Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Sprachverarbeitung II / 9 FS 2017

Algorithmen für CDHMM – Wortmodelle

Buch: Kapitel 5.4.8 bis 5.7 und 13.4

Beat Pfister

Programm heute

Vorlesung:

- Viterbi-Training für DDHMM
- grundlegende Algorithmen für CDHMM
- Training mit mehreren Merkmalssequenzen
- Underflow
- Akustische Modelle für Wörter

Übung:

⋆ Training diskreter HMM

Training von DDHMM

Gegeben: Beobachtungssequenz $\mathbf{X} = \mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_T$

Aufgabe: Bestimmung von a_{ij} und $b_j(k)$ so, dass $P(\mathbf{X}|\lambda)$ maximal

Problem: Zustände des HMM sind nicht beobachtbar

Vorgehen: Baum-Welch-Algorithmus mit Hilfswahrscheinlichkeiten $\gamma_t(i)$ und $\xi_t(i,j)$
(dafür wird das HMM bereits gebraucht)

Training von DDHMM

Ausweg aus dem Problem, dass HMM-Zustände nicht beobachtbar:

Baum-Welch:
- Berücksichtigt alle Zustandsfolgen
- Hilfswahrscheinlichkeiten $\longrightarrow$ Erwartungswerte

>>>

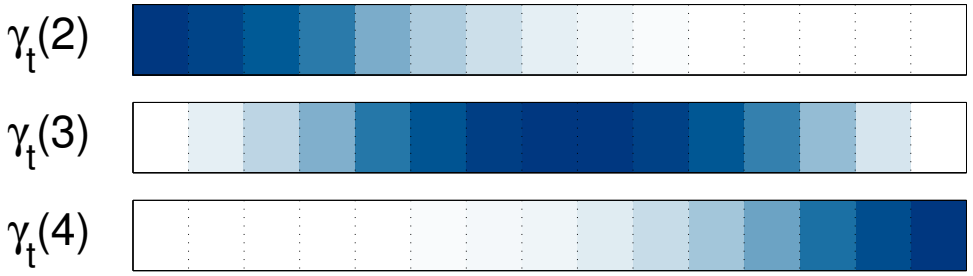
Training von DDHMM

Ausweg aus dem Problem, dass HMM-Zustände nicht beobachtbar:

Baum-Welch:

- Berücksichtigt alle Zustandsfolgen
- Hilfswahrscheinlichkeiten $\longrightarrow$ Erwartungswerte

>>>



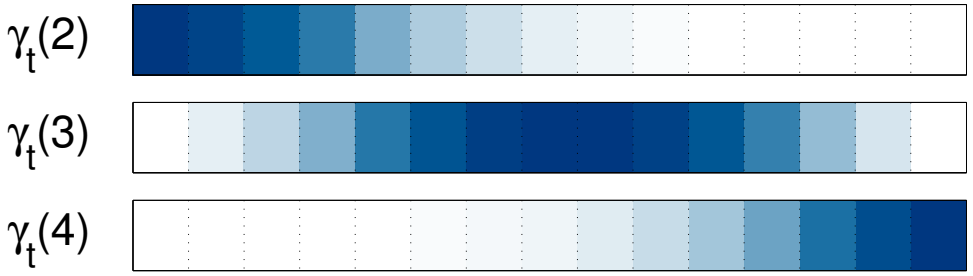
Training von DDHMM

Ausweg aus dem Problem, dass HMM-Zustände nicht beobachtbar:

Baum-Welch:

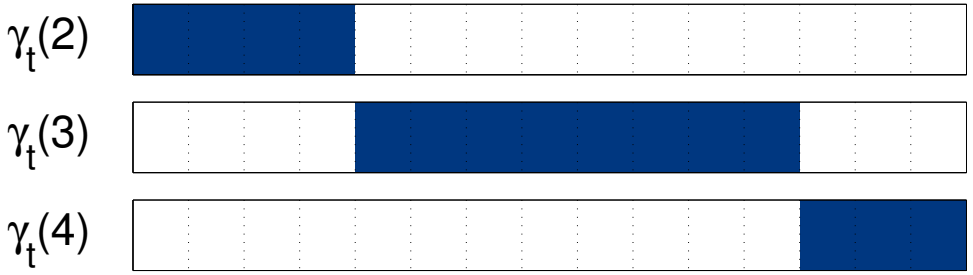
- Berücksichtigt alle Zustandsfolgen
- Hilfswahrscheinlichkeiten → Erwartungswerte

>>>



Viterbi:

- Berücksichtigt nur optimale Zustandsfolge



Viterbi-Training

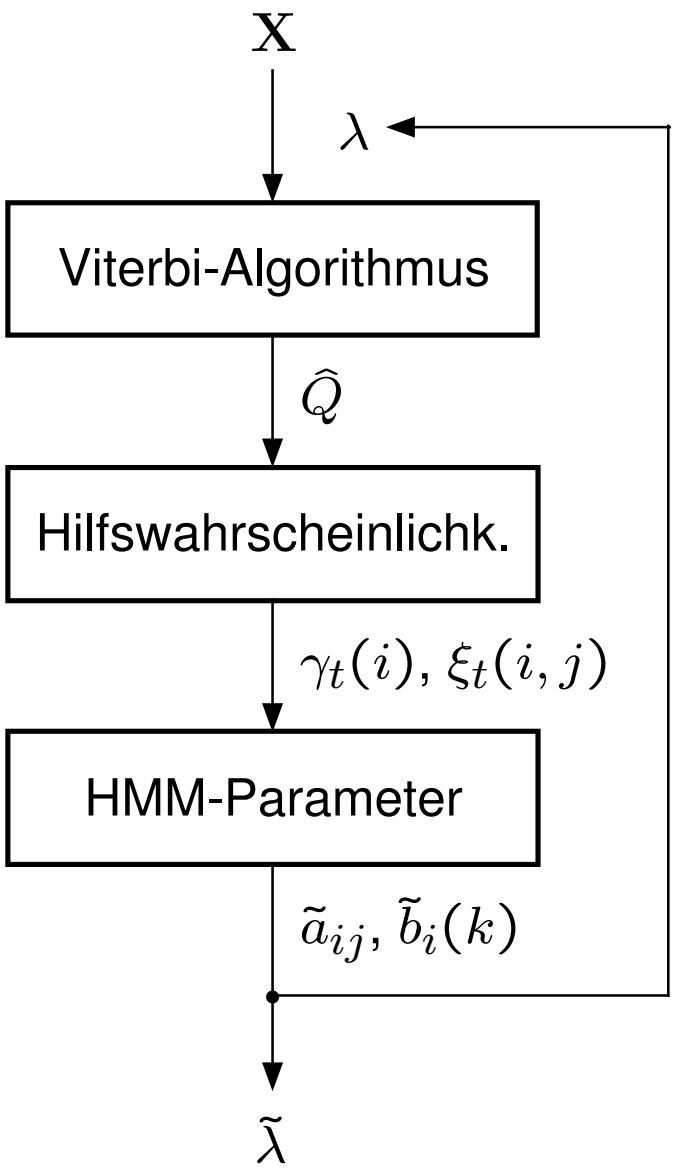
Aus der optimalen Zustandssequenz $\hat{Q} = S_1 \hat{q}_1 \ldots \hat{q}_T S_N$ können die Hilfsgrössen $\gamma_t(i)$ und $\xi_t(i,j)$ bestimmt werden mit:

$$\gamma_t(i) = \begin{cases} 1 & \text{falls } \hat{q}_t = S_i, \\ 0 & \text{sonst ,} \end{cases}$$

$$\xi_t(i,j) = \begin{cases} 1 & \text{falls } \hat{q}_t = S_i \text{ und } \hat{q}_{t+1} = S_j, \\ 0 & \text{sonst .} \end{cases}$$

Die Hilfsgrössen können in die Schätzformeln für a_{ij} und $b_j(k)$ des Baum-Welch-Algorithmus eingesetzt werden

Viterbi-Training



Initial-DDHMM

Anzahl der Zustände N, Anzahl der diskreten Beobachtungen M und Topologie (a_{ij} = 0) werden durch Anwendung bestimmt

Setzen der Initialwerte:

- Zustandsübergangswahrscheinlichkeiten: gleichverteilt

  $$a_{ij} = a_{ik}\,, \qquad \text{für } a_{ij}, a_{ik} > 0 \qquad \text{wobei } \sum_j a_{ij} = 1$$

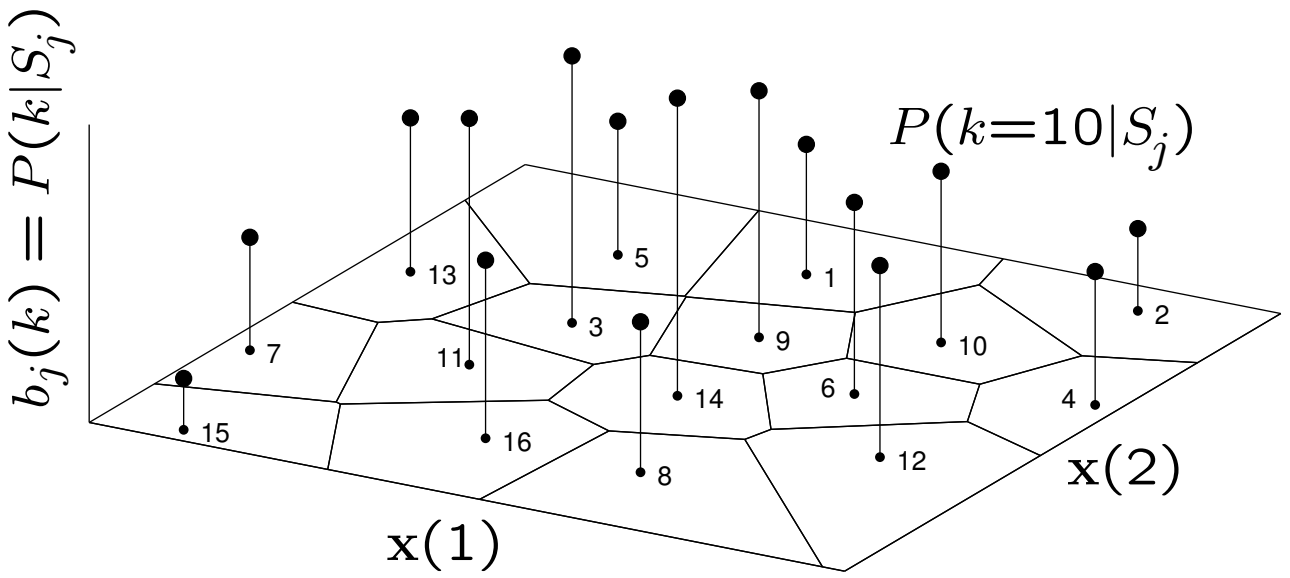
- Beobachtungswahrscheinlichkeiten: gleichverteilt

  $$b_j(k) = 1/M$$

HMM mit diskreten Beobachtungen

(DDHMM: discrete density HMM)

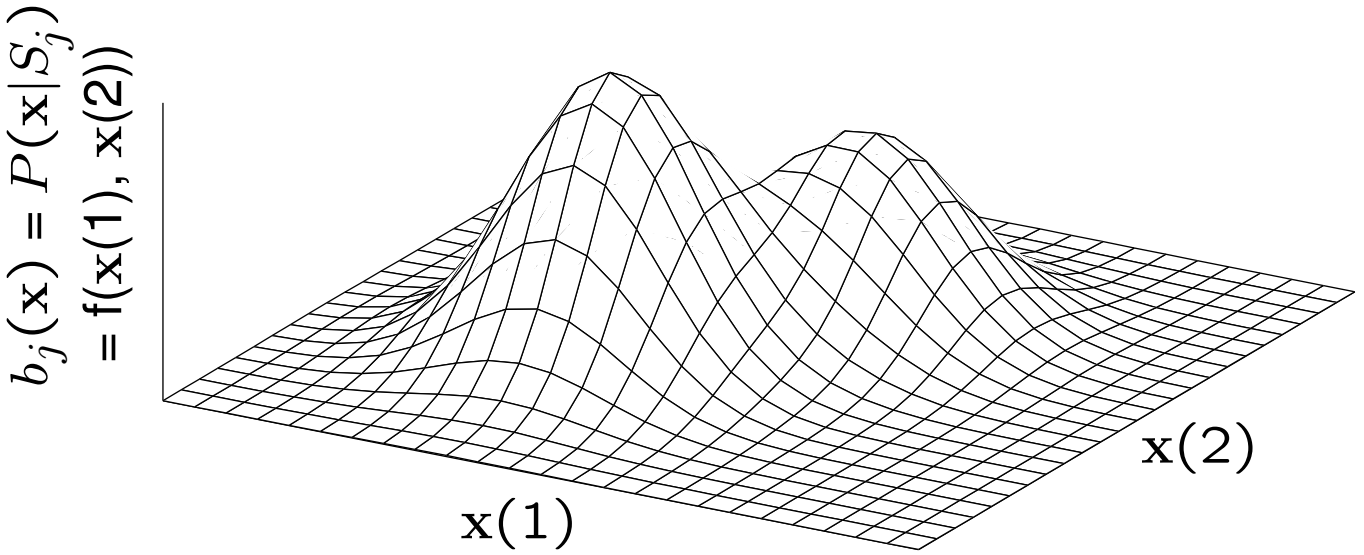
Beispiel: Wahrscheinlichkeitsverteilung für 2-dimensionale Beobachtungen mit 16 diskreten Werten



HMM mit kontinuierlichen Beobachtungen

(CDHMM: continuous density HMM)

Beispiel: Wahrscheinlichkeitsdichte für 2-dimensionale Beobachtungen
(2-dimensionale Gauss-Mischverteilung)



D-dimensionale Gauss-Mischverteilung mit M Komponenten

$$b_j(\mathbf{x}) \;=\; \sum_{k=1}^{M} c_{jk}\, b_{jk}(\mathbf{x}) \;=\; \sum_{k=1}^{M} c_{jk}\, \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})\;.$$

mit:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) \;=\; \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_{jk}|}}\, e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{jk})^{\mathsf{t}}\, \boldsymbol{\Sigma}_{jk}^{-1}\, (\mathbf{x}-\boldsymbol{\mu}_{jk})}$$

$$\sum_{k=1}^{M} c_{jk} \;=\; 1$$

$$c_{jk} \;\geq\; 0\,, \quad 1 < j < N\,, \quad 1 \leq k \leq M$$

Baum-Welch-Algorithmus für CDHMM

- Gleiche Hilfswahrscheinlichkeiten wie bei DDHMM $\gamma_t(i)$, $\xi_t(i,j)$
  aber $b_j(\mathbf{x})$ gemäss vorheriger Folie berechnen, bzw. nach Gleichung (77)

- Zusätzliche Hilfswahrscheinlichkeit $\zeta_t(j,k)$

$$\zeta_t(j,k) \quad = \quad P(q_t{=}S_j, \mathbf{x}_t{\mapsto}\mathcal{N}_{jk}|\mathbf{X}, \lambda)$$

  $\zeta_t(j,k)$ ist die Wahrscheinlichkeit, dass sich das HMM zum Zeitpunkt t im Zustand S_j befindet und die k-te Mischkomponente (des Zustandes S_j) die Beobachtung $\mathbf{x}_t$ erzeugt, gegeben die Beobachtungssequenz $\mathbf{X}$ und das Modell λ.

  Zugrundeliegende Idee: Aufteilung des Zustandes S_j >>>

Ermitteln der Hilfswahrscheinlichkeit $\zeta_t(j,k)$

$\zeta_t(j,k)$ kann mit den Vorwärts- und Rückwärtswahrscheinlichkeiten ausgedrückt werden:

$$\zeta_1(j,k) = \frac{a_{1j}\, c_{jk}\, b_{jk}(\mathbf{x}_1)\, \beta_1(j)}{P(\mathbf{X}|\lambda)}\,, \qquad 1 < j < N$$

$$\zeta_t(j,k) = \frac{\sum_{i=2}^{N-1} \alpha_{t-1}(i)\, a_{ij}\, c_{jk}\, b_{jk}(\mathbf{x}_t)\, \beta_t(j)}{P(\mathbf{X}|\lambda)}\,, \qquad \begin{array}{l} 1 < j < N \\ 1 < t \leq T \end{array}$$

Interpretation von $\zeta_t(j,k)$:

$$\sum_{t=1}^{T} \zeta_t(j,k) = \text{erwartete Anzahl Beobachtungen, die von der } k\text{-ten Mischkomponente des Zustands } S_j \text{ erzeugt wurden.}$$

Schätzformeln für CDHMM

Zustandsübergangswahrscheinlichkeiten a_{ij} (wie bei DDHMM)

$\longrightarrow$ aus Hilfswahrscheinlichkeiten $\gamma_t(i)$ und $\xi_t(i,j)$

Beobachtungswahrscheinlichkeiten (d.h. M Mischkomponenten)

c_{jk}

$\boldsymbol{\mu}_{jk}$ >>>

$\boldsymbol{\Sigma}_{jk}$

Initial-CDHMM

Anzahl der Zustände N, Topologie (a_{ij} = 0) und
Anzahl der Mischkomponenten M werden durch Anwendung bestimmt

Setzen der Initialwerte:

- Zustandsübergangswahrscheinlichkeiten: gleichverteilt

  $a_{ij} = 1/n_i\,,$ wobei $n_i =$ Anzahl der a_{ij}, die grösser als 0 sind

- Beobachtungswahrscheinlichkeiten: für alle S_j gleich

  $b_j(k) = c_k \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\,,$ wobei c_k, $\boldsymbol{\mu}_k$ und $\boldsymbol{\Sigma}_k$ aus $\mathbf{X}$ zu bestimmen sind

Training von CDHMM

Im Prinzip kann ein HMM wie folgt traininert werden:

1. Initial-HMM mit M Mischkomponenten erzeugen
2. mit Baum-Welch trainieren bis Konvergenz erreicht ist

Besser ist jedoch Mixture Splitting anzuwenden:

Verwenden eines einfacheren Modells zum Initialisieren eines komplexeren!

1. Initial-CDHMM mit 1 Mischkomponente erzeugen
2. mit Baum-Welch trainieren bis Konvergenz erreicht ist
3. Abbruch, falls Anzahl Mischkomponenten gleich M
4. Anzahl Mischkomponenten erhöhen >>>
5. bei Schritt 2 weiterfahren

Viterbi-Training (CDHMM)

$$\hat{Q} \;=\; S_1 \hat{q}_1 \ldots \hat{q}_T S_N$$

$$\gamma_t(i) \;=\; \begin{cases} 1 & \text{falls } \hat{q}_t = S_i, \\ 0 & \text{sonst ,} \end{cases}$$

$$\xi_t(i,j) \;=\; \begin{cases} 1 & \text{falls } \hat{q}_t = S_i \text{ und } \hat{q}_{t+1} = S_j, \\ 0 & \text{sonst .} \end{cases}$$

Viterbi-Training (CDHMM)

zusätzlich zur optimalen Zustandssequenz $\hat{Q} = S_1\, \hat{q}_1\, \hat{q}_2 \ldots \hat{q}_t \ldots \hat{q}_t\, S_N$ muss die optimale Mischkomponente $\hat{k}_t$ bestimmt werden:

$$\hat{k}_t \;=\; \underset{k}{\operatorname{argmax}}\; c_{jk} b_{\hat{q}_t k}(\mathbf{x}_t)$$

>>>

Damit kann die Hilfswahrscheinlichkeit $\zeta_t(i,k)$ berechnet werden:

$$\zeta_t(i,k) \;=\; \begin{cases} 1 & \text{falls } \hat{q}_t = S_i \text{ und } \hat{k}_t = k, \\ 0 & \text{sonst .} \end{cases}$$

Die Summe dieser Hilfswahrscheinlichkeit über die ganze Beobachtungssequenz hat die gewohnte Bedeutung

$$\sum_{t=1}^{T} \zeta_t(j,k) \;=\; \text{Anzahl Beobachtungen, die von der } k\text{-ten Misch-komponente des Zustands } S_j \text{ erzeugt wurden}$$

Training mit mehreren Sequenzen

Problem: Eine Beobachtungssequenz genügt nicht für Parameterschätzung!

Lösung: Schätzung der HMM-Parmeter aus einer Menge von Beobachtungssequenzen $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_S\}$

$\longrightarrow$ Maximierung der Produktionswahrscheinlichkeit $P(\mathcal{X}|\lambda)$

wobei $P(\mathcal{X}|\lambda) = \prod_{s=1}^{S} P(\mathbf{X}_s|\lambda)$.

Prinzip: Hilfswahrscheinlichkeiten über alle Sequenzen $s \in S$ aufsummieren:

$$\tilde{a}_{1j} = \frac{1}{S} \sum_{s=1}^{S} \gamma_1^s(j) \,, \qquad 1 < j < N$$

$$\tilde{a}_{ij} = \frac{\sum_{s=1}^{S} \sum_{t=1}^{T_s - 1} \xi_t^s(i,j)}{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_t^s(i)} \,, \qquad 1 < i, j < N$$

$$\tilde{a}_{iN} = \frac{\sum_{s=1}^{S} \gamma_{T_s}^s(i)}{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_t^s(i)} \,, \qquad 1 < i < N$$

Hinweis für Implementation

Für 1 Merkmalssequenz liefert der Baum-Welch-Algorithmus $\tilde{a}_{ij}$ wobei:

$$\tilde{a}_{ij} = \frac{\text{Übergänge } i \rightarrow j}{\text{Übergänge von } i \text{ aus}} = \frac{nt(i,j)}{ns(i)}, \qquad 1 \leq i,j \leq N$$

Für $s = 1 \ldots S$ Merkmalssequenzen kann somit $\tilde{a}_{ij}$ ermittelt werden mit:

$$\tilde{a}_{ij} = \frac{nt(i,j)}{ns(i)} = \frac{\sum_{s=1}^{S} nt_s(i,j)}{\sum_{s=1}^{S} ns_s(i)}, \qquad 1 \leq i,j \leq N$$

(Merke: $nt_s(i,j)$ und $ns_s(i)$ werden fortlaufend aufsummiert)

Underflow

Problem: Bei Berechnung von $\alpha_t(i)$, $\beta_t(i)$ und $\delta_t(i)$ werden die Wahrscheinlichkeiten mit zunehmendem t exponentiell kleiner

$\longrightarrow$ Unterlauf der Gleitkomma-Arithmetik

Abhilfe: Rechnen mit logarithmierten Wahrscheinlichkeiten >>>

Das Resultat wächst (betragsmässig) nur linear!

Spracherkennung mit HMM

Vorhanden: Algorithmen für DDHMM und CDHMM

$\longrightarrow$ Anwendung für die Spracherkennung

d.h. Schätzen der Wahrscheinlichkeit: $P(\mathbf{X}|W) \approx P(\mathbf{X}|\lambda_W)$

Spracherkennung mit HMM

Vorhanden: Algorithmen für DDHMM und CDHMM

$\longrightarrow$ Anwendung für die Spracherkennung

d.h. Schätzen der Wahrscheinlichkeit: $P(\mathbf{X}|W) \approx P(\mathbf{X}|\lambda_W)$

Annahme: W ist nur ein Wort lang $\longrightarrow$ $W \in V = \{v_1, v_2, \ldots, v_{|V|}\}$

Spracherkennung mit HMM

Vorhanden: Algorithmen für DDHMM und CDHMM

$\longrightarrow$ Anwendung für die Spracherkennung

d.h. Schätzen der Wahrscheinlichkeit: $P(\mathbf{X}|W) \approx P(\mathbf{X}|\lambda_W)$

Annahme: W ist nur ein Wort lang $\longrightarrow$ $W \in V = \{v_1, v_2, \ldots, v_{|V|}\}$

Gesucht: Akustische Modelle λ_j für $v_j, j = 1 \ldots |V|$

Akustische Modelle für Wörter

- Welche Sprachmerkmale sind nützlich?
- Sollen DDHMM oder CDHMM eingesetzt werden?
- Wie erhält man ein HMM für ein bestimmtes Wort?
- Wie werden gültige Wörter von andern unterschieden?

Akustische Modelle für Wörter

- Welche Sprachmerkmale sind nützlich?
- Sollen DDHMM oder CDHMM eingesetzt werden?
- Wie erhält man ein HMM für ein bestimmtes Wort?
- Wie werden gültige Wörter von andern unterschieden?

Sprachmerkmale für die Spracherkennung

Primär verwendet: MFCC (vergl. Buch Kapitel 11.7)

Häufiges Problem: verschiedene Übertragungskanäle

Mögliche Lösungen:

a) Verwendung mittelwertfreier MFCC (vergl. Kapitel 4.6.7)

Mittelwert nur aus längerem Sprachsignal schätzbar!

b) Verwendung von Delta-MFCC (vergl. Kapitel 4.6.6)
(zusätzlich zu MFCC, nicht als Ersatz)

$\longrightarrow$ Verwendung mehrerer Merkmale!

Verwendung mehrerer Sprachmerkmale

1. Die F Merkmale $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(F)}$ in einem Vektor zusammenfassen.
   Ist nur für CDHMM geeignet! (DDHMM: zu grosses Codebuch für VQ nötig)

2. Bei DDHMM: ein separates Codebuch pro Merkmal und Beobachtungswahrscheinlichkeiten ermitteln nach:

$$b_i(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(F)}) = \prod_{f=1}^{F} [b_i^{(f)}(\mathbf{x}^{(f)})]^{\alpha^{(f)}}$$

3. Je ein separates HMM pro Merkmal und Produktionswsk. ermitteln nach:

$$P(\mathbf{X}|\lambda) = \prod_{f=1}^{F} [P(\mathbf{X}^{(f)}|\lambda^{(f)})]^{\alpha^{(f)}}$$

Akustische Modelle für Wörter

- Welche Sprachmerkmale sind nützlich?
- Sollen DDHMM oder CDHMM eingesetzt werden?
- Wie erhält man ein HMM für ein bestimmtes Wort?
- Wie werden gültige Wörter von andern unterschieden?

Spracherkennung mit DDHMM

Vorteil: + praktisch kein Rechenbedarf für $b_j(k)$

Nachteile:
- – Quantisierungsfehler der VQ erhöht die Fehlerrate
- – grösseres Codebuch erhöht den Rechenbedarf
- – Bestimmung der Gewichte $\alpha^{(f)}$ für die einzelnen Merkmale ist in der Praxis schwierig

Spracherkennung mit CDHMM

Vorteile:

+ keine VQ nötig
+ bei cepstralen Merkmalen ist Kovarianzmatrix Σ durch Diagonalmatrix approximierbar
  (Reduktion der Anzahl Modellparameter auf wenige %)

>>>

Nachteil:

– erheblicher Rechenbedarf für $b_j(\mathbf{x})$

Akustische Modelle für Wörter

- Welche Sprachmerkmale sind nützlich?
- Sollen DDHMM oder CDHMM eingesetzt werden?
- Wie erhält man ein HMM für ein bestimmtes Wort?
- Wie werden gültige Wörter von andern unterschieden?

Erzeugen eines Wortmodells

- Sprachaufnahmen: auf den vorgesehenen Einsatz abstimmen
  - Auswahl und Anzahl der Sprecher
  - akust. Umgebung / Übertragungskanal
- Wahl des HMM: Initial-HMM >>>
- Training des HMM: mit Baum-Welch-Algorithmus oder Viterbi-Training

Akustische Modelle für Wörter

- Welche Sprachmerkmale sind nützlich?
- Sollen DDHMM oder CDHMM eingesetzt werden?
- Wie erhält man ein HMM für ein bestimmtes Wort?
- Wie werden gültige Wörter von andern unterschieden?

Spracherkenner mit Wort-HMM

Gegeben:
- HMM $\lambda_1, \lambda_2, \ldots \lambda_{|V|}$ für jedes Wort $v_j \in V = \{v_1, v_2, \ldots, v_{|V|}\}$
- eine Merkmalssequenz $\mathbf{X}$

Aufgabe: Bestimmung des Wortes v_j

mit: $j = \underset{i}{\operatorname{argmax}}\ P(\mathbf{X}|\lambda_i)$ (max. Produktionswahrscheinlichkeit)

oder: $j = \underset{i}{\operatorname{argmax}}\ P(\mathbf{X}, \hat{Q}_i|\lambda_i)$ (max. Viterbi-Wahrscheinlichkeit)

Problem: Für jede Merkmalssequenz $\mathbf{X}$ wird ein j gefunden,
auch wenn kein Wort des Vokabulars gesprochen worden ist!

Spracherkenner mit Wort-HMM

Gegeben:
- HMM $\lambda_1, \lambda_2, \ldots \lambda_{|V|}$ für jedes Wort $v_j \in V = \{v_1, v_2, \ldots, v_{|V|}\}$
- eine Merkmalssequenz $\mathbf{X}$

Aufgabe: Bestimmung des Wortes v_j

mit: $j = \underset{i}{\operatorname{argmax}}\ P(\mathbf{X}|\lambda_i)$ (max. Produktionswahrscheinlichkeit)

oder: $j = \underset{i}{\operatorname{argmax}}\ P(\mathbf{X}, \hat{Q}_i|\lambda_i)$ (max. Viterbi-Wahrscheinlichkeit)

Problem: Für jede Merkmalssequenz $\mathbf{X}$ wird ein j gefunden,
auch wenn kein Wort des Vokabulars gesprochen worden ist!

Lösung:

Spracherkenner mit Wort-HMM

Gegeben:
- HMM $\lambda_1, \lambda_2, \ldots \lambda_{|V|}$ für jedes Wort $v_j \in V = \{v_1, v_2, \ldots, v_{|V|}\}$
- eine Merkmalssequenz $\mathbf{X}$

Aufgabe: Bestimmung des Wortes v_j

mit: $j = \underset{i}{\text{argmax}}\ P(\mathbf{X}|\lambda_i)$ (max. Produktionswahrscheinlichkeit)

oder: $j = \underset{i}{\text{argmax}}\ P(\mathbf{X}, \hat{Q}_i|\lambda_i)$ (max. Viterbi-Wahrscheinlichkeit)

Problem: Für jede Merkmalssequenz $\mathbf{X}$ wird ein j gefunden,
auch wenn kein Wort des Vokabulars gesprochen worden ist!

Lösung: Einsatz eines Rückweisungsmodells λ_R

Rückweisungsmodell

Anforderung:
$$P(\mathbf{X}|\lambda_R) \leq P(\mathbf{X}|\lambda_w)\,, \qquad \text{für } w \in V$$
$$P(\mathbf{X}|\lambda_R) \geq P(\mathbf{X}|\lambda_w)\,, \qquad \text{für } w \notin V$$

Training: für Wörter $w \notin V$ meist keine Trainingsdaten vorhanden!

Rückweisungsmodell

Anforderung:
$$P(\mathbf{X}|\lambda_R) \leq P(\mathbf{X}|\lambda_w)\,, \qquad \text{für } w \in V$$
$$P(\mathbf{X}|\lambda_R) \geq P(\mathbf{X}|\lambda_w)\,, \qquad \text{für } w \notin V$$

Training: für Wörter $w \notin V$ meist keine Trainingsdaten vorhanden!

$\longrightarrow$ λ_R wird mit den Trainingsdaten aller $w \in V$ trainiert

Rückweisungsmodell

Anforderung:
$$aP(\mathbf{X}|\lambda_R) \leq P(\mathbf{X}|\lambda_w)\,, \qquad \text{für } w \in V$$
$$aP(\mathbf{X}|\lambda_R) \geq P(\mathbf{X}|\lambda_w)\,, \qquad \text{für } w \notin V$$

Training: für Wörter $w \notin V$ meist keine Trainingsdaten vorhanden!

$\longrightarrow$ λ_R wird mit den Trainingsdaten aller $w \in V$ trainiert

$\longrightarrow$ Optimierung von a nötig, damit Anforderung erfüllbar!

Modelle für andere akustische Ereignisse

Problem: Merkmalssequenz $\mathbf{X}$ kann auch nichtsprachliche Ereignisse enthalten!

(z.B. weil Äusserung falsch detektiert worden ist) >>>

Nötig: zusätzliche Modelle für typische Geräusche und Pausen

Erkennung: Erkennungsnetzwerk und Viterbi-Algorithmus >>>

Akustische Modelle für Wörter

- Welche Sprachmerkmale sind nützlich?
- Sollen DDHMM oder CDHMM eingesetzt werden?
- Wie erhält man ein HMM für ein bestimmtes Wort?
- Wie werden gültige Wörter von andern unterschieden?

Thema der nächsten Lektionen:

- Akustische Modellierung von Wortteilen
- Spracherkennung mit Laut-HMM

Zur Übersicht der Vorlesung *Sprachverarbeitung II* >>>

HMM:

```
a = [ 0.0  1.0  0.0  0.0  0.0          b = [ 0.0  0.0
      0.0  0.7  0.3  0.0  0.0                0.6  0.4
      0.0  0.0  0.7  0.3  0.0                0.4  0.6
      0.0  0.0  0.0  0.7  0.3                0.6  0.4
      0.0  0.0  0.0  0.0  0.0 ];             0.0  0.0 ];
```

Beobachtungssequenz:

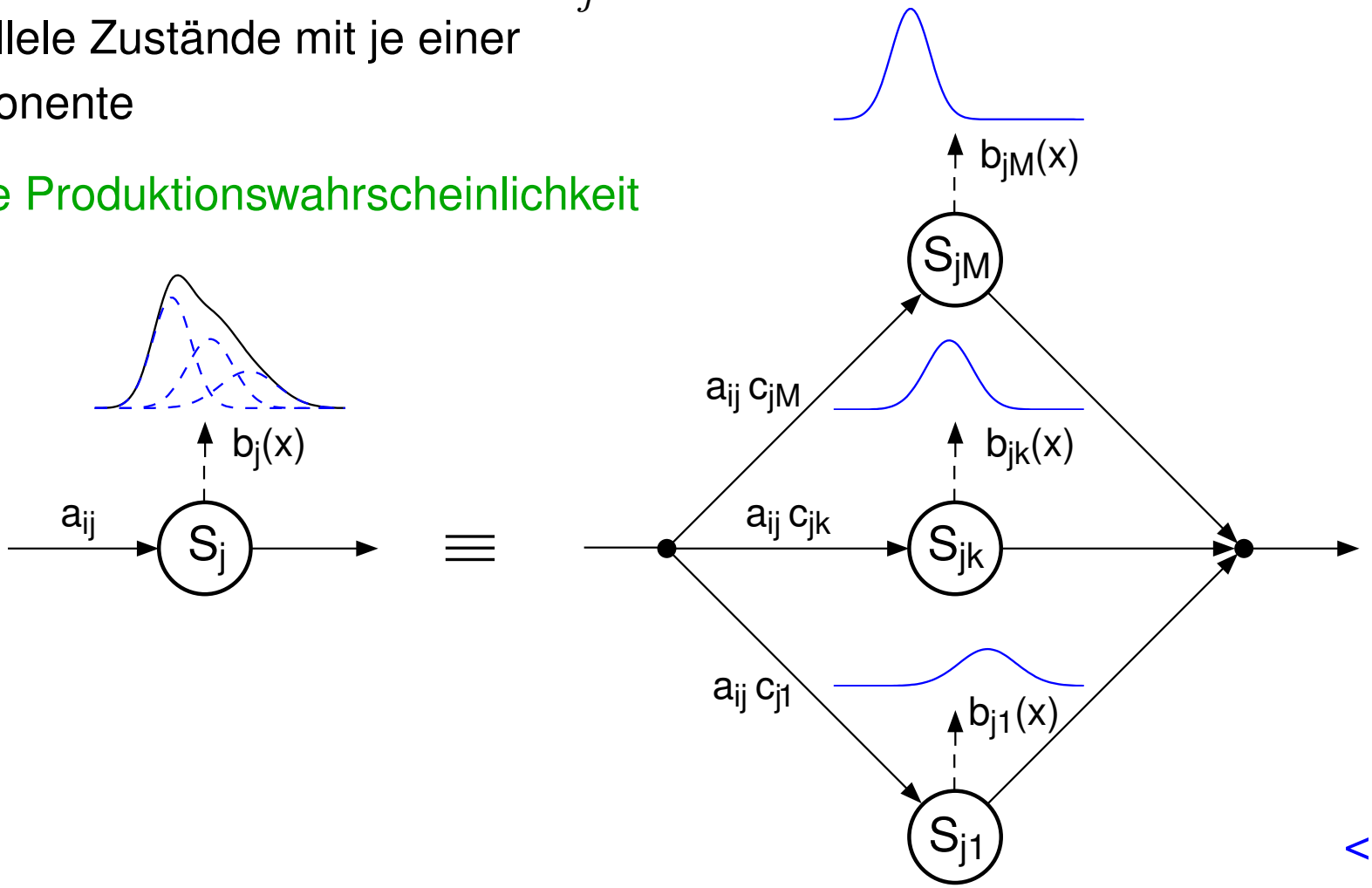
```
X = [ 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 ];
```

<<<

Aufteilung des Zustandes S_j

M Mischkomponenten des Zustandes S_j auf M parallele Zustände mit je einer Mischkomponente

$\longrightarrow$ gleiche Produktionswahrscheinlichkeit



<<<

$$\tilde{c}_{jk} = \frac{\text{erwartete Anzahl der von } b_{jk}(\mathbf{x}) \text{ erzeugten Beobachtungen}}{\text{erwartete Anzahl der von } b_j(\mathbf{x}) \text{ erzeugten Beobachtungen}} = \frac{\sum_{t=1}^{T} \zeta_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\tilde{\boldsymbol{\mu}}_{jk} = \frac{\text{erwartete Summe der von } b_{jk}(\mathbf{x}) \text{ erzeugten Beobachtungen}}{\text{erwartete Anzahl von } b_{jk}(\mathbf{x}) \text{ erzeugten Beobachtungen}} = \frac{\sum_{t=1}^{T} \zeta_t(j,k)\mathbf{x}_t}{\sum_{t=1}^{T} \zeta_t(j,k)}$$

$$\tilde{\boldsymbol{\Sigma}}_{jk} = \frac{\sum_{t=1}^{T} \zeta_t(j,k)(\mathbf{x}_t - \tilde{\boldsymbol{\mu}}_{jk})(\mathbf{x}_t - \tilde{\boldsymbol{\mu}}_{jk})^{\mathsf{t}}}{\sum_{t=1}^{T} \zeta_t(j,k)}$$

<<<

Mixture Splitting

$$
\begin{aligned}
c &\rightarrow c' = c'' = 0.5c \\
\mu &\rightarrow \mu' = \mu + 0.2\sigma\,, \quad \mu'' = \mu - 0.2\sigma \\
\Sigma &\rightarrow \Sigma' = \Sigma'' = \Sigma
\end{aligned}
$$

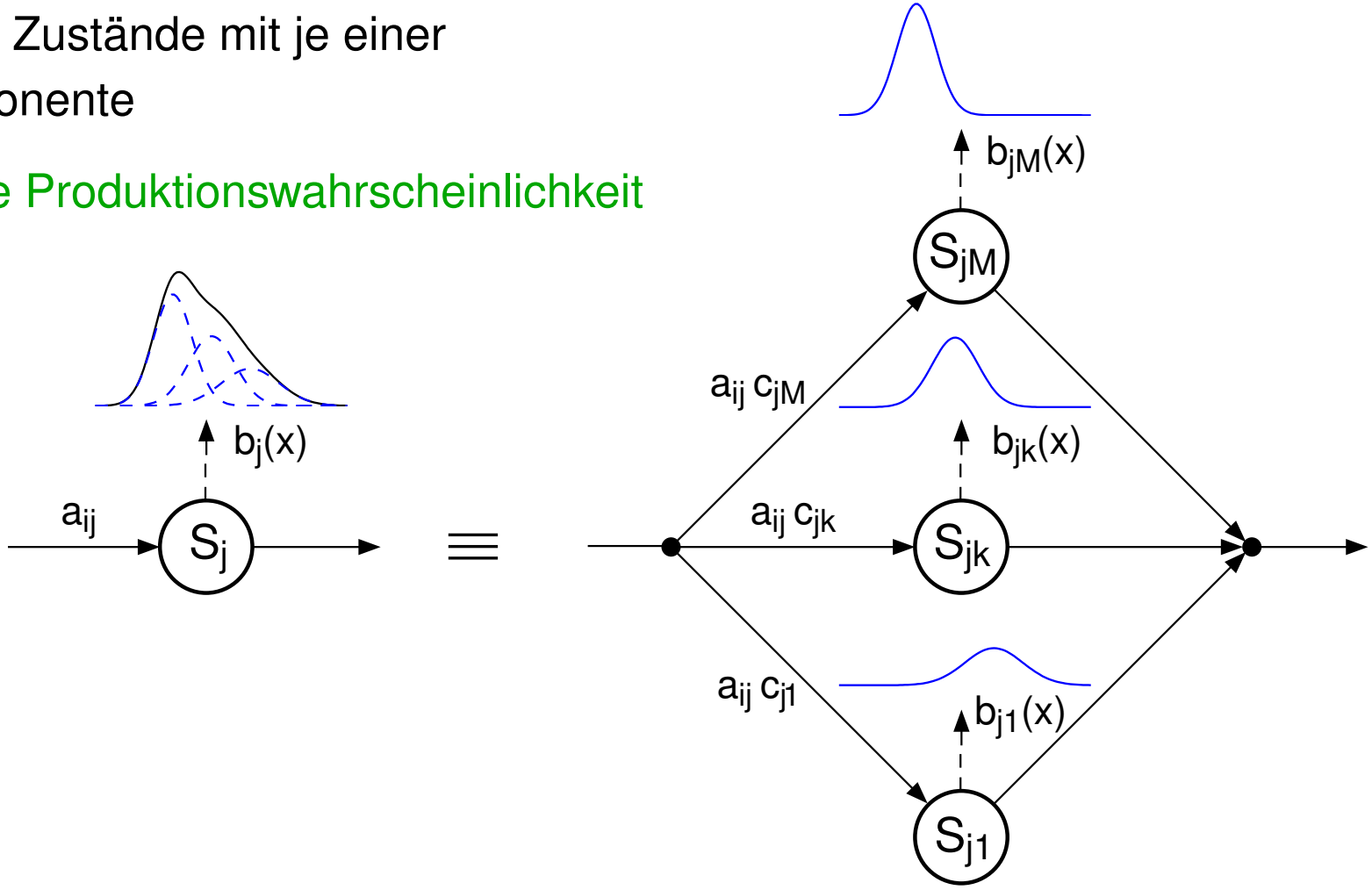
Standardabweichung $\sigma = \sqrt{\text{diag}(\Sigma)}$.

<<<

Aufteilung des Zustandes S_j

M parallele Zustände mit je einer Mischkomponente

$\longrightarrow$ gleiche Produktionswahrscheinlichkeit



<<<

Logarithmierter Viterbi-Algorithmus

Rekursionsgleichung für die Berechnung von $\delta_t(j)$:

Original: $$\delta_t(j) = \max_i \left[\delta_{t-1}(i)\, a_{ij}\right] b_j(\mathbf{x}_t)$$

Logarithmiert: $$\log \delta_t(j) = \max_i \left[\log \delta_{t-1}(i) + \log a_{ij}\right] + \log b_j(\mathbf{x}_t)$$

Problem: $\log 0$ ist nicht definiert!

In Anlehnung an die beiden Grenzwerte

$$\begin{aligned} \lim_{x \to 0+} \log_b x &= -\infty \\ \lim_{x \to -\infty} b^x &= 0 \end{aligned}$$

definiert man in der Praxis

$$\begin{aligned} \log_b 0 &\doteq -\infty \quad \text{und} \\ b^{-\infty} &\doteq 0\,, \end{aligned}$$

was zwar mathematisch nicht korrekt ist, aber die Berechnungen wesentlich vereinfacht und das Resultat nicht beeinflusst.

Logarithmus einer Summe

z.B. zum Berechnen der Vorwärtswahrscheinlichkeit:

$$\alpha_t(j) \;=\; \left[\sum_{i=2}^{N-1} \alpha_{t-1}(i)\, a_{ij}\right] b_j(\mathbf{x}_t)$$

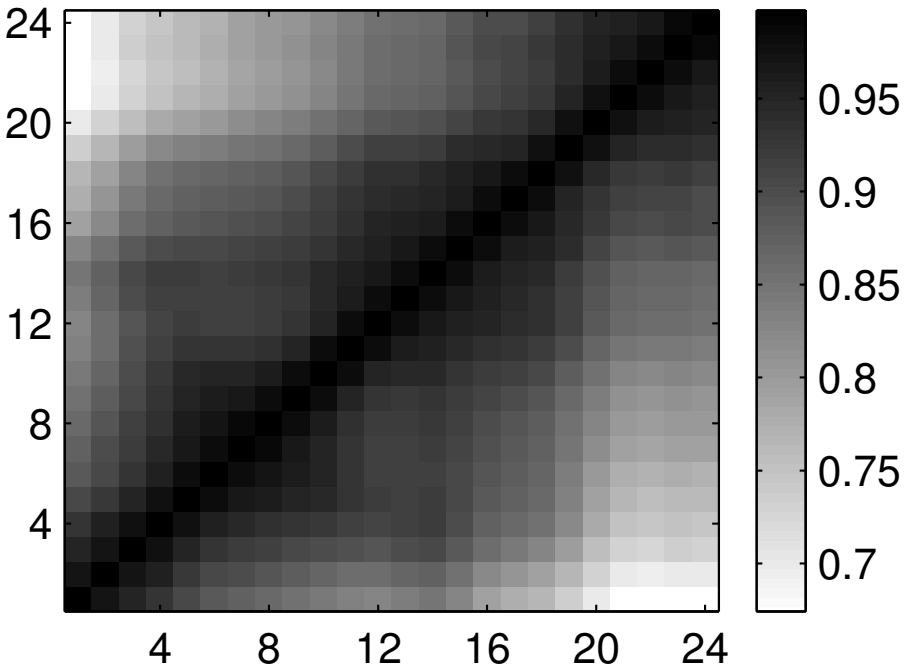
Kingsbury-Rayner-Formel:

$$\begin{aligned}
\log_b(x+y) &= \log_b\left(x\left(1+\frac{y}{x}\right)\right) \\
&= \log_b x + \log_b\left(1+\frac{y}{x}\right) \\
&= \log_b x + \log_b\left(1+b^{\log_b y - \log_b x}\right)
\end{aligned}$$

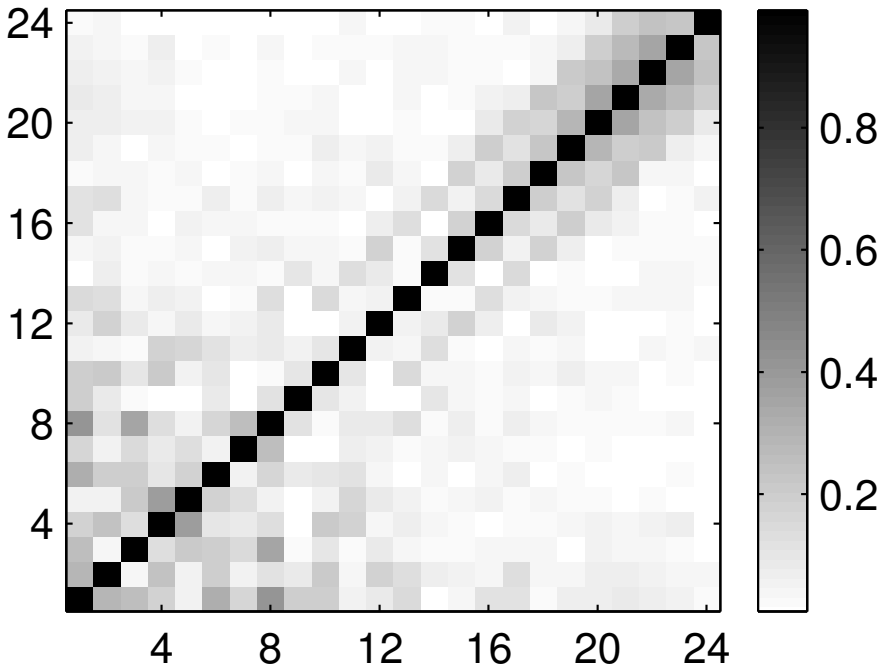
mit $x \geq y$ und $x > 0$.

<<<

Kovarianzmatrizen



24-kanalige Mel-Filterbank



MFCC

<<<

Wahl des Initial-HMM

Anzahl Zustände: HMM für Wort mit n_L Lauten: ca. $3\,n_L$ Zustände
HMM für Einzellaute: 2 – 5 Zustände

Topologie: LR-Modell oder lineares Modell oder ... >>>
$\longrightarrow$ Wahl der a_{ij} nicht kritisch

Beobachtungswsk.:

a) Segmentierte Daten: Viterbi-Training, Start mit 1 Mischkomponente

b) Flat-Start DDHMM: gleichverteilt

Flat-Start CDHMM: gleichförmige Segmentierung, 1 Mischkomponente für alle S_j

<<<

Wahl des Initial-HMM

Anzahl Zustände: HMM für Wort mit n_L Lauten: ca. $3\,n_L$ Zustände

Topologie: LR-Modell oder lineares Modell oder ... >>>
$\longrightarrow$ Wahl der a_{ij} nicht kritisch

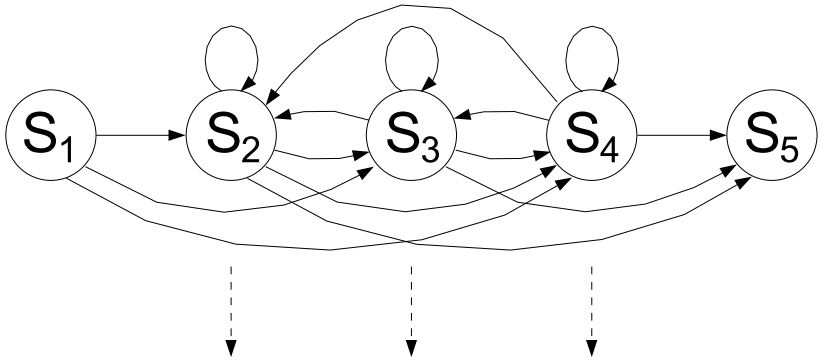
Beobachtungswsk.: DDHMM: gleichverteilt

CDHMM: 1 Mischkomponente, für alle S_j gleich
($\boldsymbol{\mu}$ und $\boldsymbol{\Sigma}$ über alle Trainingsdaten)

<<<

Topologie des HMM

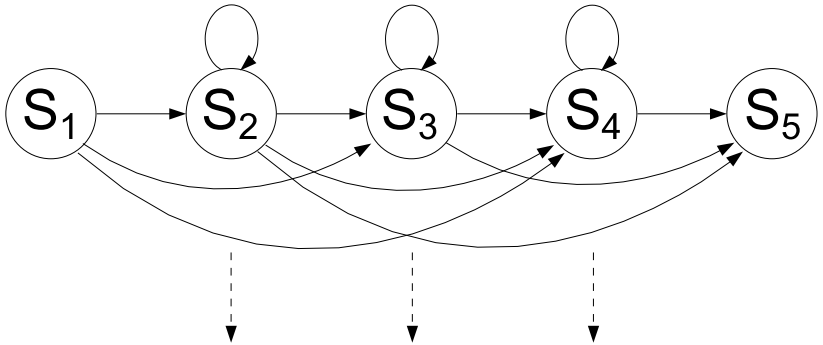
Vollverbundenes HMM:



$$A = \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} & 0 \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Topologie des HMM

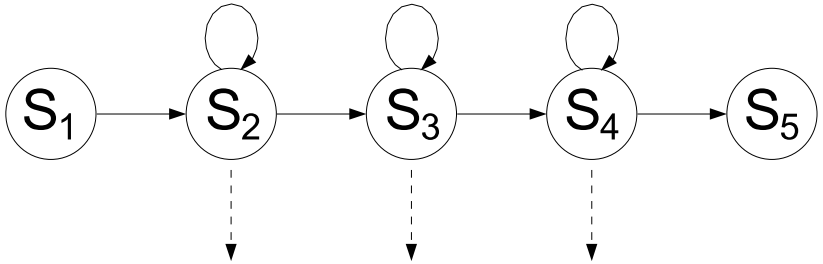
Links-Rechts-HMM: $a_{ij} = 0\ , \qquad \forall j < i$



$$A = \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} & 0 \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Topologie des HMM

Lineares HMM: $a_{ij} = 0\ , \qquad \forall j < i \quad \text{und} \quad \forall j > i+1$

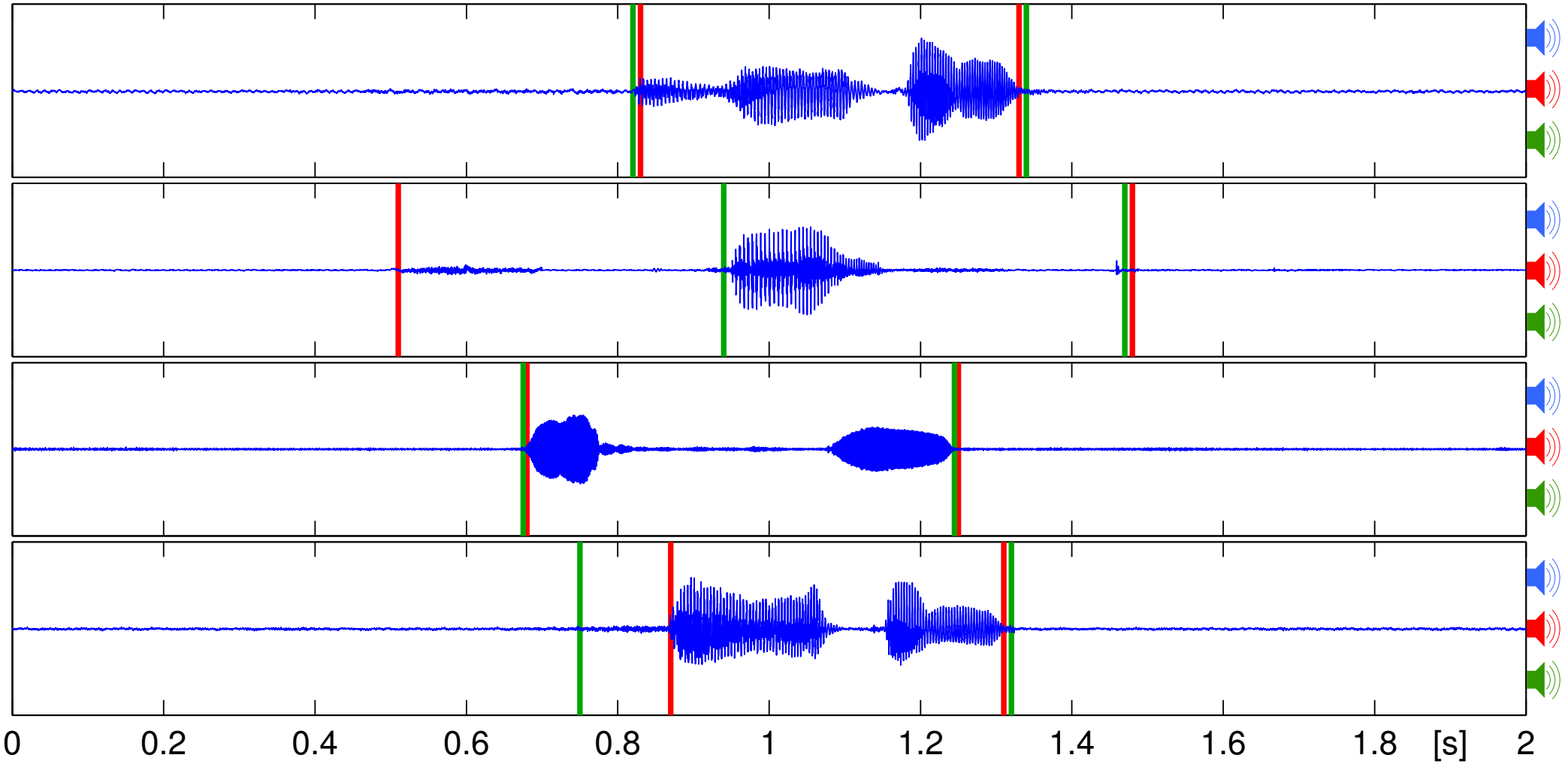


$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

<<<

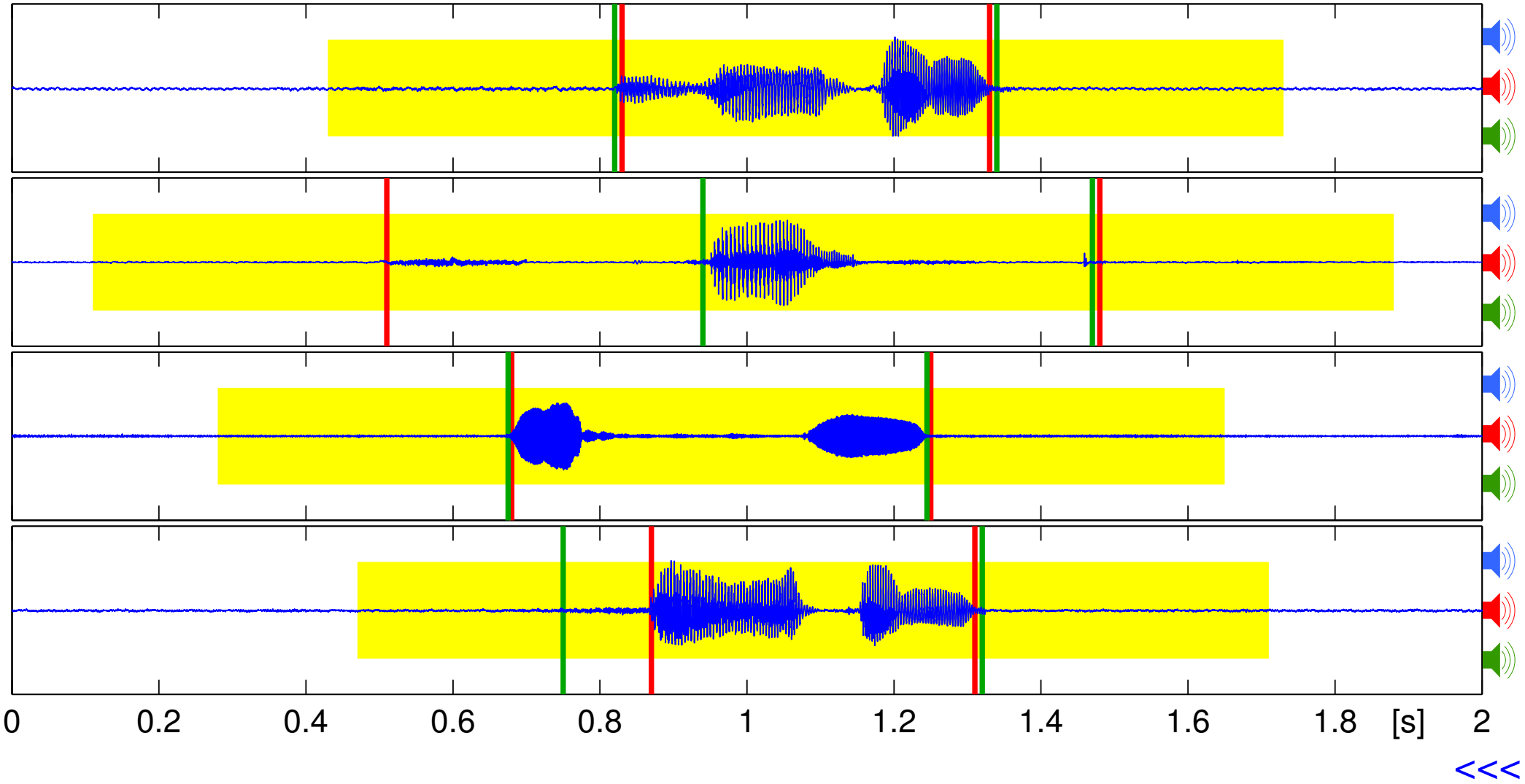
Detektion von Äusserungen

automatisch manuell



Detektion von Äusserungen

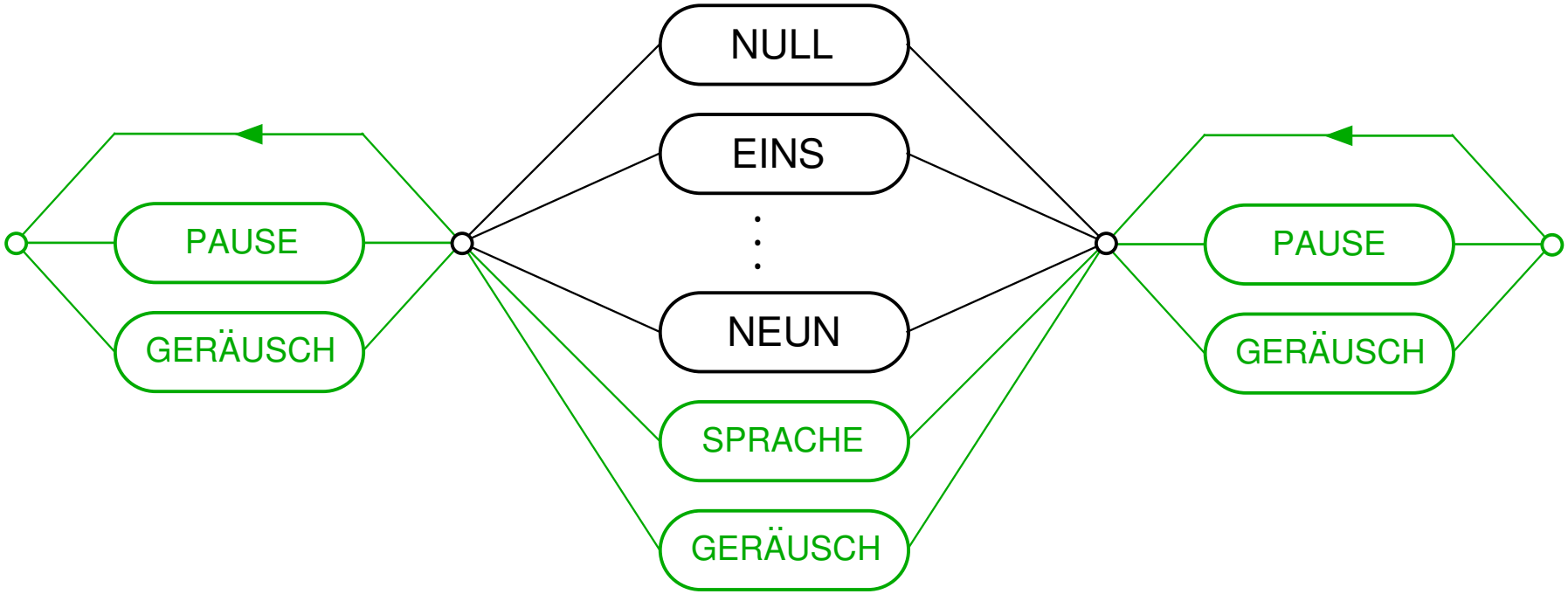
automatisch manuell



<<<

Ziffernerkenner

Erkennungsnetzwerk zur Kompensation von Mängeln der Äusserungsdetektion



<<<