

Inhaltsverzeichnis

1	fickerboy	1
2	11 - HMM (Hidden Markov Models)	1
2.1	Sequenzmodellierung und State-Modelierung	1
2.2	Dynamic Time Warping	1
2.3	Markov-Modelle	3
2.4	Hidden-Markov-Modelle	4

1 fickerboy

Hier steht der Inhalt. hören und sehen du ficker äöü

2 11 - HMM (Hidden Markov Models)

- Modellieren Sequenz von Datenpunkten
- benötigen zugrundeliegendes state modeling
- oft zusammen mit GMMs verwendet

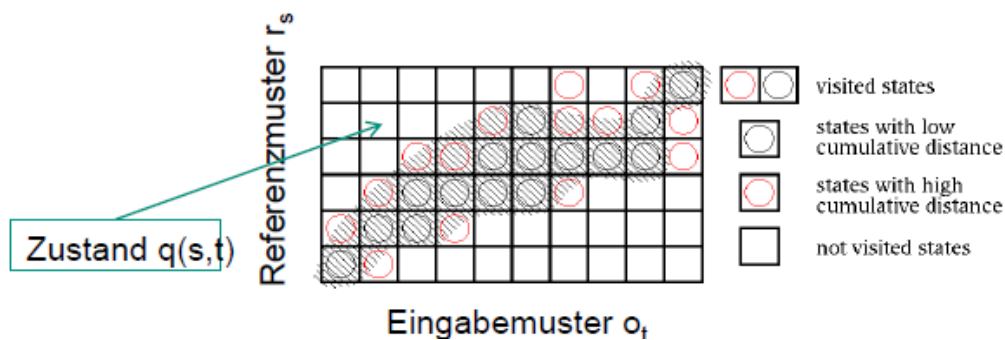
2.1 Sequenzmodellierung und State-Modelierung

- Sequenzmodellierung ist in typischer Signalverarbeitungskette letzte Schritt nach Datenverarbeitung und State Modeling
- Klassifikation und Sequenzmodellierung eng miteinander verbunden

2.2 Dynamic Time Warping

- einfaches Verfahren zum Vergleich von Sequenzen
- Algorithmen in der HMM-Modellierung sehr ähnlich zu DTW
- Wir haben: Aufnahmen von Sprachsignalen - Trainingsdaten (Beispielaufnahmen mit bekanntem Inhalt) + Testdaten (Aufnahmen mit unbekanntem Inhalt)
- Ziel: Wir wollen die Distanz einer unbekannten Sequenz und einer Beispielsequenz berechnen
- Frame für Frame-Vergleich Probleme: Signale sind unterschiedlich lang + Anfang und Ende der Äußerung nicht bekannt
- Fagot-Lösung: Lineares Alignment - für fast alle Zwecke aber viel zu unflexibel
- Killer-Lösung: DTW

- basiert auf Prinzip des dynamischen Programmierens (DP) bzw. der minimalen Editierdistanz
- Pfade durch eine Matrix von möglichen Zuordnungen berechnet
- Ergebnis: Distanzmaß zwischen den beiden Äußerungen
- Ziel: Finde Distanz zwischen den beiden Äußerungen (je niedriger desto besser)
- Problem: Alle Pfade müssen betrachtet werden um den Besten zu finden
- Lösung:
 - Berechne für jede Zeit t die kumulativen Distanzen $\alpha(s, t)$, die die Distanz der Teiläußerungen bis zu den Zuständen $q(s, t)$ ($s=1, \dots, S$) beschreiben
 - Die Distanzen für Zeitpunkt $t+1$ berechnen sich iterativ aus denen für Zeitpunkt t und hier wird Minimierung der Distanz durchgeführt
- Benötigte Distanzmaß $d(s, t)$ für den beobachteten Frame t und den Referenzframe s (z.B. euklidische Distanz)



- Welche Übergänge zwischen Frames sind möglich? Was haben sie für Distanz-Kosten?
- Erlaubt sind üblicherweise:
 - Ersetzung: Kosten = $d(.,.)$ (praktisch immer > 0)
 - Einfügung/Auslassung eines Frames: Kosten können in der Praxis ignoriert werden
 - Einfügung/Auslassung mehrerer Frames: evtl. Extra-penalty, max. Zahl von Frames, die ausgelassen werden dürfen

Ablauf des Algorithmus:

- Initialisierung: Beginne bei Startzustand $q(0, 0)$, $t := 0$, $\alpha(0, 0) := d(0, 0)$, $\alpha(x, 0) = \infty$
- Für jeden Zustand $q(s, t)$:
 - Betrachte jeden erlaubten Zustandsübergang $q(s', t-1) \rightarrow q(s, t)$
 - Finde min. Distanz zu $q(s, t)$
 - Bis Teildistanz $\alpha(s, t)$ einen gewissen Grenzwert überschreitet
- weitere Einschränkungen des Suchraums denkbar

- Komponenten der Zustandsmatrix Schritt für Schritt berechenbar (zeiteffizient + speichereffizient)
- Anwendung in der Spracherkennung
- z.B. heute noch praktisch bei der Erkennung von sehr kleinen Vokabularen

Probleme bei Unterscheidung einer kleinen Menge von Wörtern:

- benötigt eine Endpunktdetektion
- wird sehr ineffizient wenn viele Trainingsbeispiele vorhanden sind - großes Vok. braucht extrem viele Trainingsbeispiele
- Trainingsdaten können nicht zwischen verschiedenen Referenzen geteilt werden
- Erkennung unbekannter Wörter ist nicht möglich
- ungeeignet für kontinuierliche Sprache
- sehr kurze Wörter sind schwer zu trainieren

⇒ Andere Methode wird benötigt die es ermöglicht, kleinere Einheiten (Silben, Phoneme) zu trainieren und zu erkennen

2.3 Markov-Modelle

Sprachproduktion als stochastischer Prozess

- Beobachtungen zur Sprachproduktion:
 - das gleiche Wort/Phonem hört sich jedesmal anders an
 - in einem gegebenen Zustand können verschiedene Laute mit unterschiedlicher Wahrscheinlichkeit beobachtet werden
 - der Produktionsprozess kann Übergänge aus einem Zustand in einen anderen machen, aber nicht alle denkbaren Übergänge sind möglich, zumindest nicht gleich wahrscheinlich
- Sprachprozess befindet sich zu jedem Zeitpunkt in einem Zustand
- In jedem Zustand werden Laute ausgegeben entsprechend einer gewissen Wahrscheinlichkeit: Emissionswahrscheinlichkeit
- Die Übergänge zwischen Zuständen erfolgen auch entsprechend einer gewissen Wahrscheinlichkeitsverteilung: Übergangs- oder Transitionswahrscheinlichkeiten
- Markov-Modelle:
 - Es gibt eine diskrete Zustandsmenge s_1, \dots, s_N
 - Wir beobachten eine probabilistische Zustandssequenz $O = (o_1, \dots, o_T), o_i \in 1, \dots, N$
 - Markov-Annahme: Wahrscheinlich, dass wir zum Zeitpunkt t in einem gewissen Zustand sind, hängt nur von vorhergehendem Zustand ab
 - Verteilung soll stationär (zeitunabhängig) sein

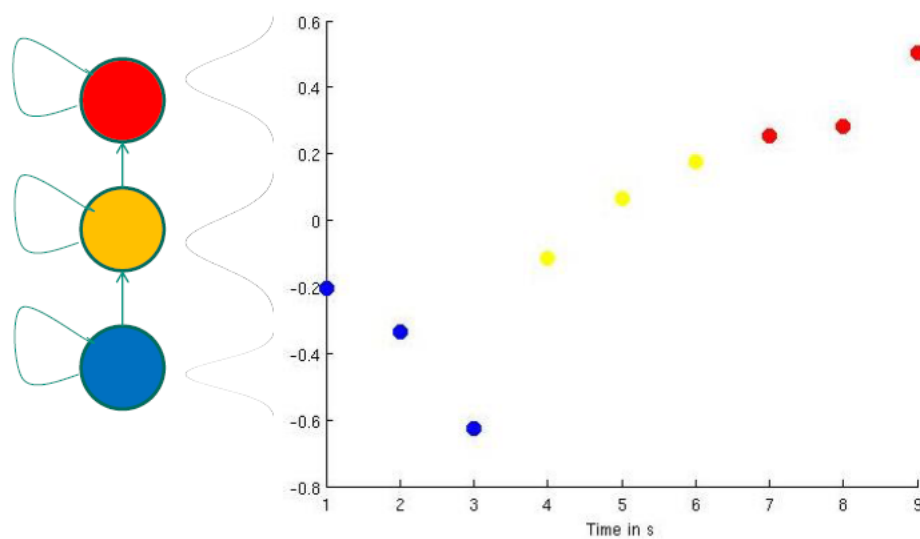
2.4 Hidden-Markov-Modelle

Markov-Modelle und Spracherkennung

- Zustand \Leftrightarrow Beobachtung
- In der Sprache haben wir aber ein Kontinuum an möglichen Tokens (typischerweise Sprachsignalframes), die endlich vielen Zuständen (Phonemen) zugeordnet werden sollen
- In der Sprache sind die Zustände versteckt (hidden)

Hidden-Markov-Modelle (HMM)

- sind ein doppelter stochastischer Prozess
 - Zustandsabfolge probabilistisch
 - Jeder Zustand emittiert seine Beobachtung: Diese Emission ist ebenfalls probabilistisch
 - Zustandsfolge ist versteckt (hidden)
- Sind Markov-Modelle (1. Ordnung)
 - Wahrscheinlichkeiten für den Eintritt in den nächsten Zustand hängen nur vom aktuellen Zustand ab
- Nichtbeobachtbarkeit der Zustandsfolge hat eine Reihe von Konsequenzen
 - Sprachdekodierung mit HMMs: Anhand der Beobachtungen auf eine mögliche Zustandssequenz rückschließen (dabei wird man nie die exakte Lösung erhalten, sondern nur eine mit höchster Wahrscheinlichkeit)
 - Training von HMMs: Kennen zwar die durchlaufene Zustandsfolge, aber nicht die Zeitpunkte der Zustandsübergänge



Formale Definition:

- HMM $\lambda = (S, \pi, A, B, V)$
- $S = s_1, \dots, s_N$ - Menge aller möglichen Zustände
- $\pi: \pi(s_i) = P(q_1 = s_i)$ - Anfangsverteilung bei $t=1$
- $A = ((a_{ij})), 1 \leq i, j \leq n$ - Matrix von Übergangswahrscheinlichkeiten