ &= \frac{\alpha_{t-1,k}^{(q)} \;\; p_\ell(x_t; \boldsymbol{\phi}_\ell^{(q)}) pl_{t\ell} \, a_{k\ell}^{(q)} \;\; \beta_{t\ell}^{(q)}}{p(\mathbf{x}, \mathbf{u})}, && (48d) \end{aligned}$$

where we have made use of the following conditional independence property:

$$\begin{aligned} &p(\mathbf{x}, \mathbf{u}|Y_{t-1} = k, Y_t = \ell) = \\ &\quad p(\mathbf{x}_{1:t-1}\mathbf{u}_{1:t-1}|Y_{t-1} = k)p(x_t, U_t = 1|Y_t = \ell) \\ &\quad\quad p(\mathbf{x}_{t+1:T}, \mathbf{u}_{t+1:T}|Y_t = \ell). \end{aligned} \quad (49)$$

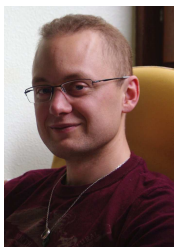
Finally, from (25), we get:

$$L(\boldsymbol{\theta}; \mathbf{x}, pl) = p(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_{tk}\beta_{tk} \quad (50)$$

for any t. In particular, taking $t = T$, we get:

$$L(\boldsymbol{\theta}; \mathbf{x}, pl) = \sum_{k=1}^{K} \alpha_{Tk}, \quad (51)$$

which completes the proof.



Emmanuel ramasso E. Ramasso received both B.Sc. and M.Sc. degrees in Computer Science and Automation from the University of Savoie in 2004, and earned his Ph.D. from the University of Grenoble in 2007. He pursued with a postdoc at the Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA) in 2008. Since 2009, he has been working as an associate professor at the National School of Engineering in Mechanics and Microtechnics (ENSMM) at Besanon (France). His research is carried out at FEMTO-ST institute and focused on pattern recognition under uncertainties with applications to Prognostics and Structural Health Management.

Thierry Denoeux Thierry Denoeux graduated in 1985 as an engineer from the Ecole des Ponts ParisTech in Paris, and received a doctorate from the same institution in 1989. Currently, he is Full Professor (Classe Exceptionnelle) with the Department of Information Processing Engineering at the Université de Technologie de Compiègne (UTC), France, and deputy director of the Heudiasyc research Lab (UMR 7253). His research interests concern the management of uncertainty in intelligent systems. His main contributions are in the theory of belief functions with applications to pattern recognition, data mining and information fusion. He is the Editor-in-Chief of the *International Journal of Approximate Reasoning*, and an Associate Editor of several journals including *Fuzzy Sets and Systems* and *IEEE Transactions on Fuzzy Systems*.