Inhaltsverzeichnis

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

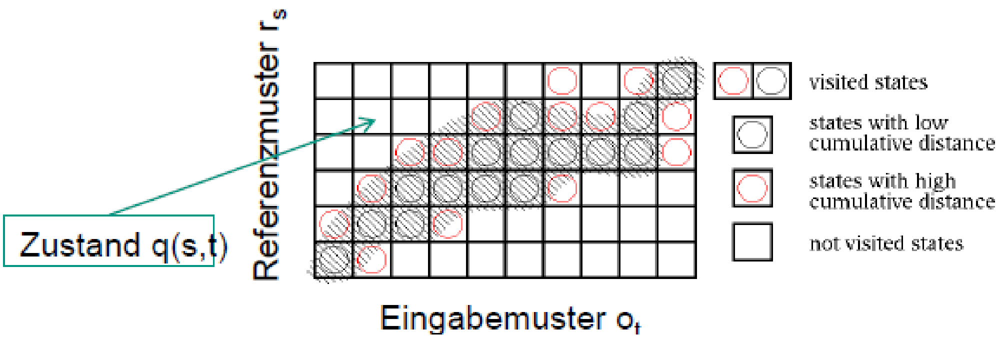
2.1 Sequenzmodellierung und State-Modelierung

- 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
- Faggot-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,..,S) beschreiben
  - Die Distanzen für Zeitpunkt t+1 berechnen sich iterativ aus denen für Zeitpunkt t und hier wird Minimierung der Distanz durchgeführt
- Benötige 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 überlicherweise:
  - 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 $q(s',t-1) -> q(s,t)$

- 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 $s_1, ..., s_N$
  - Wir beobachten eine probabilistische Zustandssequenz $O = (o_1, ..., o_T), o_i \in 1, ..., N$
  - Markov-Annahme: Wahrscheinlich, dass wir zum Zeitpunkt t in einem gewissen Zustand sind, hängt nur von vorhergehendem Zustand ab
  - Verteilung soll stationär (zeitunabhängig) sein

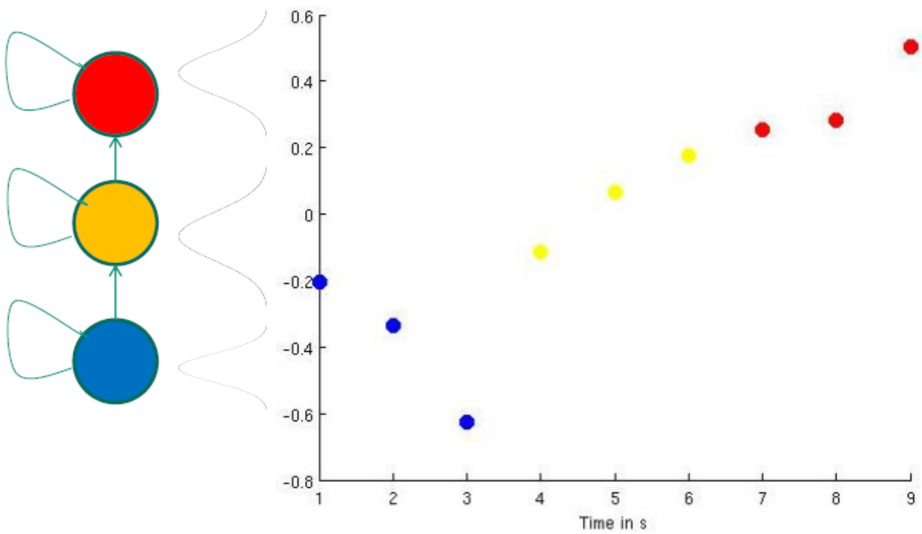
2.4 Hidden-Markov-Modelle

Markov-Modelle und Spracherkennung

- Zustand <=> 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 versteck (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:

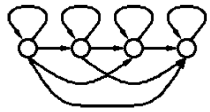
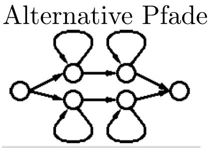
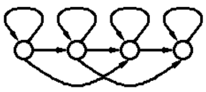
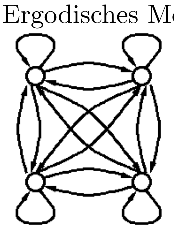
- HMM $\lambda = (S, \pi, A, B, V)$
- $S = s_1, ..., s_N$ - Menge aller möglichen Zustände
- π: $\pi(s_i) = P(q_1 = s_i)$ - Anfangsverteilung bei t=1
- $A = ((a_{ij})), 1 \leq i, j \leq n$ - Matrix von Übergangswahrscheinlichkeiten
- $B = (b_i)$ - Vektor von Ausgabewahrscheinlichkeiten, d.h. $b_i(v) = P(o_t = v|q_t = s_i)$. Dabei ist $v \in V$
- V - Vokabular, Menge der Ausgabesymbole (diskret oder kontinuierlich)

Diskrete HMMs:$V = x_1, x_2, ..., x_v$, dann sind die b_i diskrete Wahrscheinlichkeiten

Kontinuierliche HMMs: $V = R^d$, dann sind die b_i stetige Wahrscheinlichkeitsdichten

- Für die Anfangswahrscheinlichkeiten gilt $\sum_i \pi(s_i) = 1$. Vereinfachend nimmt man oft einfach: $\pi(s_1) = 1$, $\pi(s_j) = 0$, $j \neq 0$, d.h. es gibt einen ausgezeichneten Startzustand)
- Es gilt $\sum_j a_{ij} = 1$ für alle i und meistens ist $a_{ij} = 0$ für die meisten Folgezustände j

Die Struktur eines HMMs nennt man Topologie:

- Lineares Modell:
  
- Links-nach-Rechts-Modell:
  
- Alternative Pfade:
  
- Bakis model (lin. Modell + kann je 1 Zustand übersprungen werden):
  
- Ergodisches Modell (Jeder Zustand ist von jedem anderen Zustand erreichbar):
  

HMM-Theorie kennt drei Hauptaufgaben: Evaluationsproblem, Dekodierungsproblem, Optimierungsproblem

Evalutationsproblem: Berechne die Wahrscheinlichkeit der Beobachtung $P(O|\lambda)$

- Entspricht der Durchführung des DTW-Algorithmus
- Forward-Algorithmus löst dieses Problem
- Herausforderung: Wir müssen die Wahrscheinlichkeit der Beobachtung entlang aller möglichen Pfade berechnen
- Sehr aufwendig - Finden effizienter Algorithmen
- Frage: Wie summieren wir die Wahrscheinlichkeiten entlang aller möglichen Pfade effizient auf?
- Lösung: Ansatz ist wie beim dyn. Programmieren:
  - Berechne iterativ für jeden Frame und jeden Zustand die Vorwärts-Teilwahrscheinlichkeiten (forward probabilities) α
  - Dann ergibt sich eine Matrix A der Vorwärtswahrscheinlichkeiten

    $$A = \begin{array}{|c|c|c|c|} \hline \alpha_1(1) & \alpha_2(1) & \cdots & \alpha_T(1) \\ \hline \alpha_1(2) & \alpha_2(2) & \cdots & \alpha_T(2) \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \alpha_1(N) & \alpha_2(N) & \cdots & \alpha_T(N) \\ \hline \end{array}$$

  - $\alpha_t(j)$ bezeichnet die Wahrscheinlichkeit, bei gegebener Teilbeobachtung zum Zeitpunkt t im Zustand j zu sein. Zur Berechnung iteriert man über die Zeit:
  - Init: $\alpha_1(j) = b_j(o_1)\pi(s_j)$
  - Induktion: $\alpha_t(j) = b_j(o_t) \cdot \sum_{i=1..n} a_{ij}\alpha_{t-1}(i)$
  - Ergebnis: $p(o_1, o_2, ..., o_T|\lambda) = \sum_{j=1..n} \alpha_T(j)$
  - Komplexität: $O(N^2T)$

Dekodierungsproblem: Berechne die wahrscheinlichste Zustandsfolge bei der gegebenen Beobachtung $(q_1^*, ..., q_{t-1}^*, q_t^*) = arg\max_{q_1,...,q_t} P(q_1, ..., q_t|O, \lambda)$. Viterbi-Algorithmus:

- Definiere: $z_t(j) := \max_{q_1,...,q_{t-1}} P(q_1, ..., q_{t-1}, q_t = j, o_1, ..., o_t)$
- z_t ist die maximale Wahrscheinlichkeit (maximiert über alle Zustandsfolgen bis Zeitpunkt t), mit der bei der gegebenen Teilbeobachtung zum Zeitpunkt t der Zustand j erreicht wird
- Man kann $z_t(j)$ iterativ berechnen, indem man alle möglichen Vorgängerzustände betrachtet und maximiert:
  $z_t(j) = \max_i z_{t-1}(i) \cdot a_{ij} \cdot b_j(o_t)$
  $z_1(j) = \pi(s_j) \cdot b_j(o_1)$
- Ergibt sich eine Matrix Z, ähnlich wie beim Forward-Algorithmus

  $$Z = \begin{array}{|c|c|c|c|} \hline z_1(1) & z_2(1) & \cdots & z_T(1) \\ \hline z_1(2) & z_2(2) & \cdots & z_T(2) \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline z_1(N) & z_2(N) & \cdots & z_T(N) \\ \hline \end{array}$$

- Rechenaufwand: $O(N^2T)$
- Außerdem speichern wir für jeden Zustand den optimalen Vorgängerzustand:
  $r_t(j) = argmax_t(z_{t-1}(i)a_{ij})$
- Wenn alle z_t und r_t berechnet sind, kann Rückwärtszeiger r_t benutzt werden, um die gesuchte optimale Zustandsfolge zu berechnen

- Beginne beim letzten Zeitpunkt T und suche den wahrscheinlichsten Zustand. Dann gehe entlang der Rückwärtszeiger schrittweise zurück:

$$q_t^* = \begin{cases} \arg\max_j z_T(j) & \text{für } t = T \\ r_t(q_{t+1}^*) & \text{für } t < T \end{cases}$$

Optimierungsproblem: Finde ein HMM λ', so dass $P(O|\lambda') > P(O|\lambda)$ (Ein gegebenens HMM λ soll verbessert werden)

- Welche Parameter können trainiert werden: Übergangswahr., Emissionswahr., Anfangswahr.
- Probleme bei der Optimierung: unbekannt zu welchem Zeitpunkt wir in welchem Zustand sind, Wahrscheinlichkeit berechenbar dass zu Zeitpunkt t im Zustand s_j (diese Information können wir zur Gewichtung nutzen)
- Lösung des Optimierungsproblem besteht aus 2 Schritten:
- Estimation Step: Berechne die Zuordnungswahrscheinlichkeit von Trainingsdaten zu HMM-Zuständen
  - Trainingsbeobachtung $O = (o_1, ..., o_T)$
  - für jedes Sample die Wahrscheinlichkeit berechnen, dass es einem gewissen HMM-Zustand zuzuordnen ist
  - für jede Kombination von aufeinanderfolgenden Samples die Wahrscheinlichkeit berechnen, dass sie einem gewissen Zustandsübergang zuzuordnen sind
- Maximization Step: Optimiere die Parameter von Emissionswahrscheinlichkeiten, (Übergangswahrscheinlichkeiten und Anfangswahrscheinlichkeiten)

Forward-Backward-Algorithmus:

- Berechnet die Zuordnungswahrscheinlichkeiten von Trainingsdaten zu HMM-Zuständen, löst also den Expectiation Step
- Betrachten wir eine Beobachtung $o = (o_1, ..., o_T)$
- Gesucht sind 2 Parameter:
  - $Y_t(j) = P(q_t = j|o_1, ..., o_T, \lambda)$ ist die Wahrscheinlichkeit, dass der Beobachtungsvektor o_t zum Zustand j gehört
  - $\xi_t(i, j) = P(q_t = i, q_{t+1} = j|o_1, ..., o_T, \lambda)$ ist die Wahrscheinlichkeit, dass zum Zeitpunkt t im Zustand i befinden und dann in den Zustand j übergehen
- Beide Wahrscheinlichkeiten hängen von der gesamten Beobachtung o ab
- Berechnung von y und ξ
  - Forward-Algorithmus berechnet die Wahrscheinlichkeit, nach der Teilbeobachtung $(o_1, ..., o_t)$ also zum Zeitpunkt t im Zustand j zu sein
  - Backward-Algorithmus berechnet die Wahrscheinlichkeiten im Zustand j zu sein und dann die Teilbeobachtung $(o_{t+1}, ..., o_T)$ zu machen
  - Kombination der beiden ergibt Forward-Backward-Algorithmus, der $Y_t(j)$ berechnet
  - Berechne $P(q_t = j|o_1, ..., o_T, \lambda)$ druch Aufspaltung in Forward-Teil (bis Zeit t) und Backward-Teil (nach Zeit t).

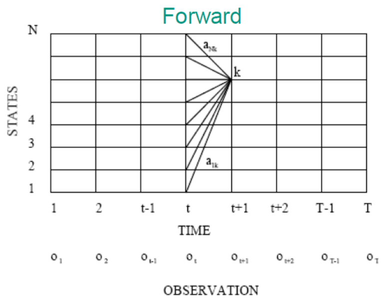
- $\alpha_t(j) := P(q_t = j, o_1, ..., o_t|\lambda)$
  $\beta_t(j) := P(q_{t+1} = j, o_1, ..., o_T|\lambda)$
  $\Rightarrow P(q_t = j|o_1, ..., o_T, \lambda) = \alpha_t(j) \cdot \beta_t(j)$. Anwendung der Bayes-Regel ergibt:

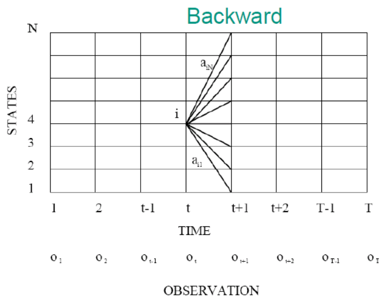
$$\gamma_t(j) = P(q_t = j \mid O, \lambda) = \frac{P(q_t = j, O \mid \lambda)}{P(O \mid \lambda)} = \frac{\alpha_t(j) \cdot \beta_t(j)}{\sum_i \alpha_t(i) \cdot \beta_t(i)}$$

  mit

$$P(O \mid \lambda) = \sum_i \alpha_t(i) \cdot \beta_t(i)$$

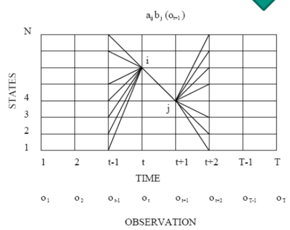
- β_t können ähnlich wie die α_t rekursiv berechnet werden, aber rückwärts:
  * Init: $\beta_T(i) = 1$ für alle Zustände i
  * Induktion: $\beta_t(t) = \sum_{j=1..n} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \;\; t = T-1, ..., 1$
  * Durch Aufsummieren: $\Rightarrow p(o_{t+1}, o_{t+2}, ..., o_T|\lambda) = \sum_{j=1..n} \beta_T(j)$





- $\xi_t(i,j)$: auch mit Forward-Backward-Algorithmus $\xi_t(i,j) = P(q_t = i, q_{t+1} = j|O, \lambda)$

$$\frac{P(q_t = i, q_{t+1} = j, O \mid \lambda)}{P(O \mid \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_l \alpha_t(l) \cdot \beta_t(l)}$$



Optimierung des HMMs:

- Trainingsdaten den HMM-Zuständen und Zustandsübergängen zugeordnet
- Das war der Expectation Step des Algorithmus
- Führe nun Maximierung der HMM-Ausgabewahrscheinlichkeit durch (Maximization Step)
- Zuordnung der Samples zu HMM-Zuständen, gegeben durch $\alpha_t(j), \beta_t(j), Y_(j)$
- Falls Emissionswahrscheinlichkeiten durch Gauss-Mischverteilungen modellieren, dann verwende EM-Algorithmus zum Training

Optimierung der Emissionswahrscheinlichkeiten:
Betrachten Emissionswahrscheinlichkeiten $B^{'} = (b_1^{'}, ..., b_N^{'})$ bei diskretem Ausgabealphabet V. Zu jedem Zustand i = 1,...,N gehört eine Verteilung b_i, so dass $b_i(v_k) \in [0,1]$ die Wahrscheinlichkeit angibt im Zustand i die Beobachtung v_k zu machen.

$$b_i'(v_k) = \frac{\sum_{t=1, o_t=v_k}^{T} P(O, q_t = i \mid \lambda)}{\sum_{t=1}^{T} P(O, q_t = i \mid \lambda)}$$

Danach werden die $b_i^{'}$ bestimmt:
Das heißt, $b_i^{'}(v_k)$ entspricht dem Anteil der Emissionen von v_k an der Gesamtzahl der "Besuche"von Zustand i.

Optimierung der Übergangswahrscheinlichkeiten:

- $sum_{t=1..T-1}\xi_t(i,j)$ ist der Erwartungswert der Anzahl der Transitionen von i nach j
- Der neue Wert $a_{ij}^{'}$ ist der Anteil der Transitionen von i nach j, normalisiert durch die Gesamtzahl der Transitionen (=Besuche) von i
- Summiere über alle Zeitpunkte t=1,...,T: $a_{ij}^{'} = \frac{\sum_t \xi_t(i,j)}{\sum_t \gamma_t(i)}$
- Der neue Wert $\pi^{'}(i)$ ist dementsprechend die Anzahl der Besuche von i zur Zeit t = 1, also $\pi_t^{'} = \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum_l \alpha_1(l)\beta_1(l)}$

Baum-Welch-Regeln:

- Alle Parameter des HMMs an die aktuelle probabilistische Zuordnung der Samples, gegeben durch $\gamma_t(i)$, angepasst. Es ergibt sich also ein neues HMM: $\lambda^{'} = (S, \pi^{'}, A^{'}, B^{'}, V)$
- Dieser Algorithmus wird, wie es für EM-Algorithmen typisch ist, so lange iterativ wiederholt, bis ein Abbruchkriterium erfüllt ist
- Die Regeln der letzten "Folienßind die Baum-Welch-Regeln

Training von HMMs mit Viterbi:

- Viterbi, der immer nur maximale Wahrscheinlichkeiten betrachtet, geht viel schneller als Forward-Backward-Algorithmus
- In der Spracherkennung ist Viterbi-Training Standard

3 12 - Spracherkennung

3.1 Schall als Luftdruckwelle

Was ist Schall?

- Druckwelle, die von einem vibrierenden Objekt erzeugt wird
- Vibration überträgt sich auf Partikel des umgebenden Trägermediums (z.B. Luft) - Energietransport über Medium findet statt
- Partikel parallel zur Ausbreitungsrichtung der Welle - spricht man von Longitudinalwelle
- Longitudinalwelle besteht aus Kompressionen (Verdichtungen) + Rarefaktionen (Verdünnungen) der Luft

- Lässt sich durch Sinusfkt. beschreiben
- Amplitude entspricht der Dichte der Luft an der betreffenden Stelle
- Ausbreitungsgeschwindigkeit in der Luft: 331,5 + 0,6 T m/s, T = Temperatur in C

Messung der Schallintensität

- leseste hörbare Ton moduliert den Luftdruck um etwa $10^{-5}Pa$, Schmerzgrenze: $100 = 10^2 Pa$
- Wird in Dezibel [dB] gemessen (dB ist Verhältnis von zwei Schallintensitäten)
- Schalldruckpegel (sound pressure level, SPL) misst den absoluten Schalldruck in dB
- Referenzgröße P_0 ist die Hörschwelle von $2 \cdot 10^{-5}Pa$ $SPL = 20 \cdot log_{10}(P/P_0)$

3.2 Der menschliche Sprachproduktionsapparat

Sprachproduktionsapparat

- Sprache besteht aus Luftdruckwellen - diese werden von Mund und Nase ausgestoßen
- Erzeugung dieser Wellenform besteht aus 2 Schritten
  - Stimmbänder und Kehlkopf erzeugen eine Grunderregung
  - Der Dokaltrakt (Mundhöhle, Nasaltrakt) wirkt als ein Filter auf diese Grunderregung und moduliert sie
- Grunderregung kann eine periodische Schwingung oder aperiodisches Rauschen sein
- Häufige Annahme: Erregung und Filter sind unabhängig

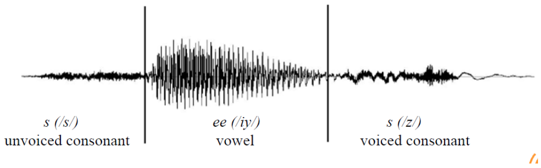
Grundfrequenz
Betrachten wir zuerst den Fall, dass die Stimmbänder sich öffnen und schließen und so die Grunderregung erzeugen

- Periodisches öffnen und schließen der Stimmbänder erzeugt periodische Schwingung (Grunderregung)
- Dauer eine Periode hängt von Länge und Anspannung der Stimmbänder und dem von der Lunge erzeugten Luftdruck ab
- Die Periode kann vom Sprecher in gewissen Grenzen moduliert werden, um die Tonhöhe (pitch) zu modulieren
- Öffnungszyklus der Stimmbänder:
  - Stimmbänder widerstehen dem Lungenluftdruck
  - Unter immer stärkerem Druck öffnen sich die Stimmbänder
  - Wenn der Druck wieder gering ist fallen die elastischen Stimmbänder wieder in die Ausgangsposition
- Anzahl dieser Öffnungsvorgänge pro Sekunde als Grundfrequenz der Sprache f_0

- Variiert von 60 Hz (große Männer) bis 300 Hz (Kinder)
- Grundfrequenz bestimmt die Periode für die höherfrequenten harmonischen Schwingungen des Vokaltrakts

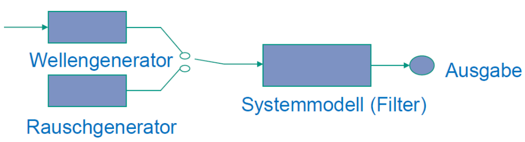
Stimmhafte vs. stimmlose Phoneme

- Laute sind stimmhaft, wenn während der Artikulation eine Vibration der Stimmbänder vorliegt
- Andernfalls sind die Stimmbänder geöffnet, und die Grundanregung des Lautes ist ein Rauschen mit gewissen chaotischen Eigenschaften
- Beispiel für beide Lautarten: Wellenform des engl. Wortes sees



Das Quelle-Filter-Modell

- Wellengenerator: Periodische Schwingung der Stimmbänder
- Rauschgenerator: Luftstrom bei stimmlosen Phonen
- Systemmodell: Filtereigenschaft des Vokaltraktes (Mundraums)



3.3 Akustische Phonetik und Phonologie

Phonetik und Phonologie

- Phonetik: Studium der Produktion, Klassifikation und Transkriptiopn von Sprachlauten -> Fokus liegt auf der akustischen Realisierung der Sprachlaute
- Phonologie: Studium der Verteilung und Struktur von Lauten in einer Sprache -> Hauptziel ist es, übergreifende Charakteristiken von Sprachlauten zu finden

Phonetik

- Artikulatorische Eigenschafen
  - Laute können neben stimmhaft/stimmlos nach artikulatorischen Eigenschaften unterschieden werden
  - Unterscheidung erfolgt entsprechend der Anatomie der wichtigen Artikulatoren und ihrer Position im Vokaltrakt

  - Hauptkomponenten des menschl. Sprachproduk.apparats: Lungen, Trachea (Luftröhre), Pharynx (Rachen), Nasenhöhle und Mundhöhle. (Rachen + Mundhöhle = Vokaltrakt, Nasenhöhle = Nasaltrakt)
  - Innerhalb des Vokaltrakts sind Stimmbänder, weicher Gaumen (Velum), harter Gaumen (Palatum), Zunge, Zähne und Lippen

- Benennung von Sprachlauten
  - Nasallaute:Luftstrom hauptsächlich durch Nase, gesenktes Velum (/n/)
  - Orallaute: Luftstrom durch Mund, Velum verschließt Nasalraum
  - Stoplaute (Plosive):Vokaltrakt kurzzeitig vollst. verschlossen (/p/, /b/)
  - Frikative: Vokaltrakt teilw. verschlossen, Reibung entsteht (/f/)
  - Approximanten: Vokaltrakt verengt, keine Reibung (/j/)
  - Labial: Artikulationsort Lippen (/b/, /w/)
  - Dental: Artikulation Zunge an Zähnen
  - Alveolar: Alveole (Zahndamm) aktiv
  - Palatal: Harter (vorderer) Gaumen aktiv
  - Velar: Weicher (hinterer) Gaumen aktiv
  - Glottal: Glottis, Stimmbänder aktiv (z.B. Be-amte)