SUBSYMBOL-SYMBOL-TRANSDUKTOREN

Matthias Wolff[1]*, Constanze Tschöpe*[2]*, Ronald Römer*[1] *und Günther Wirsching*[3]

[1]*BTU Cottbus,* [2]*Fraunhofer IZFP Dresden,* [3]*KU Eichstätt-Ingolstadt*
matthias.wolff@tu-cottbus.de

Kurzfassung: In [1, 2, 3] haben wir eine Formulierung von kontinuierlichen Hidden-Markov-Modellen (CD-HMM) als endliche Transduktoren (*finite state transducers*, FST) vorgeschlagen. In diesem Beitrag entwickeln wir diesen Ansatz weiter und zeigen, dass so formulierte Hidden-Markov-Modelle als Komposition aus elementaren Subsymbol-Symbol-Transduktoren (SST), einem Mischungstransduktor und dem klassischen „versteckten" Zustandsautomaten aufgefasst werden können.

Der Vorteil dieser Sichtweise liegt zum einen in einer klaren Trennung der klassischen *endlichen* Komponente (FSM) von der notwendigen Erweiterung auf ein *unendliches* (kontinuierliches) Eingabealphabet in Form des neu zu definierenden Subsymbol-Symbol-Transduktors. Zum anderen erlaubt sie eine mathematisch saubere Behebung des potenziellen Konflikts zwischen dem Gewichtshalbring der Subsymbol-Symbol-Übersetzung (typisch: logarithmischer Halbring), dem Gewichtshalbring der Mischung von Eingabeverteilungsdichten (typisch: logarithmischer Halbring) und dem Gewichtshalbring des versteckten Automaten (tropischer Halbring bei Viterbi-Dekodierung).

1 Gewichtete endliche Transduktoren (wFST)

1.1 Definition nach Mohri

Wir gehen von der Definition gewichteter endlicher Transduktoren (*weighted finite state transducers*, wFST) nach M. Mohri [4, S. 3f] aus:

> **Definition 1** *A weighted transducer $\mathcal{T}$ over a semiring $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ is an 8-tuple $\mathcal{T} = (X, Y, Z, I, F, E, \lambda, \rho)$ where X is a finite input alphabet, Y a finite output alphabet, Z is a finite set of states, $I \subseteq Z$ the set of initial states, $F \subseteq Z$ the set of final states, E a finite multiset of transitions, which are elements of $Z \times (X \cup \varepsilon\}) \times (Y \cup \varepsilon\}) \times \mathbb{K} \times Z$, $\lambda : I \to \mathbb{K}$ an initial weight function, and $\rho : F \to \mathbb{K}$ a final weight function mapping F to $\mathbb{K}$.*[1]

Ein solcher Transduktor ordnet nach [4] Paaren $(\mathbf{x}, \mathbf{y}) \in X^* \times Y^*$ von Zeichenketten das Gewicht

$$\mathcal{T}(\mathbf{x}, \mathbf{y}) = \bigoplus_{U \in \mathcal{P}(I, \mathbf{x}, \mathbf{y}, F)} \lambda(p[U]) \otimes w(U) \otimes \rho(n[U]) \tag{1}$$

zu. Dabei steht $U \in \mathcal{P}(I, \mathbf{x}, \mathbf{y}, F)$ für einen durchgehenden Weg, welcher die Eingabezeichenkette $\mathbf{x}$ akzeptiert und die Ausgabezeichenkette $\mathbf{y}$ generiert, und $p(U) \in I$ für den Anfangszustand sowie $n(U) \in F$ für den Schlusszustand dieses Weges.

[1] Wir haben uns erlaubt, einige Formelzeichen in Mohris Definition an unsere Notation anzupassen.

1.2 Erweiterte Definition

Wir erweitern Definition 1 und schreiben sie etwas um:

Definition 2 *Ein gewichteter endlicher Transduktor $\mathcal{T}$ über einem Halbring $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ ist ein 6-Tupel $\mathcal{T} = (Z, I, F, X, Y, w)$ bestehend aus einer endlichen Menge Z von Zuständen, einer nicht-leeren Menge $I \subseteq Z$ von Anfangszuständen, einer nicht-leeren Menge $F \subseteq Z$ von Schlusszuständen, einem (nicht notwendig endlichen) Eingabealphabet X, einem (nicht notwendig endlichen) Ausgabealphabet Y sowie einer (algorithmisch definierten) Verhaltensfunktion $w : Z \times X^\circ \times Y^\circ \times Z \to \mathbb{K}$ mit der Notation $X^\circ := X \cup \{\varepsilon\}$ und $Y^\circ := Y \cup \{\varepsilon\}$.*

Die Unterschiede zu Definition 1 nach Mohri sind:

1. Die Ein- und Ausgabealphabete X und Y können unendlich sein.
2. Die Multimenge E der Kanten wurde durch die Verhaltensfunktion w ersetzt.
3. Es wurde auf die Definition von Anfangs- und Schlussgewichten (λ und ρ) verzichtet.[2]

Auch diese Form des Transduktors übersetzt Eingabezeichenketten $\mathbf{x} \in X^*$ in Ausgabezeichenzeichenketten $\mathbf{y} \in Y^*$ und ordnet dieser Übersetzung das Gewicht

$$\mathcal{T}(\mathbf{x}, \mathbf{y}) = \bigoplus_{U \in \mathcal{P}(I, \mathbf{x}, \mathbf{y}, F)} w(U) = \bigoplus_{U \in \mathcal{P}(I, \mathbf{x}, \mathbf{y}, F)} \bigotimes_{e^k \in U} w(e^k) \tag{2}$$

zu wobei e^k für den k-ten Übergang in der Übergangsfolge (d. h. im Weg) U steht. Wir verwenden tiefgestellte Indizes für *Mengen*elemente und hochgestellte Indizes für *Folgen*elemente. Die Bezeichung x_n^k steht beispielsweise für das n-te Element einer Menge an der k-ten Stelle einer Folge.

In Anlehnung an [4] verwenden wir folgende Notationen für Wege

- $\mathcal{P}(Z_1, Z_2)$: Menge aller Wege von Zuständen $Z_1 \in Z$ zu Zuständen $Z_2 \in Z$,
- $\mathcal{P}(Z_1, \mathbf{x}, Z_2)$: Teilemenge der Wege $\mathcal{P}(Z_1, Z_2)$, welche die Eingabe $\mathbf{x}$ akzeptiert.
- $\mathcal{P}(Z_1, \infty, \mathbf{y}, Z_2)$: Teilemenge der Wege $\mathcal{P}(Z_1, Z_2)$, welche die Ausgabe $\mathbf{y}$ generiert.[3]
- $\mathcal{P}(Z_1, \mathbf{x}, \mathbf{y}, Z_2)$: Teilemenge der Wege $\mathcal{P}(Z_1, \mathbf{x}, Z_2)$ und $\mathcal{P}(Z_1, \infty, \mathbf{y}, Z_2)$, welche die Eingabe $\mathbf{x}$ akzeptiert *und* die Ausgabe $\mathbf{y}$ generiert.

1.3 Formen

Wir unterscheiden folgende Formen von gewichteten endlichen Automaten:

- Transduktor $\mathcal{T}(\mathbf{x}, \mathbf{y})$ (siehe oben),

[2] Dies geschieht hauptsächlich aus Gründen der Lesbarkeit. Anfangs- und Schlussgewichte können durch ε-Übergänge realisiert werden. In der Praxis kann die Verwendung von Anfangs- und Schlussgewichten dennoch zweckmäßig sein, da sich dann einige wFST-Algorithmen [4] vereinfachen lassen.

[3] Die Notation ∞ stammt aus der Definition des Zeichenkettenhalbrings [5], in dem ∞ das neutrale Element bezüglich der durch das längste gemeinsame Präfix definierten Zeichenkettenaddition ist.

- Akzeptor $\mathcal{A}(\mathbf{x})$: Transduktoren ohne Ausgabe, ein Transduktor kann durch Projektion [4] in einen Akzeptor überführt werden

$$\mathcal{A}(\mathbf{x}) = \downarrow \mathcal{T}(\mathbf{x}) = \bigoplus_{\mathbf{y}} \mathcal{T}(\mathbf{x}, \mathbf{y}), \tag{3}$$

- Generator $\mathcal{G}(\infty, \mathbf{y})$: Transduktoren ohne Eingabe, ein Transduktor kann durch Inversion und Projektion [4] in einen Generator überführt werden

$$\mathcal{G}(\infty, \mathbf{y}) = \left[\downarrow \left(\mathcal{T}^{-1} \right) \right]^{-1} (\mathbf{y}) = \bigoplus_{\mathbf{x}} \mathcal{T}(\mathbf{x}, \mathbf{y}). \tag{4}$$

Die Inversion eines Akzeptors ist ein Generator und umgekehrt. Im Zusammenhang mit Hidden-Markov-Modellen kommt endlichen Generatoren eine besondere Bedeutung zu.

2 Subsymbol-Symbol-Transduktoren (SST)

Die Aufgabe eines Subsymbol-Symbol-Transduktors ist, Objekte unterhalb der Symbolebene in Objekte auf Symbolebene zu übersetzen. Insbesondere haben wir dabei die Situation in der automatischen Spracherkennung vor Augen, bei der einem Merkmalvektor zu jedem Element aus einer vorgelegten endlichen Menge von phonetischen Einheiten (z. B. Phoneme oder Triphone) eine Likelihood zugeordnet wird. Grundlage der formalen Definition ist die Vorstellung, dass „Subsymbole“ Elemente eines kontinuierlichen Maßraums sind, während jedes „Symbol“ durch eine Wahrscheinlichkeitsdichte auf diesem dargestellt ist. In der akustischen Mustererkennung ist der kontinuierliche Maßraum typischerweise ein hochdimensionaler reeller Vektorraum, und die Wahrscheinlichkeitsdichten sind in der Regel gewichtete Mischungen von Gauß- oder Laplace-Verteilungsdichten, die zur Ermittlung einer Likelihood punktweise ausgewertet werden. Die folgende Definition ist eine abstrakte Beschreibung dieser Situation.

2.1 Konstruktion

Definition 3 *Ein Subsymbol-Symbol-Transduktor $\mathcal{Q}$ über einem Halbring $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ ist ein Tripel $\mathcal{Q} = (z_0, \mathcal{O}, Q)$ bestehend aus genau einem Zustand z_0, einer nicht-leeren Menge $\mathcal{O}$ von Subsymbolen sowie einer endlichen Menge $Q = \{q_1, \ldots, q_N\}$ von Symbolen, wobei jedes Symbol eine (algorithmisch definierte) Funktion $q_n : \mathcal{O} \to \mathbb{K}$ ist.*

Der einzige Zustand z_0 ist notwendigerweise Anfangs- und Schlusszustand. Die Menge $\mathcal{O}$ der Subsymbole ist typischerweise ein d-dimensionaler Merkmalvektorraum: $\mathcal{O} = \mathbb{R}^d$.

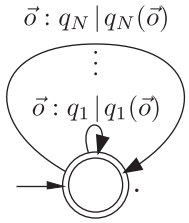
SSTs fallen unter Definition 2, wenn man $Z = I = F = \{z_0\}$, $X = \mathcal{O}$, $Y = Q$ sowie

$$w : \{z_0\} \times \mathcal{O} \times Q \times \{z_0\} \to \mathbb{K}, \quad w(z_0, \vec{o}, q_n, z_0) := q_n(\vec{o}) \tag{5}$$

setzt.

Die Verhaltensfunktion (5) definiert je eine Schleife für jede der Funktionen $q \in Q$ am Zustand z_0, deren Eingabesymbol ein Vektor $\vec{o} \in \mathcal{O}$, deren Ausgabesymbol die Funktion q_n und deren Gewicht der Funktionswert von q_n an der Stelle $\vec{o}$ ist. Die grafische Darstellung sieht wie folgt

aus:

$\mathcal{Q} =$ 

(6)

SSTs werden üblicherweise über dem Wahrscheinlichkeits-, dem Max-Mal-, dem logarithmischen oder dem tropischen Halbring verwendet (siehe [2, Tab. 1]), welche über die Viterbi-Approximation bzw. durch Isomorphie miteinander in Beziehung stehen [2, Bild 1].

2.2 Eigenschaften

SSTs führen eine weiche Vektorquantisierung [6] aus. Sie übersetzen Subsymbolfolgen (Merkmalvektorfolgen) $\vec{\mathbf{o}} = \vec{o}^{\,1} \ldots \vec{o}^{\,K} \in \mathcal{O}^*$ in Symbolfolgen (Zeichenketten) $\mathbf{q} = q^1 \ldots q^K \in Q^*$ derselben Länge K und ordnen dieser Übersetzung das Gewicht

$$\mathcal{Q}(\vec{\mathbf{o}}, \mathbf{q}) = \bigotimes_{k=1}^{K} q^k\big(\vec{o}^{\,k}\big) \tag{7}$$

zu. Der inverse SST $\mathcal{Q}^{-1}$ ([4], vgl. Gl. 4) übersetzt Zeichenketten $\mathbf{q}$ in Vektorfolgen $\vec{\mathbf{o}}$ der selben Länge und ordnet dieser Übersetzung das Gewicht

$$\mathcal{Q}^{-1}(\mathbf{q}, \vec{\mathbf{o}}) = \mathcal{Q}(\vec{\mathbf{o}}, \mathbf{q}) \tag{8}$$

zu.

SSTs sind offensichtlich nicht determiniert. Im Gegenteil wird jede beliebige Vektorfolge $\vec{\mathbf{o}}$ von *jedem* durchgehenden Weg gleicher Länge mit dem Gewicht nach Gleichung (7) akzeptiert. Die Projektion eines SST ist also ein *Subsymbol-Akzeptor* (SSA, vgl. Gl. 3) welcher Vektorfolgen $\vec{\mathbf{o}}$ das Gewicht

$$\mathcal{Q}(\vec{\mathbf{o}}) = \bigoplus_{\mathbf{q} \in Q^K} \mathcal{Q}(\vec{\mathbf{o}}, \mathbf{q}) = \bigoplus_{\mathbf{q} \in Q^K} \left[\bigotimes_{k=1}^{K} q^k\big(\vec{o}^{\,k}\big) \right] \tag{9}$$

zuordnet. Die Inversion eines SSA ergibt schließlich einen *Subsymbol-Generator* (SSG, vgl. Gl. 4) welcher entsprechend die Vektorfolge $\vec{\mathbf{o}}$ mit dem Gewicht

$$\mathcal{Q}^{-1}(\infty, \vec{\mathbf{o}}) = \mathcal{Q}(\vec{\mathbf{o}}) \tag{10}$$

erzeugt.

2.3 Trellis

Im Gegensatz zu herkömmlichen endlichen Automaten besitzen SSTs „zeitabhängige“ Gewichte. Dies wird deutlich, wenn man bedenkt, dass die Übergangsgewichte nach Gleichung (5) von den Eingabesubsymbolen $\vec{o}$ und damit auch von deren Position k in der eingebenen Folge $\vec{\mathbf{o}} = \vec{o}^{\,1} \ldots \vec{o}^{\,K}$ von Subsymbolen abhängen und dass jeder Übergang zu allen Zeitpunkten $1 \leq k \leq K$ benutzt werden kann. Dieser Sachverhalt wurde auch durch die zeitliche Indizierung in den Gleichungen (7) und (9) ausgedrückt.

Die Zeitabhängigkeit der Übergangsgewichte hat wesentlichen Einfluss auf die Konstruktion von Algorithmen zur Bestimmung kürzester Wege. Auf zwei dieser Algorithmen wurde in [3] eingegangen: der Viterbi-Algorithmus [3, S. 163f] (auch [2]) und die zeitvariante A*-Suche [3, S. 211ff].

Eine einfache Rückführung auf einen Automaten mit zeitinvarianten Gewichten ist durch Komposition der Subsymbolfolge

$$\vec{\mathbf{o}} = \vec{o}^{\,1}\,\vec{o}^{\,2}\,\ldots\,\vec{o}^{\,K} = \longrightarrow \bigcirc \xrightarrow{\varepsilon:\vec{o}^{\,1}\,|\,\overline{1}} \bigcirc \xrightarrow{\varepsilon:\vec{o}^{\,2}\,|\,\overline{1}} \cdots \xrightarrow{\varepsilon:\vec{o}^{\,K}\,|\,\overline{1}} \bigcirc \tag{11}$$

(dargestellt als trivialer Subsymbol-Generator) mit dem Subsymbol-Symbol-Transduktor $\mathcal{Q}$ möglich. Das Ergebnis

$$\vec{\mathbf{o}} \circ \mathcal{Q} = \longrightarrow \bigcirc \underset{\varepsilon:q_1\,|\,q_1(\vec{o}^{\,1})}{\overset{\varepsilon:q_N\,|\,q_N(\vec{o}^{\,1})}{\vdots}} \bigcirc \underset{\varepsilon:q_1\,|\,q_1(\vec{o}^{\,2})}{\overset{\varepsilon:q_N\,|\,q_N(\vec{o}^{\,2})}{\vdots}} \cdots \underset{\varepsilon:q_1\,|\,q_1(\vec{o}^{\,K})}{\overset{\varepsilon:q_N\,|\,q_N(\vec{o}^{\,K})}{\vdots}} \bigcirc \tag{12}$$

dieser Komposition heißt Trellis. Hier sind alle Übergänge jeweils nur zu genau einem Zeitpunkt k benutzbar. Die Gewichte sind nicht mehr im obigen Sinne zeitabhängig, da keine Eingabesymbole existieren.[4] Trellis-Generatoren können nicht nur durch Komposition mit SSTs, sondern auch durch Komposition mit Hidden-Markov-Transduktoren (siehe Abschnitt 3) gebildet werden. In ihnen sind Algorithmen zur Bestimmung kürzester Pfade ohne besondere Einschränkungen anwendbar. Praktisch wird man von der expliziten Berechnung der Komposition aus Aufwandsgründen allerdings Abstand nehmen und spezielle „zeitvariante“ Algorithmen (siehe oben) benutzen. Diese lassen sich jedoch immer als Kombination einer impliziten Komposition nach Gleichung (12) und eines entsprechenden „zeitinvarianten“ Algorithmus interpretieren.[5] Das gilt speziell auch für die oben genannten Beispiele (Viterbi-Algorithmus und A*-Suche).

3 Hidden-Markov-Transduktoren (HMTs)

3.1 Kontinuierliche HMTs

Mithilfe von SSTs können kontinuierliche Hidden-Markov-Modelle (CD-HMM, [7]) wie folgt konstruiert werden:

Definition 4 *Die Komposition $\mathcal{H} = \mathcal{Q} \circ \mathcal{A}$ eines Subsymbol-Symbol-Transduktors $\mathcal{Q} = (z_0, \mathcal{O}, Q)$ mit einem endlichen Transduktor $\mathcal{A} = (Z, I, F, Q, Y, w)$ mit endlichem Ausgabealphabet Y, heißt (kontinuierlicher) Hidden-Markov-Transduktor (HMT), wenn $\mathcal{A}$ die stochastische Randbedingung $\bigoplus_{U \in \mathcal{P}(I,F)} w(U) = \overline{1}$ erfüllt.*

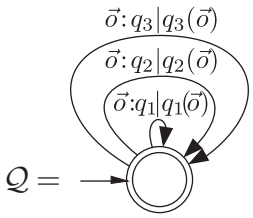
Die stochastische Randbedingung ist hier erforderlich, damit eine Parameterschätzung mittels des EM-Algorithmus sinnvoll definiert ist. Bei Verwendung von Viterbi-Halbringen zur effizienten Auswertung ist die stochastische Randbedingung in der Regel verletzt. Dies ist hinnehmbar, wenn man die Rechnungen im Viterbi-Halbring als Approximation von Rechnungen im logarithmischen Halbring betrachtet, was in der Praxis meist ziemlich gut gerechtfertigt ist [3].

[4] Anschaulich können alle Gewichte also bereits *vor* der Abarbeitung des Automaten berechnet und an die Übergänge geschrieben werden.

[5] Damit ist nicht gesagt, dass die entsprechende Implementierung auch effizient ist.

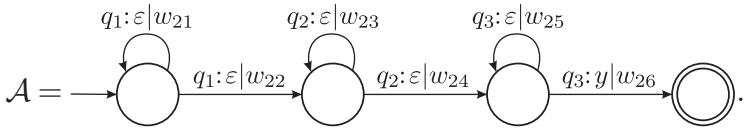
Bei so definierten Hidden-Markov-Transduktoren stellt $\mathcal{A}$ den „versteckten“ Automaten und $\mathcal{Q}$ die kontinuierlichen Ausgabefunktionen dar.

Betrachten wir als Beispiel einen Subsymbol-Symbol-Transduktor



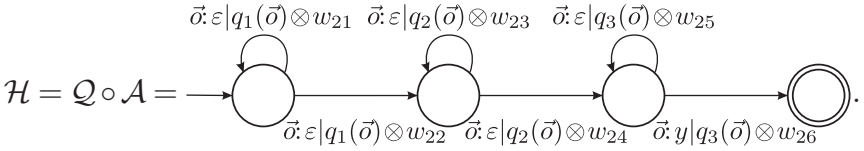
(13)

sowie einen „versteckten“ Automaten



(14)

Die Komposition von (13) und (14) lautet dann:



(15)

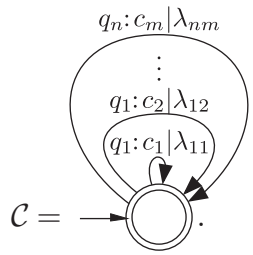
Wir sehen, dass sich die für HMMs typischen Produkte $q(\vec{o})\otimes w$ aus Ausgabe- und Übergangswahrscheinlichkeit auf natürliche Weise aus der Komposition ergeben.

3.2 Semikontinuierliche HMTs

Zur Konstruktion semikontinuierlicher Hidden-Markov-Modelle führen wir Mischungstransduktoren ein.

Definition 5 *Ein Mischungstransduktor $\mathcal{C}$ über einem Halbring $(\mathbb{K},\oplus,\otimes,\overline{0},\overline{1})$ ist ein 4-Tupel $\mathcal{C}=(z_0,Q,C,\lambda)$ bestehend aus genau einem Zustand z_0, einer endlichen Menge $Q=\{q_1,\ldots,q_N\}$ von Symbolen, einer endlichen Menge $C=\{c_1,\ldots,c_M\}$ von Mischungen sowie einer Verhaltensfunktion $\lambda:\{z_0\}\times Q\times C\times\{z_0\}\to\mathbb{K}$, welche für alle $c_m\in C$ die stochastische Randbedingung $\bigoplus_n\lambda(z_0,q_n,c_m,z_0)=\overline{1}$ erfüllt.*

Der Mischungstransduktor ist ein gewichteter endlicher Automat mit endlichen Ein- und Ausgabealphabeten und entspricht somit Definition 2. Seine grafische Darstellung ist:



(16)

Ein semikontinuierliches Hidden-Markov-Modell kann durch die Komposition eines SSTs $\mathcal{Q}$ nach Definition 3, eines Mischungstransduktors $\mathcal{C}$ nach Definition 5 und eines „versteckten“ Automaten $\mathcal{A}$, dessen Eingabealphabet die Menge C der Mischungen ist, dargestellt werden:

$$\mathcal{H}=\mathcal{Q}\circ\mathcal{C}\circ\mathcal{A}=\mathcal{Q}\circ\mathcal{A}' \tag{17}$$

Falls der Automat $\mathcal{A}$ die in Definition 4 angeführte stochastische Randbedingung erfüllt, gilt das aufgrund der in Definition 5 geforderten stochastischen Randbedingung auch für die Komposition $\mathcal{A}' = \mathcal{C} \circ \mathcal{A}$. Somit sind semikontinuierliche HMTs nach Gleichung (17) ein Spezialfall kontinuierlicher HMTs. Sie bedürfen in keiner Hinsicht, insbesondere auch nicht für die Suche und die Parameterschätzung, einer mathematischen Sonderbehandlung. Weiterführende Betrachtungen zur Darstellung von Mischverteilungsdichten durch endliche Automaten können in [8] nachgelesen werden.

3.3 Die Rabinerschen HMM-Probleme

Wir haben in [2] FST-Formulierungen für den Forward-Backward- sowie den Viterbi-Algorithmus vorgestellt und ausgeführt, dass beide Algorithmen sich nur durch den verwendeten Gewichtshalbring unteschеiden. Abgesehen vom Backtracking sind die Berechnung des Emissionsgewichts (HMM-Problem 1 [9]) und des optimalen versteckten Weges (HMM-Problem 2) also gleich.

Der in [2, Abschnitt 3] angegebenen Algorithmus zur dynamischen Programmierung (also Forward-Backward- und Viterbi-Algorithmus) ist für erweiterte wFST nach Definition 2 anwendbar.

Das dritte Rabinersche HMM-Problem [9] bezieht sich bekanntlich auf die Parameterschätzung. Als „Parameter“ gelten dabei die Übergangswahrscheinlichkeiten des versteckten Automaten sowie die Parameter der Ausgabeverteilungsdichten (üblicherweise Normalverteilungsdichten oder Gaußsche Mischverteilungsdichten).

Der von uns in [2, Abschnitt 3] vorgestellte Algorithmus für die Baum-Welch- und Viterbi-Parameterschätzung für Hidden-Markov-Modelle ist für SSTs nach Definition 3 sowie für HMTs nach Definition 4 anwendbar, falls die Funktionen $q_n : \mathcal{O} \to \mathbb{K}$ Normalverteilungsdichten über einem d-dimensionalen reellen Vektorraum $\mathcal{O} = \mathbb{R}^d$ sind. Wie in Abschnitt 3.2 und ebenfalls bereits in [2] ausgeführt, muss in der FST-Formulierung lediglich der Fall einfacher Normalverteilungsdichten berücksichtigt werden. Gaußsche Mischverteilungsdichten werden durch Komposition eines Mischungstransduktor nach Definition 5 mit dem eigentlichen versteckten Automaten auf einfache Normalverteilungsdichten zurückgeführt. Allerdings werden im letzteren Fall nur Produkte $\lambda \otimes w$ aus Mischungs- und Übergangsgewichten geschätzt. Eine getrennte Schätzung ist nicht möglich.

4 Zusammenfassung und Ausblick

Wir haben eine erweiterte Definition von gewichteten endlichen Automaten (wFST) angegeben, welche unendliche Ein- und Ausgabealphabete ermöglicht. Basierend auf dieser Erweiterung haben wir Subsymbol-Symbol-Transduktoren (SST) entwickelt, welche – aus Sicht der akustischen Mustererkennung – in der Lage sind, zwischen Merkmalvektorfolgen und akustischen Elementarsymbolen („HMM-Zuständen“) zu übersetzen. Dazu wird zusätzlich zu einem unendlichen Eingabealphabet auch die Abhängigkeit der Gewichte vom aktuellen Eingabesymbol und damit eine Zeitabhängigkeit der Gewichte zugelassen. Wir haben weiterhin gezeigt, wie man Hidden-Markov-Modelle als Komposition aus einem SST und einem herkömmlichen wFST darstellen kann.

Der in dieser Arbeit dargestellte Formalismus umfasst Transduktoren, welche Vektorfolgen in Zeichenketten, Zeichenketten in Vektorfolgen, aber auch Zeichenketten in Zeichenketten und

Vektorfolgen in Vektorfolgen übersetzen können. Da zeitdiskrete Signale als eindimensionale Vektorfolgen interpretiert werden können, schlagen SSTs eine direkte Brücke bis zur zeitdiskreten Signalverarbeitung. Mit einigen zusätzlichen Erweiterungen – beispielsweise Abhängigkeit der Ausgabe(sub)symbole von den Eingabe(sub)symbolen an jedem Übergang – können sie im Prinzip auch Digitalfilter, Merkmalextraktion, Merkmaltransformation usw. beschreiben. Inwieweit dies auch tatsächlich sinnvoll ist, bleibt allerdings noch zu klären.

Literatur

[1] TSCHÖPE, C.: *Akustische zerstörungsfreie Prüfung mit Hidden-Markov-Modellen*. Dissertationsschrift, Technische Universität Dresden, 2012. Bd. 60 d. Reihe *Studientexte zur Sprachkommunikation*, TUDpress, Dresden.

[2] TSCHÖPE, C. und WOLFF, M.: *Zur Formulierung von Hidden-Markov-Modellen als endliche Transduktoren*. In: WOLFF, M. (Herausgeber): *Elektronische Sprachsignalverarbeitung 2012 (ESSV 2012)*, Band 64 der Reihe *Studientexte zur Sprachkommunikation*, Seiten 120–128. TUDpress, Dresden, 2012.

[3] WOLFF, M.: *Akustische Mustererkennung*. Habilitationsschrift, Technische Universität Dresden, 2011. Bd. 57 d. Reihe *Studientexte zur Sprachkommunikation*, TUDpress, Dresden.

[4] MOHRI, M.: *Weighted Automata Algorithms*. In: DROSTE, M., W. KUICH und H. VOGLER (Herausgeber): *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science. An EATCS Series, Seiten 213–254. Springer Berlin Heidelberg, 2009.

[5] MOHRI, M.: *Minimization algorithms for sequential transducers*. Theoretical Computer Science, 234:177–201, 2000.

[6] HUANG, X., ACERO, A. und HON, H.-W.: *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. In: *Englewood Cliffs, New Jersey: Prentice Hall*, 2001.

[7] HUANG, X., AKIRI, Y. und JACK, M.A.: *Hidden Markov Models for Speech Recognition*. In: *Edinburgh University Press New Jersey*, 1990.

[8] DUCKHORN, F., WOLFF, M. und HOFFMANN, R.: *Realisierung von Mischverteilungsdichten durch gewichtete Automaten (Finite-State-Transducer)*. In: *9. ITG-Fachtagung Sprachkommunikation*, 2010.

[9] RABINER, L. R.: *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2):257–286, Febr. 1989.