3.2 Formale Grundlagen des EM-Algorithmus

Der *Erwartungswert-Maximierungs-Algorithmus* (EM-Algorithmus) wurde von Dempster et al. (1977) eingeführt und benannt. In diesem Artikel finden sich einige grundlegende Eigenschaften des EM-Algorithmus, sowie mehrere (nicht-linguistische) Beispiele und Anwendungen. Da die Idee hinter dem EM-Algorithmus so intuitiv und allgemein verständlich ist, wurde der EM-Algorithmus allerdings von vielen Autoren schon vor 1977 in einer Vielzahl von Anwendungen benutzt. Speziell für die Linguistik ist interessant, dass Baker (1979) den Inside-Outside-Algorithmus für kontextfreie Grammatiken in Chomsky-Normalform einführte und ihn mit dem Inside-Outside-Algorithmus für Hidden-Markov--Modelle verglich, welcher seinerseits von Baum (1972) in seiner endgültigen Form, als dynamische Programmiervariante, vorgestellt wurde. In Abschnitt 3.8 wird gezeigt werden, dass der Inside-Outside-Algorithmus eine Instanz des EM-Algorithmus für kontextfreie Grammatiken ist. Obwohl Baker seine Arbeit mit einer linguistischen Anwendung motivierte (Modellierung von Fehlern in der Spracherkennung), wird der EM-Algorithmus erst in neuerer Zeit in der Computer-Linguistik (insbesondere für Parsing und lexikalische Semantik) verstärkt angewendet.

In dieser Arbeit wird der EM-Algorithmus als stochastische Grundlage der *automatischen Lernverfahren* benutzt, die in Kapitel 4 und 5 vorgestellt werden. Die Ziele dieser Lernverfahren sind:

- semantische Klassifizierung von Verb-Argument-Daten,
- Induktion semantisch annotierter Lexika,
- Identifikation von Verb-Nomen-Kollokationen,
- Auflösung lexikalischer Ambiguitäten,
- phonologische Klassifizierung von Silben,
- Training von probabilistischen lexikalisierten kontext-freien Grammatiken,
- Training von stochastischen unifikationsbasierten Grammatiken.

Diese Lernverfahren zeichnen sich dadurch aus, dass es stochastische Verfahren sind, deren stochastischen Parameter in einer iterativen Trainingsphase vermittels des Maximum-Likelihood-Kriteriums optimal bestimmt werden. Ihr charakterisierendes Merkmal, was sie von den meisten anderen Lernverfahren unterscheidet, ist, dass dank der Benutzung des

EM-Algorithmuss in der Lernphase ein Trainingskorpus benutzt werden darf, der unannotiert (d.h. ambig) ist[3]. Das bedeutet:

- Zur semantischen Klassifizierung von Verb-Argument-Daten ist kein semantisch annotierter Trainingskorpus, wie SemCor, nötig,
- Die Induktion semantisch annotierter Lexika wird ohne Benutzung semantisch fixierter Taxonomien, wie WordNet oder GermaNet geschehen,
- Die Identifikation von Verb-Nomen-Kolloktionen ist auf der Grundlage automatisch klassifizierter Verb-Nomen-Paare möglich,
- Die Auflösung lexikalischer Ambiguitäten gelingt ohne Zugriff auf WordNet- bzw. Hector- Annotationen,
- Eine phonologische Klassifizierung von Silben ist auf der Grundlage grosser transkribierter Texte möglich,
- Ein unbeaufsichtigtes Training lexikalisierter probabilistischer kontext-freier Grammatiken ist ohne Baumbanken möglich, wie sie beispielsweise für Englisch mit der Penn-Treebank, oder dem Negra-Korpus für Deutsch, in relativ kleinem Umfang zur Verfügung stehen,
- Ein Training lexikalisierter stochastischer unifikationsbasierter Grammatiken ist ohne Parse-Banken, die nur in einem kleinen Umfang vorhanden sind, z.B. der Homecenter-Korpus oder Verbmobil-Baumbank, nicht nur möglich, sondern zeigt auch sehr viel bessere Ergebnisse.

Neben den schon genannten Artikeln von Baum und Eagon (1966), Baum und Sell (1968), Baum et al. (1970), Baum (1972), Dempster et al. (1977), Baker (1979) ist das Buch von McLachlan und Krishnan (1997) eine empfehlenswerte Literaturquelle zum EM-Algorithmus. Leider erschliessen sich diese Arbeiten ohne mathematische Kenntnisse nur schwer.

Formale Definition des EM-Algorithmus

Rein formal kann der EM-Algorithmus als ein 8-Tupel:

$$< \mathcal{Y},\ \tilde{p}(.),\ \mathcal{X},\ X(.),\ \Omega,\ \Omega_0,\ \{p_\Theta(.)|\Theta \in \Omega\},\ \mathcal{M}(.) >$$

[3]Ein Standardwerk für maschinelle Lernverfahren ist Mitchell (1997). Der EM-Algorithmus wird allerdings erst in Kapitel 6.12 behandelt, während einige seiner einfachen Spezialfälle, z.B. die auf *Entscheidungsbäumen* basierenden Lernverfahren, bevorzugt präsentiert werden. Ein weiteres interessantes Buch ist Frey (1998), in dem "graphische Modelle" präsentiert werden, die eine Reihe von stochastischen Lernverfahren verallgemeinern.

mit den folgenden Elementen definiert werden:

- $\mathcal{Y}$, die Menge der *unvollständigen Datentypen*,
- $\tilde{p}(.)$, die *empirische Wahrscheinlichkeitsverteilung der unvollständigen Datentypen*,
- $\mathcal{X}$, die Menge der *vollständigen Datentypen*,
- $X(.)$, die *symbolische Analyse-Komponente*,
- Ω, der *Parameterraum*,
- Ω_0, die *Menge der Startparameter*,
- $\{p_\Theta(.) \mid \Theta \in \Omega\}$, die Menge der *parametrisierten vollständigen Wahrscheinlichkeitsverteilungen*,
- $\mathcal{M}(.)$, die *EM-Parameter-Abbildung*.

Für die Anwendung des EM-Algorithmus müssen diese acht Elemente spezifiziert, d.h. berechnet oder mit Werten initialisiert werden. Im folgenden werden sie definiert.

- Die Menge $\mathcal{Y}$ der *unvollständigen Datentypen* ist eine diskrete (endliche oder abzählbar unendliche) Menge von linguistischen Datentypen. Dem Sprachgebrauch liegt die Vorstellung zugrunde, dass diese Datentypen zusätzliche strukturelle Eigenschaften besitzen, die versteckt sind, aber mit geeigneten Analysemethoden aufgedeckt werden können. Daher ist es angemessen, sich unvollständige Datentypen als im Prinzip linguistisch reich strukturierbare Daten vorzustellen, von denen aber lediglich die "Oberfläche" sichtbar ist. Ein geeignetes Beispiel unvollständiger Datentypen sind die natürlichsprachigen Sätze (Wortfolgen). Im Rahmen des EM-Algorithmus werden sie als unvollständig angesehen, wenn sie (was in freiem Text, nicht aber in einer Baumbank, der Fall ist) ohne die zusätzliche Angabe ihrer grammatischen Analysen vorliegen.

- Die *empirische Wahrscheinlichkeitsverteilung* $\tilde{p}(.)$ *der unvollständigen Datentypen* ist wie in Abschnitt 2.1 definiert. Jedem unvollständigen Datentyp $y \in \mathcal{Y}$ wird als Wahrscheinlichkeit die relative Häufigkeit $f(y)$ seines Vorkommens in einem gegebenen Korpus zugewiesen:

  $$\tilde{p} : \mathcal{Y} \to [0, 1], \quad y \mapsto |f|^{-1} \cdot f(y) \ .$$

  Hierbei ist $|f| = \sum_{y \in \mathcal{Y}} f(y)$ die Grösse des Korpus, ausgedrückt durch die Anzahl aller seiner Datentoken. Die Korpusgrösse muss daher als ein Parameter des EM-Algorithmus angesehen werden, der möglicherweise (vermittels der empirischen Wahrscheinlichkeitsverteilung) die Resultate des EM-Algorithmus beeinflussen wird.

- Die Menge $\mathcal{X}$ der *vollständigen Datentypen* ist ebenfalls eine diskrete Menge von linguistischen Datentypen. Im Vergleich mit den unvollständigen Datentypen sollte man sich die vollständigen Datentypen als linguistisch reich strukturierte Datentypen vorstellen, die keinerlei relevante Information vermissen lassen und die Informationen der unvollständigen Datentypen ergänzen. Ein geeignetes Beispiel vollständiger Datentypen sind die linguistischen Analysen von natürlichsprachigen Sätzen, die als Datentoken etwa in Baumbanken vorliegen. Im Rahmen des EM-Algorithmus können diese linguistische Analysen (Bäume) als vollständig angesehen werden, weil sie Sätze (Wortfolgen) um strukturelle, linguistische Informationen anreichern. Natürlich ist es in diesem Zusammenhang völlig unerheblich, ob man für die Beschreibung der linguistischen Informationen einen kontextfreien oder unifikationsbasierten Formalismus wählt.

- Die *symbolische Analyse-Komponente* $X(.)$ hat die Fähigkeit, jeden unvollständigen Datentyp analysieren zu können, d.h. dessen versteckte, strukturelle Eigenschaften mit geeigneten Methoden aufzudecken. Da ein unvollständiger Datentyp $y \in \mathcal{Y}$ mehr als eine einzige Analyse besitzen darf, ist es zweckmässig, die Menge $X(y)$ aller seiner Analysen als eine Teilmenge der vollständigen Datentypen zu formalisieren. Folglich ist die symbolische Analyse-Komponente $X(.)$ eine Abbildung von der Menge der unvollständigen Datentypen auf die Potenzmenge der vollständigen Datentypen:

  $$X : \mathcal{Y} \to \mathcal{P}(\mathcal{X}), \quad y \mapsto X(y) \ .$$

  Das typische Beispiel einer symbolischen Analyse-Komponente ist der symbolische Parser, da dieser einem Satz (d.h. einem unvollständigen Datentyp) unter Benutzung einer geeigneten Grammatik eine Menge von Satzanalysen (d.h. eine Teilmenge der vollständigen Datentypen) zuweisen wird. Weitere Beispiele symbolischer Analyse--Komponenten sind semantische Tagger, aber natürlich auch gewöhnliche Part-Of-Speech-Tagger. Die natürlichsprachigen Wörter können in diesem Zusammenhang als unvollständige, und die Wort-Tag-Paare als vollständige Datentypen angesehen werden. Ein symbolischer Tagger bildet dann jedes Wort auf eine Anzahl von möglichen Wort-Tag-Paaren ab.

  Es ist überaus wichtig, dass die symbolische Analyse-Komponente im Szenario des EM-Algorithmus die folgenden zwei Eigenschaften besitzt: (i) kein vollständiger Datentyp darf die Analyse von zwei verschiedenen unvollständigen Datentypen sein, (ii) jeder vollständige Datentyp ist die Analyse von wenigstens einem unvollständigen Datentyp. Diese Eigenschaften könnten mit *Disjunkheit* und *Vollständigkeit* umschrieben werden. Formal gilt:

$$\begin{array}{lll} (\textit{Disjunkheit}) & X(y) \cap X(y') = \emptyset & \text{für } y \neq y' \ , \\ (\textit{Vollständigkeit}) & \bigcup_{y \in \mathcal{Y}} X(y) \supseteq \mathcal{X} & . \end{array}$$

Natürlich führen Disjunkheit und Vollständigkeit dazu, dass die Menge der vollständigen Datentypen in die Menge der Analysen der unvollständigen Datentypen *partitioniert* wird:

$$\sum_{y \in \mathcal{Y}} X(y) = \mathcal{X} \ .$$

Sobald die für den EM-Algorithmus relevanten Wahrscheinlichkeitsverteilungen eingeführt werden, wird klar werden, warum für die Analysen Disjunkheit und Vollständigkeit gefordert wird. An dieser Stelle soll zumindest darauf hingewiesen werden, dass die bisher eingeführten zwei Beispiele von symbolischen Analyse-Komponente diese beiden Eigenschaften besitzen. Ein symbolischer Syntax-Parser wird keinen zwei verschiedenen Sätzen dieselbe Analyse zuweisen, auch wird er jedem grammatischen Satz analysieren können (wenn schon nicht in der Praxis, dann zumindest in der Theorie). Ein (semantischer oder Part-Of-Speech) Tagger wird zwei verschiedenen Worten zwar möglicherweise dasselbe Tag zuweisen, aber nicht dasselbe Wort-Tag-Paar. Aus diesem Grund ist es sehr wichtig, dass nicht die Tags, sondern die Wort-Tag-Paare als vollständige Daten benutzt werden[4]. Ferner wird ein EM-basierter Tagger kein Wort unanalysiert lassen.

- Der *Parameterraum* Ω ist im allgemeinen eine kontinuierliche Menge von reellwertigen, mehr-dimensionalen Veränderlichen, die einer Reihe von Nebenbedingungen unterworfen sein können. Parameter $\Theta \in \Omega$ dienen in der Regel einem einfachen Zweck: sie sollen relevante Entitäten, in dieser Arbeit sind dies Wahrscheinlichkeitsverteilungen, vermittels simpler Indizierung benennen und zu einer weiterführenden Betrachtung freigeben. Ein Parameterraum muss im Rahmen des EM-Algorithmus eine Reihe mathematischer Eigenschaften aufweisen, deren Zusammenspiel aber zum grössten Teil noch unerforscht ist. In dieser Arbeit wird versucht, an geeigneter Stelle auf einige dieser Eigenschaften hinzuweisen.

- Die *Menge der Startparameter* Ω_0 ist eine endliche, in der Regel relativ kleine Teilmenge des Parameterraumes:

$$\Omega_0 \subseteq \Omega \ .$$

Im Gegensatz zu vielen Darstellungen des EM-Algorithmus, die mit einem einzigen Startparameter arbeiten, wird in dieser Arbeit eine leicht modifizierte Version des

[4]Die Tags repräsentieren in diesem Fall in sehr anschaulicher Weise die Information, die aus einem unvollständigen Datentyp (Wort) einen vollständigen Datentyp (Wort-Tag-Paar) macht.

EM-Algorithmus vorgestellt, die es nicht nur erlaubt, sondern sogar explizit fördert, mit einer beliebigen Anzahl von Startparametern zu arbeiten. Es wird vermutet, dass die hier vorgestellte Version schon für $|\Omega_0| = 2$ in den meisten Fällen wesentlich bessere Ergebnisse liefern wird, als für $|\Omega_0| = 1$ (Standard-Version in der Literatur). Das hat den folgenden Grund: Der EM-Algorithmus ist ein iterativer Algorithmus, von dem theoretisch gezeigt werden kann, dass er zu jedem fixen Startparameter $\Theta_0 \in \Omega$ eine Folge von (in einem bestimmten Sinn) immer besser werdenden Parametern produziert. Es ist sehr gut bekannt, dass verschiedene Startparameter in der Praxis oft zu qualitativ sehr verschiedenen Ergebnissen führen können (Müller (2000), Müller (2001b)). Leider weiss man aber nur in einigen wenigen Spezialfällen (Riezler et al. 2000), welche Wahl des Startparameters Θ_0 (zumindest theoretisch) optimal sein könnte. Konzeptionell ist es daher sinnvoll, den EM-Algorithmus nicht nur mit einem einzigen Startparameter durchzuführen, sondern mit mehreren Startparametern $\Theta_0 \in \Omega_0$.

- Die Menge $\{p_\Theta(.) \mid \Theta \in \Omega\}$ fasst die *parametrisierten Wahrscheinlichkeitsverteilungen $p_\Theta(.)$ über der Menge $\mathcal{X}$ der vollständigen Datentypen*, auch *parametrisierte vollständige Wahrscheinlichkeitsverteilungen* genannt, für alle Parameter $\Theta \in \Omega$ des Parameterraumes zusammmen. Es gilt also:

$$p_\Theta : \ \mathcal{X} \to [0,1], \quad x \mapsto p_\Theta(x) \ ,$$

mit:

$$p_\Theta(\mathcal{X}) = \sum_{x \in \mathcal{X}} p_\Theta(x) = 1 \ .$$

Im Rahmen des EM-Algorithmus sind die parametrisierten vollständigen Wahrscheinlichkeitsverteilungen diejenigen Objekte, auf die sich das ganze weitere Interesse bündelt. Dies kann in zwei Schritten eingesehen werden: (i) die parametrisierten vollständigen Wahrscheinlichkeitsverteilungen können (siehe unten) zu parametrisierten unvollständigen Wahrscheinlichkeitsverteilungen verallgemeinert werden, (ii) auf der Grundlage der unvollständigen Wahrscheinlichkeitsverteilungen und dem gegebenen Korpus $\tilde{p}(.)$ von unvollständigen Daten könnte mit der Maximum--Likelihood-Estimierung ein optimaler Parameter bestimmt werden. Die Maximum--Likelihood-Estimierung zur Bestimmung eines Parameters, welcher dem gegebenen Korpus $\tilde{p}(.)$ eine möglichst hohe Wahrscheinlichkeit zuweist, wurde bereits in Abschnitt 2.4 motiviert. An dieser Stelle soll daher auf Schritt (i) eingegangen werden. Es ist beachtenswert, dass die parametriserten unvollständigen Wahrscheinlichkeitsverteilungen in sehr natürlicher Weise zu Wahrscheinlichkeitsverteilungen über der

Menge der unvollständigen Daten führen, die im folgenden *parametrisierte unvollständige Wahrscheinlichkeitsverteilungen* genannt werden:

$$p_\Theta : \mathcal{Y} \to [0,1], \quad y \mapsto p_\Theta(X(y)) \ .$$

Wie bereits in dieser Definition angegeben wurde, ist es hierzu lediglich nötig, jeden unvollständigen Datentyp $y \in \mathcal{Y}$ mit der Menge seiner Analysen $X(y)$ zu identifizieren, d.h. die Wahrscheinlichkeit dieses unvollständigen Datentyps als die Wahrscheinlichkeit des Ereignisses $X(y)$ zu definieren:

$$p_\Theta(y) := p_\Theta(X(y)) \ .$$

Gemäss Abschnitt 2.2 ist dies die Wahrscheinlichkeit, dass mindestens eine der Analysen $x \in X(y)$ eintritt:

$$p_\Theta(X(y)) = \sum_{x \in X(y)} p_\Theta(x) \ .$$

Es kann nun leicht überprüft werden, dass aus dieser Definition tatsächlich eine Wahrscheinlichkeitsverteilung auf den unvollständigen Datentypen resultiert. Hierzu benutzt man als Schlüsselschritt, dass die Menge der vollständigen Datentypen in die Menge der Analysen der unvollständigen Datentypen partitioniert wird (siehe oben):

$$p_\Theta(\mathcal{Y}) = \sum_{y \in \mathcal{Y}} p_\Theta(y) = \sum_{y \in \mathcal{Y}} p_\Theta(X(y)) = p_\Theta\left(\sum_{y \in \mathcal{Y}} X(y)\right) = p_\Theta(\mathcal{X}) = 1 \ .$$

- Aus prozeduraler Sicht besteht der EM-Algorithmus in einer iterativen Anwendung der *EM-Parameter-Abbildung* $\mathcal{M}(.)$ auf die Startwerte des EM-Algorithmus. Somit könnte die EM-Parameter-Abbildung als der eigentliche Kern des EM-Algorithmus bezeichnet werden. Formal ist sie eine Abbildung von dem Parameterraum auf sich selbst und legt zu jedem Parameter $\Phi \in \Omega$ in zwei Schritten, (i) dem *Erwartungswert-Schritt (E-Schritt)* und (ii) dem *Maximierungs-Schritt (M-Schritt)* einen *re-estimierten Parameter* $\hat{\Theta} \in \Omega$ fest. Offensichtlich waren beide Schritte für den EM-Algorithmus namensgebend. Von der folgenden Definition der EM-Parameter-Abbildung lässt sich der M-Schritt bereits deutlich ablesen, der in der Maximierung der sogenannten EM-Hilfsfunktion besteht:

$$\mathcal{M} : \ \Omega \to \Omega, \quad \Phi \mapsto \hat{\Theta} = \mathrm{argmax}_{\Theta \in \Omega} \ \ Q_\Phi(\Theta) \ .$$

Der dem M-Schritt vorangehende E-Schritt besteht darin, die *EM-Hilfsfunktion* $Q_\Phi(.)$ zu berechnen. Die EM-Hilfsfunktion ordnet jedem Parameter einen Wert zu,

von dem gezeigt werden wird, dass er (i) ein doppelter Erwartungswert ist, und (ii) in enger Verwandschaft mit der Log-Likelihood steht:

$$Q_\Phi : \ \Omega \to \mathbb{R}, \quad \Theta \mapsto \sum_{y \in \mathcal{Y}} \tilde{p}(y) \sum_{x \in X(y)} p_\Phi(x|y) \cdot \log \ p_\Theta(x) \ .$$

Die in dieser Definition vorkommende *bedingte Wahrscheinlichkeit* $p_\Phi(x|y)$ *einer Analyse* $x \in X(y)$ *gegeben deren unvollständigen Datentyps* $y \in \mathcal{Y}$ kann aus der gewöhnlichen Definition der bedingten Wahrscheinlichkeit abgeleitet werden. Hierzu ist es wieder nur nötig, wie bei der Definition der unvollständigen Wahrscheinlichkeitsverteilungen, einen unvollständigen Datentyp y mit seinen Analysen $X(y)$ zu identifizieren:

$$p_\Phi(x|y) = p_\Phi(x|X(y)) \ .$$

Am Ende von Abschnitt 2.2 wurden bedingte Wahrscheinlichkeiten formal eingeführt. Unter Benutzung dieser Definition folgt:

$$p_\Phi(x|X(y)) = \frac{p_\Phi(x)}{p_\Phi(X(y))} = \frac{p_\Phi(x)}{\sum_{x \in X(y)} p_\Phi(x)} \ .$$

Der EM-Algorithmus ist mit den vorangehenden Ausführungen formal völlig ausreichend erklärt worden. Da es aber für die Namensgebung des EM-Algorithmus massgebend ist, soll noch darauf hingewiesen werden, dass die Werte $Q_\Phi(\Theta)$ der EM-Hilfsfunktion als (doppelte) Erwartungswerte aufgefasst werden können. Um die Definition der Erwartungswerte anwenden zu können, ist es nötig, dass die bedingten Wahrscheinlichkeiten $p_\Phi(x|y)$ für jeden unvollständigen Datentyp $y \in \mathcal{Y}$ eine Wahrscheinlichkeitsverteilung $p_\Phi(.|y)$ auf der Menge seiner Analysen $x \in X(y)$ bilden:

$$p_\Phi(.|y) : \ X(y) \to [0,1], \quad x \mapsto \frac{p_\Phi(x)}{\sum_{x \in X(y)} p_\Phi(x)} \ ,$$

mit:

$$\sum_{x \in X(y)} p_\Phi(x|y) = 1 \ .$$

Die letzte Gleichung ist sehr einfach nachzuprüfen und erlaubt es, die EM-Hilfsfunktion als eine doppelte Erwartung zu identifizieren. Unter Benutzung der Definition der Erwartungswerte (siehe Abschnitt 2.3) gilt:

$$\begin{aligned} Q_\Phi(\Theta) &= \sum_{y \in \mathcal{Y}} \tilde{p}(y) \sum_{x \in X(y)} p_\Phi(x|y) \cdot \log \ p_\Theta(x) \\ &= \sum_{y \in \mathcal{Y}} \tilde{p}(y) \cdot p_\Phi(.|y) \, [\ \log \ p_\Theta \ ] \\ &= \tilde{p} \, [\ p_\Phi(.|.) \, [\ \log \ p_\Theta \ ] \ ] \ . \end{aligned}$$

Zusammenfassend folgt, dass ein re-estimierter Parameter im EM-Algorithmus (iterativ) wie folgt berechnet wird:

$$\begin{aligned}\hat{\Theta} &= \text{argmax}_{\Theta\in\Omega}\ \ Q_\Phi(\Theta) \\ &= \text{argmax}_{\Theta\in\Omega}\ \ \tilde{p}\,[\ p_\Phi(.|.)\,[\ \log\ p_\Theta\ ]\ ]\ .\end{aligned}$$

Prozedurale Sicht auf den EM-Algorithmus

Abbildung 3.7 zeigt den EM-Algorithmus aus prozeduraler Sicht. Seine Eingabe besteht aus einem Trainingskorpus $\tilde{p}(.)$ von unvollständigen Datentypen, einer symbolischen Analyse-Komponente $X(.)$, die zu jedem unvollständigen Datentyp $y \in \mathcal{Y}$ die Menge $X(y)$ seiner Analysen bereitstellt. Als weitere Eingabe liegen parametrisierte vollständige Wahrscheinlichkeitsverteilungen $p_\Theta(.)$ vor, für die (in einem gewissen Sinn) optimale Parameter zu bestimmen sind. Ferner gehört zu dieser Version des EM-Algorithmus im Gegensatz zur Standardversion nicht nur ein einziger, sondern eine Menge $\Omega_0 \subseteq \Omega$ von Startparametern, sowie anstatt eines Abruchkriteriums eine (im allgemeinen relativ hohe) Anzahl *numberOfIterations* ≥ 1 von festgelegten Iterationen. Die Bevorzugung vieler Startparameter gegenüber einem einzigen Startparameter wurde schon motiviert. Die Benutzung einer hohen Anzahl von Iterationen ist gegenüber einem Log-Likelihood-basierten Abruchkriterium zu bevorzugen, weil ein (für eine konkrete linguistische Anwendung) optimaler Parameter nur mit einer zusätzlichen linguistischen Evaluierung bestimmt werden kann, und diese linguistische Evaluierung erfahrungsgemäss selten mit einem Log-Likelihood-basierten Abruchkriterium korreliert.

Die Ausgabe des EM-Algorithmus besteht aus einer Folge von re-estimiertemn Parametern

$$\Theta_1,\ \Theta_2,\ \Theta_3,\ \ldots \in \Omega\ ,$$

d.h. aus Parametern, die durch eine iterierte Anwendung der EM-Parameter-Abbildung $\mathcal{M}(.)$ enstehen:

$$\Theta_i = \mathcal{M}(\Theta_{i-1}) \qquad (i = 1, 2, 3, \ldots)\ .$$

Dempster et al. (1977) zeigten, dass die zu diesen Parametern gehörige Folge der Log-Likelihood-Werte auf dem Trainingskorpus beschränkt und monoton wachsend ist:

$$L(\Theta_1) \leq L(\Theta_2) \leq L(\Theta_3) \leq \ldots \in [L(\Theta_0),\ 0]\ .$$

Während ein einfacher Satz der Analysis besagt, dass jede beschränkte und monoton wachsende Folge *konvergent* ist, d.h.

$$L(\Theta_i) \to L^* \qquad (i \to \infty)\ ,$$

Eingabe: Ein *Trainingskorpus* $\tilde{p}(.)$ *von unvollständigen Datentypen*, eine *symbolische Analyse-Komponente* $X(.)$, parametrisierte vollständige Wahrscheinlichkeitsverteilungen $p_\Theta(.)$, eine endliche Menge Ω_0 von *Startparametern*, sowie eine *Iterationsanzahl numberOfIterations* ≥ 1.

Ausgabe: Für jeden Startwert $\Theta_0 \in \Omega_0$ eine Folge von re-estimierten Parametern:

$$\Theta_1,\ \Theta_2,\ \Theta_3,\ \ldots \in \Omega\ ,$$

sodass die zugehörige Folge der Log-Likelihood-Werte auf dem Trainingskorpus beschränkt und monoton wachsend ist:

$$L(\Theta_1) \leq L(\Theta_2) \leq L(\Theta_3) \leq \ldots \in [L(\Theta_0),\ 0]\ .$$

Prozedur:

1. * Version mit Variation der Startparameter: *\
2. **for each** $\Theta_0 \in \Omega$ **do**
3. **for each** $i := 1, \ldots, \mathit{numberOfIterations}$ **do**
4. $\Phi := \Theta_{i-1}$;
5. **(E-step) compute** (as function of Θ)

$$Q_\Phi(\Theta) = \sum_{y\in\mathcal{Y}} \tilde{p}(y) \sum_{x\in X(y)} p_\Phi(x|y) \cdot \log\ p_\Theta(x);$$

6. **(M-step) compute**

$$\hat{\Theta} = \text{argmax}_{\Theta\in\Omega}\ \ Q_\Phi(\Theta);$$

7. $\Theta_i := \hat{\Theta}$;
8. **print** $\Theta_0,\ \Theta_1,\ \Theta_2,\ \Theta_3,\ \ldots$;

Abbildung 3.7: EM-Algorithmus

konnte Wu (1983) (mit zusätzlichen Voraussetzungen an den EM-Algorithmus) zwei weitere schöne Eigenschaften des EM-Algorithmus zeigen: (i) L^* ist ein lokales Maximum der Log-Likelihood und (ii) die Folge der Parameter konvergiert:

$$\Theta_i \rightarrow \Theta^* \qquad (i \rightarrow \infty) \ .$$

Beide Eigenschaften sind natürlich sehr wünschenswert, da sie es ermöglichen, den EM-Algorithmus als ein iteratives Lösungsverfahren für die in Kapitel 2 vorgestellte Maximum-Likelihood-Estimierung aufzufassen.

In der 2. Zeile der Prozedur des EM-Algorithmus wird eine Schleife über die Startparameter initialisiert, deren Schritte in der 3. bis 8. Zeile ausgeführt werden. In der 3. Zeile wird eine Schleife über die Trainingsiterationen initialisiert, deren Schritte in der 4. bis 7. Zeile ausgeführt werden. In der 4. Zeile wird der sogenannte *gegenwärtige Parameter* Φ der EM-Hilfsfunktion gesetzt, die als E-Schritt in der 5. Zeile berechnet wird. In der 6. Zeile wird der M-Schritt ausgeführt, der die EM-Hilfsfunktion maximiert und den zugehörigen sogenannten *neuen Parameter* $\hat{\Theta}$ bestimmt. Ohne eine konkrete Instantiierung des EM-Algorithmus stellen die 5. und 6. Zeile die allgemeinste Vorgehensweise für die Berechnung des neuen Parameters $\hat{\Theta}$ dar. Es wurde schon von Dempster et al. (1977) darauf hingewiesen, dass der Ausdruck "Algorithmus" für den EM-Algorithmus kritisiert werden könnte, weil die Sequenz der Berechnungsschritte, die für die Ausführung eines E- oder M-Schrittes nötig sind, nicht spezifiziert werden kann. Es ist offensichtlich, dass detaillierte Implementationen des EM-Algorithmus sich hinsichtlich Komplexität und Anwendbarkeit stark voneinander unterscheiden können. In einer konkreten Instanz des EM-Algorithmus ist es oft angebracht, beide Schritte zusammenzufassen, um eine konkrete mathematische Lösung für $\hat{\Theta}$ auszuarbeiten. In der Regel ist eine solche Lösung relativ einfach zu finden (siehe z.B. Abschnitt 3.8). Manchmal jedoch, wie z.B. bei den *log-linearen Modellen* (siehe Abschnitt 5.3), kann eine Lösung nur in einem iterativen Verfahren gefunden werden, sodass der EM-Algorithmus insgesamt zu einem sogenannten *doppelt-iterativen Algorithmus* wird (wenn die Schleife über die Startparameter nicht mitgezählt wird, wie es beim Standard-EM-Algorithmus leider üblich ist). In der 7. Zeile wird der neue neue Parameter gespeichert, um ihn für die nächste Iteration (als gegenwärtigen Parameter) benutzen zu können. In der 8. Zeile wird zu jedem Startparameter Θ_0 die Folge der aus ihm berechneten re-estimierten Parameter ausgegeben.

Bemerkung: Da alle re-estimierten Parameter vom EM-Algorithmus ausgegeben werden, ist es sowohl sinnvoll, als auch leicht möglich, eine zusätzliche systematische linguistische Evaluierung aller vom EM-Algorithmus berechneten Parameter extern vorzunehmen.