Available online at www.sciencedirect.com
SCIENCE DIRECT®

Pattern Recognition 38 (2005) 1349–1371

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms

P. Dupont[a,*], F. Denis[b], Y. Esposito[b]

[a] *INGI, Université catholique de Louvain, Place Sainte-Barbe 2, B-1348 Louvain-la-Neuve, Belgium*
[b] *LIF-CMI, UMR 6166, 39, Rue F. Joliot Curie, 13453 Marseille Cedex 13, France*

Received 17 March 2004; accepted 17 March 2004

Abstract

This article presents an overview of Probabilistic Automata (PA) and discrete Hidden Markov Models (HMMs), and aims at clarifying the links between them. The first part of this work concentrates on probability distributions generated by these models. Necessary and sufficient conditions for an automaton to define a probabilistic language are detailed. It is proved that probabilistic deterministic automata (PDFA) form a proper subclass of probabilistic non-deterministic automata (PNFA). Two families of equivalent models are described next. On one hand, HMMs and PNFA with no final probabilities generate distributions over complete finite prefix-free sets. On the other hand, HMMs with final probabilities and probabilistic automata generate distributions over strings of finite length. The second part of this article presents several learning models, which formalize the problem of PA induction or, equivalently, the problem of HMM topology induction and parameter estimation. These learning models include the PAC and identification with probability 1 frameworks. Links with Bayesian learning are also discussed. The last part of this article presents an overview of induction algorithms for PA or HMMs using state merging, state splitting, parameter pruning and error-correcting techniques.
© 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Probabilistic automata; Hidden Markov models; Grammar induction; PAC learning; Bayesian learning; Induction algorithms; HMM topology learning

1. Introduction

Hidden Markov Models (HMMs) are widely used in many pattern recognition areas, including applications to speech recognition [1–4], biological sequence modeling [5,6], information extraction [7] and optical character recognition [8], to name a few. In many of these cases, the model structure, also referred to as topology, is defined according to some prior knowledge of the application domain. In some cases however, attempts are made to induce automatically the model structure from training data. The learning problem combines then structural induction and parameter estimation.

Grammar Induction, also known as Grammatical Inference, is a collection of techniques for learning grammars from training data [9–12]. Early works on grammar induction already covered learning techniques for probabilistic (or stochastic[1]) grammars [13–16]. Probabilistic regular grammars form a particular class of interest. These models

* Corresponding author. Tel.: +32 10 479114; fax: +32 10 450345.
E-mail addresses: pdupont@info.ucl.ac.be (P. Dupont), fdenis@cmi.univ-mrs.fr (F. Denis), esposito@cmi.univ-mrs.fr (Y. Esposito)
URLs: http://www.info.ucl.ac.be/∼pdupont/ (P. Dupont), http://www.cmi.univ-mrs.fr/∼fdenis/ (F. Denis).

[1] We consider that the term *stochastic* qualifies a process, while the term *probabilistic* qualifies a model of such process. We use therefore the term probabilistic grammars (or probabilistic automata) since we consider them as models.

0031-3203/$30.00 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2004.03.020

are equivalent to certain types of probabilistic automata (PA), for which several induction techniques have been proposed [17–22].

This article presents an overview of probabilistic automata and discrete HMMs, and aims at clarifying the links between them. These links allow to apply induction techniques and learnability results developed in one formalism to the other.

The first part of this work (Sections 2 and 3) concentrates on probability distributions generated by PA and HMMs. Necessary and sufficient conditions for an automaton to define a probabilistic language are detailed. The distinction between probabilistic deterministic automata (PDFA) and probabilistic non-deterministic automata (PNFA) is introduced. This distinction matters for the learning problem as it is proved in Section 3 that PDFA form a proper subclass of PNFA. Two families of equivalent models are described next. On one hand, HMMs and PNFA with no final probabilities generate distributions over complete finite prefix-free sets. On the other hand, HMMs with final probabilities and probabilistic automata generate distributions over strings of finite length.

The second part of this article (Sections 4 and 5) presents several learning models. Learning a probabilistic automaton aims, in a broad sense, at inducing an automaton generating a distribution $\hat{P}$ from a sample drawn according to some unknown target distribution P. The distribution $\hat{P}$ forms the learned hypothesis that approximates the target. The purpose of a learning model is to formalize the notion of learning when a specific quality measure defines the distance between P and $\hat{P}$. We discuss adaptations of the PAC learning and identification in the limit frameworks to the learning of probabilistic automata. Links with Bayesian learning are also discussed. A learning model includes a learning protocol specifying the prior knowledge given to the learner, the required quality of the proposed hypothesis, and, possibly, some bounds on the computational complexity of the learning process. Once a learning model has been defined, the question of what can be learned by any algorithm following the learning protocol, can be addressed. Several learning results are presented in this context in Section 5.

The last part of this article (Section 6) presents an overview of induction algorithms for PA or HMMs. State merging is a generalization technique starting from an initial model fitting perfectly a given learning sample. An opposite approach is state splitting where a very general model is progressively specialized to best fit the training data. Structural induction can also be embedded into parameter estimation combined with parameter pruning. Finally, error-correcting techniques greedily adapt an initial structure by minimizing some edition costs to best incorporate new samples.

2. Probabilistic languages, automata and HMMs

Probabilistic languages are defined in Section 2.1. We discuss in Section 2.2 various equivalent definitions of semi-probabilistic automata. The main result of Section 2.3 is the Proposition 2 which establishes the necessary and sufficient conditions for a semi-probabilistic automaton to be probabilistic, that is, to define a distribution on words (or strings). Probabilistic automata considered in the present work can be considered as a representation of probabilistic regular grammars (see e.g. [16]). The notions of probabilistic non-deterministic versus deterministic automata are introduced next. This distinction matters, as demonstrated in Section 3, for the class of distributions generated by the latter form a proper subclass of the class of distributions generated by the former. Section 2.4 concentrates on probabilistic automata with no final probabilities and details the type of distributions they generate. Hidden Markov Models are described in Section 2.5.

2.1. Probabilistic languages

2.1.1. Notations

Σ denotes a finite *alphabet*, Σ^* (respectively Σ^∞) denotes the set of words of finite (respectively infinite) length over Σ. For any word $u \in \Sigma^*$, $u\Sigma^*$ (respectively $u\Sigma^\infty$) denotes the set of finite (respectively infinite) words with prefix u. ε denotes the *empty word* and $|u|$ the *length* of a word u. For any $n \in \mathbb{N}$, Σ^n (respectively $\Sigma^{\leqslant n}$) denotes the set of words of length n (respectively less or equal to n).

Definition 1. Let Σ be a finite alphabet, a *semi-distribution* over Σ^* is a function $\psi : \Sigma^* \to [0, 1]$ satisfying $\sum_{u\in\Sigma^*} \psi(u) \leqslant 1$.

Definition 2. The *support* $L_\psi \subseteq \Sigma^*$ of the semi-distribution ψ is the language $L_\psi = \{u \in \Sigma^* | \psi(u) > 0\}$.

Definition 3. A *distribution* or *probabilistic language* ψ over Σ^* is a semi-distribution such that $\sum_{u\in\Sigma^*} \psi(u) = 1$.

2.2. Semi-probabilistic automata

Definition 4. A *semi-probabilistic automaton*[2] (semi-PA) is a 5-tuple $\langle \Sigma, Q, \phi, \iota, \tau\rangle$ where Σ is a finite alphabet, Q is a finite set of states, $\phi : Q \times \Sigma \times Q \to [0, 1]$ is a mapping defining the transition probability function, $\iota : Q \to [0, 1]$ is a mapping defining the initial probability of each state, and $\tau : Q \to [0, 1]$ is a mapping defining the final probability of each state. The following constraints must

[2] Such an automaton is called a semi-PA and not a PA as it defines a semi-distribution (see Corollary 1). The supplementary conditions to be satisfied to define a distribution are detailed in Definition 9.

Download English Version:

https://daneshyari.com/en/article/10360630

Download Persian Version:

https://daneshyari.com/article/10360630

Daneshyari.com