ML for Speech Processing (includes Paralinguistic Speech Processing)

Dr. Mathew Magimai Doss

December 15, 2022

Outline

Outline

Machine learning

- Learning is an essential part of living
- Learning typically means changing/adapting to be better, as per a given criterion, when a similar situation arrives
- Challenge lies in generalizing to new or unobserved situations

Challenge

- Learning needs training data. We have access to only a finite amount of training data.
- Variabilities in the data
  - Language level: isolated words, sentence, spoken language, read speech, spontaneous speech, dialect ...
  - Speaker level: gender, adult versus child, dialect, age, accent, impaired versus unimpaired (pathological speech), emotion, mood, stress ....
  - Noise
    - Convolutive: recording/transmission condition, reverberation
    - Additive: recording environment, transmission
    - Lombard effect: speaker level variability in noisy environment
- Depending upon the task, a few variabilities are desirable or of interest while others are undesirable or of not interest.

Types of learning

1. Supervised learning
   - Training data is labelled. For example, for a frame of feature vector or a sequence feature vectors we have a "class" label associated to it.
2. Reinforcement learning
   - Training data has partial labels/targets. For example, did a robot carry out the desired action or not?
3. Unsupervised learning
   - Training data does not contain class labels or targets. But, often there is a hidden goal associated with the task. For example, data clustering tasks have a hidden goal such as, minimization of a distance function or maximization of likelihood.

Statistical Pattern Recognition

- Classification
  - Static classification
  - Sequence classification
- Detection: can be regarded as a two class classification problem
- Regression: relation between two variables, namely, measured variable and explanatory variable

Three Key Statistical Rules

1. Bayes's rule:

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A)$$

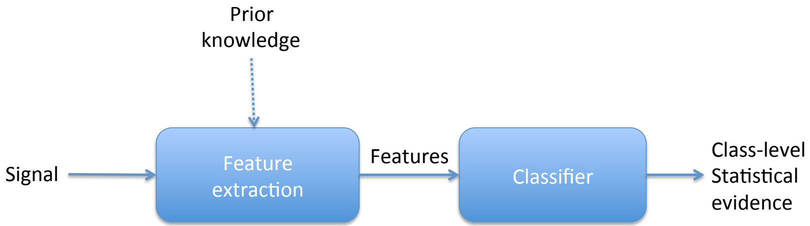
2. If B_k $(k = 1, \ldots, K)$ are mutually exclusive and collectively exhaustive ($\sum_{k=1}^{K} P(B_k) = 1$)

$$P(A) = \sum_{k=1}^{K} P(A, B_k)$$

3. Gibbs sampler:

$$P(B_1, \ldots, B_k, \ldots, B_K) = \prod_{k=1}^{K} P(B_k | B_{k-1}, \ldots, B_1)$$

Static Classification



Speech

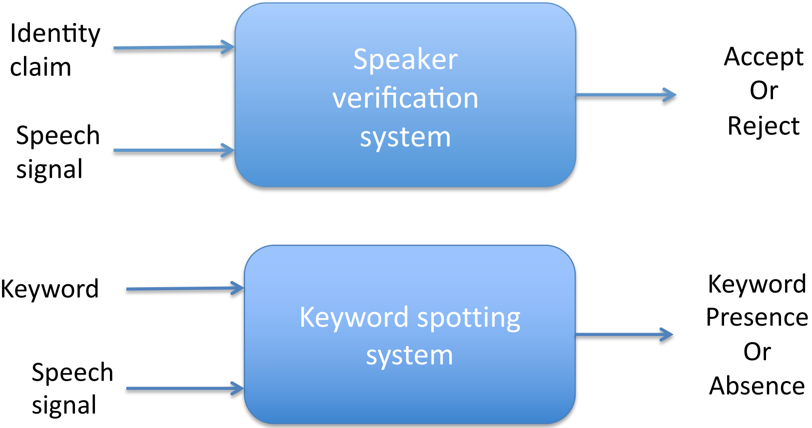
Phoneme
Gender
Speaker
Emotion..

Face image

Person
Age
Gender
Emotion..

Sequence classification

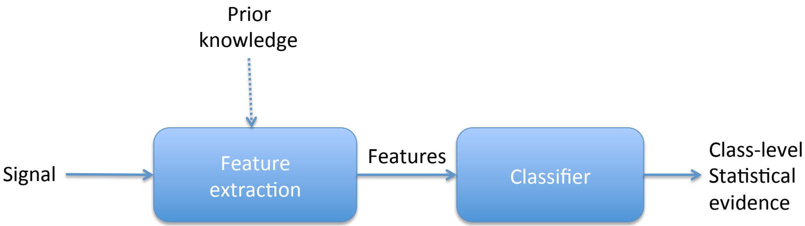
Detection



Hypothesis testing problem

Outline

Static Classification



Speech

Phoneme
Gender
Speaker
Emotion..

Face image

Person
Age
Gender
Emotion..

Theoretical formulation

$$P(C_k|x_m,\Theta) = \frac{p(x_m|C_k,\Theta)\cdot P(C_k|\Theta)}{p(x_m|\Theta)}\ \forall k \in \{1,\cdots K\} \qquad (1)$$

- $P(C_k|x_m,\Theta)$: Posterior probability of class C_k
- $p(x_m|C_k,\Theta)$: Likelihood of class C_k
- $P(C_k|\Theta)$: Prior probability of class C_k
- $p(x_m|\Theta) = \sum_{j=1}^{K} p(x_m|C_j,\Theta)\cdot P(C_j|\Theta)$: Observation likelihood
- Θ: parameters of the statistical model
- $0 \leq P(C_k|x_m,\Theta) \leq 1$ and $\sum_{k=1}^{K} P(C_k|x_m,\Theta) = 1$
- $0 \leq P(C_k|\Theta) \leq 1$ and $\sum_{k=1}^{K} P(C_k|\Theta) = 1$

Generative approach

- Estimate or model $p(x_m|C_k)$ by a probability density function
  - Gaussian or Normal distrution

$$
\begin{aligned}
p(x_m|C_k,\Theta_k) &= N(x_m,\mu_k,\Sigma_k) \\
&= \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{(x_m-\mu_k)^t\Sigma_k^{-1}(x_m-\mu_k)}{2}\right) \\
\approx & \quad \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}\sigma_k^d} \exp\left(-\frac{1}{2}\left(\frac{x_m^d-\mu_k^d}{\sigma_k^d}\right)^2\right)
\end{aligned}
$$

  - Gaussian mixture models (GMM)

$$
p(x_m|C_k,\Theta_k) = \sum_{j=1}^{J} c_k^j \cdot N(x_m,\mu_k^j,\Sigma_k^j)
$$

- Estimate prior probability $P(C_k)$ (typically done through counting)
- Apply Eqn. (1)
- Θ: means, variance and Gaussian weights (in the case of GMMs)

EM algorithm for GMMs (1)

$$p(x_m) = \sum_{j=1}^{J} c_j p(x_m|\mu_j, \Sigma_j)$$

with $c_j = P(G_j)$ (Weight fot Gaussian j).

Estimation step:

$$P(G_j|x_m) = \frac{c_j p(x_m|\mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_j c_j p(x_m|\mu_j^{(t)}, \Sigma_j^{(t)})}$$

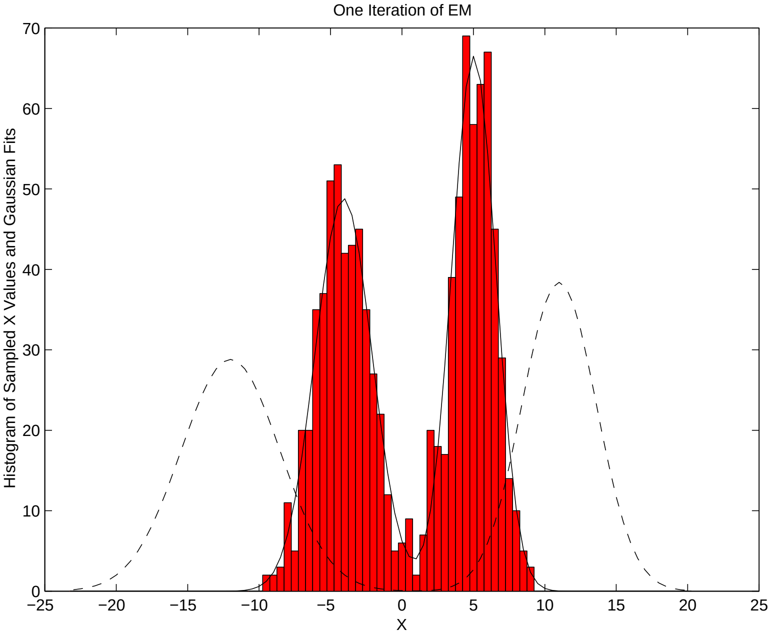
Maximization step:

$$\mu_j^{(t+1)} = \frac{\sum_{m=1}^{n=M} x_m P(G_j|x_m)}{\sum_{m=1}^{m=M} P(G_j|x_m)}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{m=1}^{m=M} P(G_j|x_m)(x_m - \mu_j^{(t+1)})(x_m - \mu_j^{(t+1)})^T}{\sum_{m=1}^{m=M} P(G_j|x_m)}$$

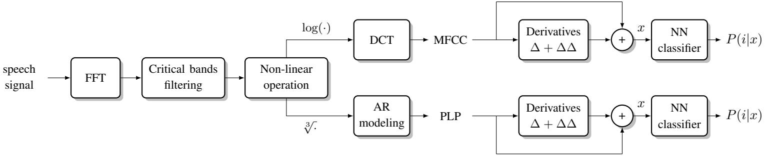
$$c_j^{(t+1)} = \frac{1}{M} \sum_{m=1}^{m=M} P(G_j|x_m)$$

EM algorithm for GMMs (2)



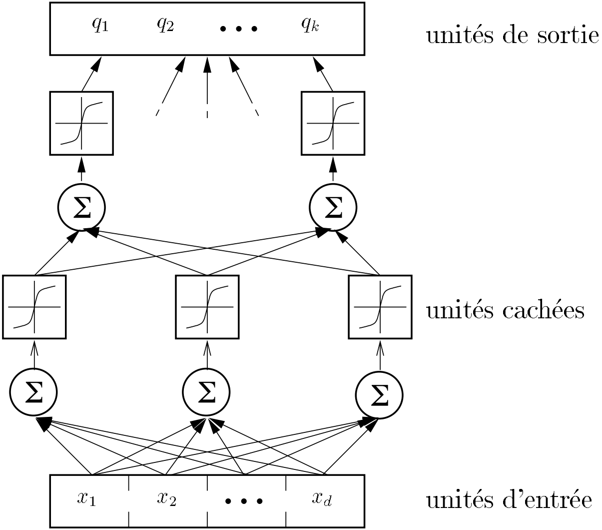
Discriminative approach

- Artificial neural network trained with one hot encoding of target and cross entropy error function (or mean square error function) can directly estimate $P(C_k|x_m, \Theta)$



- Support vector machines (estimation of posterior probability of class not trivial, see Platt's method)
- Θ: parameters of the artificial neural networks (weights and biases) or support vector machines

ANN: Multilayer perceptrons



Typical multilayer perceptron (MLP) architecture, each unit approximating a perceptron.

ANN Training

Supervised training: Input vector sequence:

$$X = \{\mathrm{x}_1, \ldots, \mathrm{x}_n, \ldots, \mathrm{x}_M\}$$

Desired output sequence associated with X:

$$D = \{d(\mathrm{x}_1), \ldots, d(\mathrm{x}_m), \ldots, d(\mathrm{x}_M)\}$$

and $d(\mathrm{x}_m) = (d_1(\mathrm{x}_m), \ldots, d_k(\mathrm{x}_m), \ldots, d_K(\mathrm{x}_m))^{\mathrm{T}}$
In classification mode:

$$d_k(x_m) = \delta_{k\ell} \quad \text{if} \quad \mathrm{x}_m \in C_\ell$$

Parameters Θ: weights and biases

Training Criteria

- Mean Square Error:

$$\underset{\{\Theta\}}{\operatorname{argmin}} \; E = \frac{1}{2} \sum_{m=1}^{M} \sum_{k=1}^{K} \left[g_k(x_m, \Theta) - d_k(x_m) \right]^2$$

- Entropy or relative entropy:

$$\begin{aligned} \underset{\{\Theta\}}{\operatorname{argmin}} \; E_e \quad = \quad & \sum_{m=1}^{M} \sum_{m=1}^{M} \left[d_k(x_m) \log \frac{d_k(x_m)}{g_k(x_m, \Theta)} \right. \\ & \left. + (1 - d_k(x_m) \log \frac{1 - d_k(x_m)}{1 - g_k(x_m, \Theta)} \right] \end{aligned}$$

$g_k(x_m, \Theta)$ denotes the output of the neural network.

Error back propagation training

Minimization of E (or E_e) in the parameter space Θ (weights + biases)

$$\frac{\bigtriangledown E}{\bigtriangledown \Theta} = 0$$

Done via a gradient procedure:

$$\triangle w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}}$$

α denotes learning rate

Adjust w_{ij} based on $\triangle w_{ij}$.

Offline Error Back-Propagation

Initialize network at random; choose "large" learning rate
Until convergence = true
    For m=1 to M
        Forward computation of $g_k(\mathrm{x}_m, \Theta)$
        Error calculation and global error update
        Error backward propagation
        and compute local $\delta\Theta(\mathrm{x}_m)$
    Θ **update** $= \sum_{m=1}^{M} \delta\Theta(\mathrm{x}_m)$
    If error (on cross-validation set) decreases
        save new parameters
    Otherwise, don't save new parameters
        and decrease learning rate
If learning rate $<$ threshold then convergence = true;

Online Error Back-Propagation

Initialize network at random
Choose "large" learning rate; convergence = false
Until convergence = true
    For m=1 to M (or something else)
        Pick x_m at random
        Forward computation of $g_k(x_m, \Theta)$
        Error calculation
        Error backward propagation and Θ **update**
    If error (on cross-validation set) decreases
        save new parameters
    Otherwise, don't save new parameters
        and decrease learning rate
If learning rate $<$ threshold: convergence = true;

In practice: Mini-batch training, a combination of offline and online error back propagation.

Cross-validation training (1)

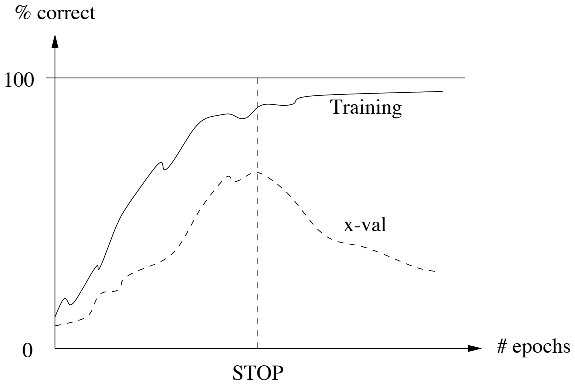
After each MLP training epoch:

1. check recognition performance on independent data set
2. stop training if rec performance starts to decrease and learning rate below a given threshold

Remarks: there are other solutions like

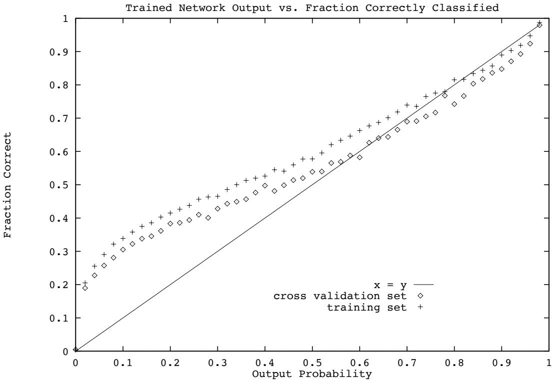
- Forcing small weights
- "Optimal Brain Damage"
- Regularized training (Bayesian approach)

Cross-validation training (2)



Example of crossvalidation training. x-val represents the crossvalidation data on which classification performance is regularly checked. Training is stopped when performance on x-val data reaches the maximum.

Interpretation of ANN output $g_k(\mathrm{x}_m, \Theta)$



ANN output $g_k(\mathrm{x}_m, \Theta)$ is an estimate of posterior probability of class C_k, i.e. $P(C_k|\mathrm{x}_m)$.

Decision making

- Maximum likelihood

$$C^*_{mle} = \arg\max_k p(X|C_k, \Theta)$$

$$p(X|C_k, \Theta) = \prod_{m=1}^{M} P(x_m|C_k, \Theta) \text{(Assuming i.i.d)}$$

- Maximum aposteriori probability

$$C^*_{map} = \arg\max_k p(C_k|X, \Theta)$$

$$P(C_k|X, \Theta) = \frac{1}{Z} \cdot \prod_{m=1}^{M} P(C_k|x_m, \Theta) \text{(Assuming i.i.d)}$$

or

$$P(C_k|X, \Theta) = \frac{1}{Z} \cdot \sum_{m=1}^{M} P(C_k|x_m, \Theta)$$

Z is a normalization factor.

Better to perform computation using logarithm to avoid underflow issues.

Outline

Sequence classification

Statistical ASR

Given feature observation $X = \{x_1, \cdots x_m, \cdots x_M\}$ predict the most probable word sequence $W^* = \{w_1^* \cdots w_n^* \cdots w_N^*\}$

$$
\begin{aligned}
W^* &= \arg\max_{W_i \in \mathcal{W}} P(W_i | X, \Theta) \\
&= \arg\max_{W_i \in \mathcal{W}} \frac{p(X|W_i, \Theta_a) \cdot P(W_i|\Theta_l)}{p(X|\Theta)} \\
&= \arg\max_{W_i \in \mathcal{W}} p(X|W_i, \Theta_a) \cdot P(W_i|\Theta_l),
\end{aligned}
$$

where W_i denotes a word hypothesis, $\mathcal{W}$ denotes a set of word hypotheses and $\Theta = \{\Theta_a, \Theta_l\}$

- Acoustic modeling: estimation of $p(X|W_i, \Theta_a)$ using hidden Markov models (HMMs)
- Language modeling: estimation of $P(W_i|\Theta_l)$ using discrete Markov models (DMMs)

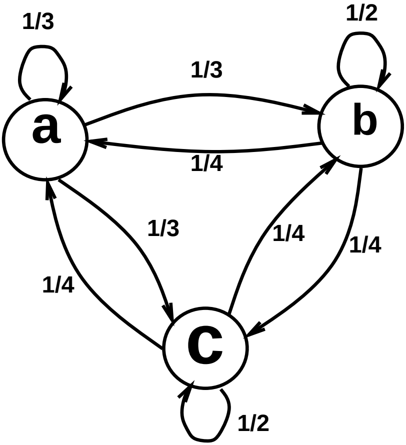
Discrete Markov Model (DMM)

- Stochastic finite state automaton
- *MM* built up from states q_ℓ from a set of classes (states) $\Omega = \{\omega_1, \ldots, \omega_k, \ldots, \omega_K\}$
- q^n particular state of M visited at time n,
- $q_\ell^n \; \equiv \; \{q^n = q_\ell\}$, $q_l \in \Omega$
- *MM* is defined by topology, i.e., how the states are connected
  See HMM lab exercise 1
- Parametrized by:

$$\begin{aligned} P(q_\ell^n | q_k^{n-1}, q_j^{n-2}, \ldots) &\simeq P(q_\ell^n | q_k^{n-1}) \quad (\textit{1st order Markov}) \\ &\simeq P(q_\ell | q_k) = P_{k\ell} \quad (\textit{time independent}) \end{aligned}$$

Note: $q_l, q_k \in \Omega$
Transition probability matrix: $A = \{P_{k,\ell}\}$.

DMM (2)



Example of fully connected discrete Markov model with $\Omega = \{a, b, c\}$. For example, in case of weather model: "a" = "cloudy", "b"= "rainy" and "c" = "sunny"

Typical Problems (1)

- **Probability of a particular path**

$$P(Q|MM) = P(q^1|q_i^0)P(q^2|q^1)\dots P(q^n|q^{n-1})\dots P(q^N|q^{N-1}) = \prod_{n=1}^{N} P(q^n|q^{n-1})$$

One way to estimate $P(W_i|\Theta_l)$, e.g.

$$P(W^*|\Theta_l) = \prod_{n=1}^{N} P(w_n^*|w_{n-1}^*)$$

- **State duration distribution**

Probability to stay in state q_i for *exactly* d time steps?

$$Q = \{q_i^0, q_i^1, q_i^2, \dots, q_i^d, q_j^{d+1}\}, \text{ with } j \neq i$$

and:

$$P(Q|MM) = (P_{ii})^{d-1}(1 - P_{ii})$$

See HMM lab exercise 1

Typical Problems (2)

- **Probability to go from state q_i to q_j in N steps**

$$P(q_j^N|q_i^0) = \sum_{n=0}^{N} \sum_{\ell=1}^{L} P(q_j^N, q_\ell^n|q_i^0)$$

Defining

$$\alpha(\ell, n) = P(q_\ell^n|q_i^0, N)$$

We have:

$$\alpha(\ell, n+1) = \sum_k \alpha(k, n) P_{k\ell}$$

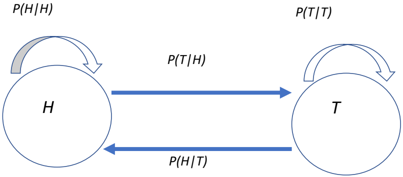
$$P(q_j^N|q_i^0) = \alpha(j, N)$$

Typical Problems (3)

- **Probability of best path of length N between q_i and q_j**
  If $\overline{P}(k,n)$ is probability of best path to go from q_i to q_k in n steps:
  $$\overline{P}(\ell, n+1) = \max_k \overline{P}(k,n) P_{k\ell}$$
  and
  $$\overline{P}(q_j^N | q_i^0) = P(j, N)$$
  Generalization:
  $$A^n(i,j) = P(q_j^n | q_i^0)$$

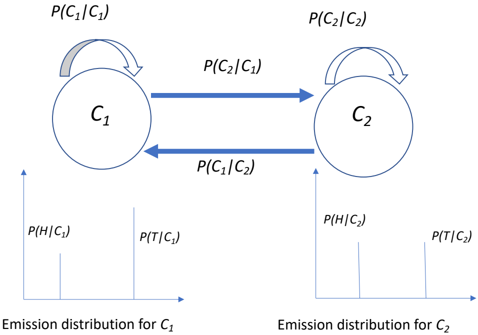
From DMM to HMM (1)

- Coin tossing and only the results of each coin toss i.e. Heads (H) or Tails (T) is revealed
  $H\ H\ T\ T\ T\ H\ H\ H\ H\ T\ T\ldots$
- 1-coin model: DMM with two states $\Omega = \{H, T\}$ and all transition probabilities are equal to 0.5.



From DMM to HMM (2)

- 2-coin model: HMM with two states $\Omega = \{C_1, C_2\}$; each state associated with $P(H|C_i)$ and $P(T|C_i)$ $i \in \{1, 2\}$; and the transition probability reflects the probability of choosing one of the (possibly biased) coins



Estimation of $P(X|W_i, \Theta_a)$ using HMMs

$$
\begin{aligned}
P(X|W_i, \Theta_a) &= \sum_{Q \in W_i} P(X, Q|W_i, \Theta_a) \\
&= \sum_{Q \in W_i} P(X|Q, W_i) P(Q|W_i),
\end{aligned}
$$

where $Q = \{q_1, \cdots q_m, \cdots q_M\}$ denotes sequence of HMM states.
After i.i.d and first order Markov assumption

$$
\begin{aligned}
P(X|W_i, \Theta_a) &= \sum_{Q \in W_i} \prod_{m=1}^{M} p(x_m|q_m) \prod_{m=1}^{M} P(q_m|q_{m-1}) \\
&= \sum_{Q \in W_i} \prod_{m=1}^{M} p(x_m|q_m) P(q_m|q_{m-1}) \text{ Full likelihood} \\
&\approx \max_{Q \in W_i} \prod_{m=1}^{M} p(x_m|q_m) P(q_m|q_{m-1}) \text{ Viterbi approx.}
\end{aligned}
$$

For the sake of simplicity, Θ_a and W_i are dropped in the latter equations.
$p(x_m|q_m)$ is referred to as emission likelihood and $P(q_m|q_{m-1})$ state transition probabilities. *See HMM lab exercises 3 and 4.*

Estimation of emission likelihood

- Gaussians and Gaussian-Mixtures

$$p(x_m|q_m = k, \Theta_a) = N(x_m, \mu_k, \Sigma_k)$$
$$= \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{(x_m - \mu_k)^t \Sigma_k^{-1}(x_m - \mu_k)}{2}\right)$$

OR :

$$p(x_m|q_m = k, \Theta_a) = \sum_{j=1}^{J} c_k^j N(x_m, \mu_k^j, \Sigma_k^j)$$

- Artificial neural networks: estimate scaled-likelihood $p_{sl}(x_m|q_m = k, \Theta_a)$

$$p_{sl}(x_m|q_m = k, \Theta_a) = \frac{p(x_m|q_m = k, \Theta_a)}{p(x_m|\Theta_a)} = \frac{P(q_m = k|x_m, \Theta_a)}{P(q_m = k|\Theta_a)}$$

Outline

Detection

$$\frac{p(X|\text{genuine speaker})}{p(X|\text{impostor})} \geq \delta_{asv}$$

Challenge: modeling or estimation of $p(X|\text{impostor})$

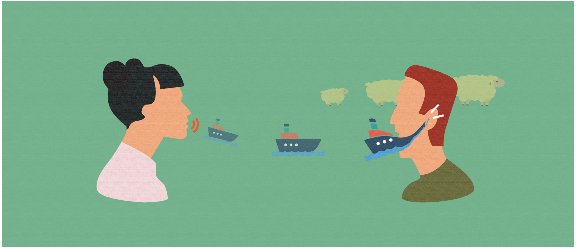
$$\frac{p(X|\text{keyword})}{p(X|\text{not keyword})} \geq \delta_{kws}$$

Challenge: modeling or estimation of $p(X|\text{not keyword})$

Two types of error: False negative, False positive

Outline

Paralinguistics



States and traits manifested in speech communication

- long-term: age, gender, accent, personality ...
- medium-term: mood, sleepiness, intoxication, social role ...
- short-term: emotion, voice quality ...
- speech and language pathologies

Paralinguistics $\propto$ speech, language, physiology, sociology, psychology, health ....

Paralinguistics

Recent examples from workshops within the community:
Interspeech Computational Paralinguistics Challenge [1]
ACM Audio/Visual Emotion Challenge Workshop [2]

- Styrian dialects: northern, urban, eastern
- native language: 11 languages (ger, fra, ita, spa, hin)
- Degree of nativeness: 0 (no foreign accent) ... 6 (strong foreign accent)
- emotions: anger, empathic, neutral, positive, rest
- alcohol intoxication: no/yes
- Parkinson's disease: intelligibility 0 - 100
- sleepiness: 1 - 9
- Alzheimer's disease: no/yes

---

[1] http://www.compare.openaudio.eu/
[2] https://sites.google.com/view/avec2019

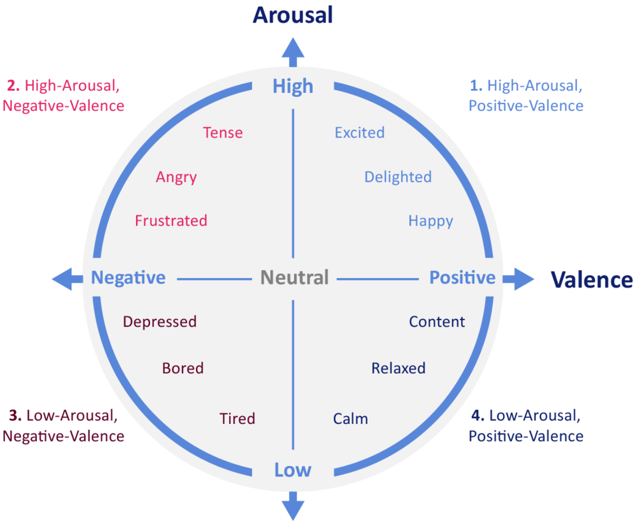
Example application areas

Challenges in Paralinguistics (1)

Data Labeling

- Establishing a ground-truth is not a trivial task
  $\rightarrow$ labels can be based on other fields
- humans are typically used to label/rate the data
  $\rightarrow$ how to select human raters?
- human ratings can differ
  $\rightarrow$ need to measure inter-rater agreement: see Cohen's kappa
- labels are typically assigned to a whole utterance or recording
  (may refer to only some part of recording)

Emotion class labels



Sleepiness rating

Rating	Verbal descriptions
1	Extremely alert
2	Very alert
3	Alert
4	Fairly alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy, but no effort to keep alert
8	Sleepy, some effort to keep alert
9	Very sleepy, great effort to keep alert, fighting sleep

also see: Likert scale

Challenges in Paralinguistics (2)

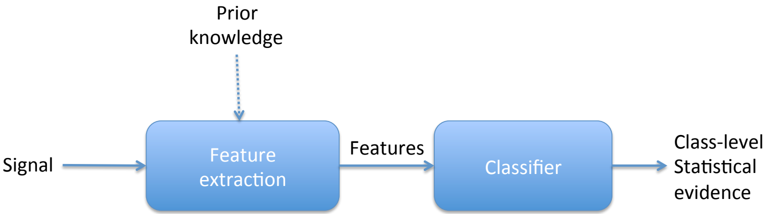
Features

- which features are relevant?
- relevant information is present at short-term (segmental) level or/and long-term (supra-segmental) level?
- how to use prior knowledge or feature selection?
- how to obtain short-term or long-term representations?

Challenges in Paralinguistics (3)

- Data scarcity
  - Collecting large data sets is not an easy task
    - simulated versus real e.g., acted emotion versus real emotion
    - ethical restrictions e.g. in the case of health care
    - privacy issues
    - ...
  - transfer learning from resource-rich task (e.g. with deep learning)
  - data augmentation - risk of obliterating important information
- Cross-database, cross-lingual and cross-cultural generalisation is not trivial.

Pattern recognition



Speech

Phoneme
Gender
Speaker
Emotion..

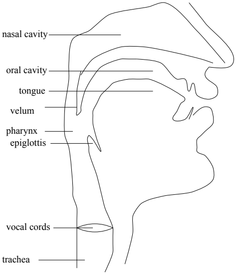
Face image

Person
Age
Gender
Emotion..

Speech Analysis

Various aspects of speech production and perception:

- Respiration
  - pause patterns, speech rate, loudness through excitation strength
- Phonation
  - analysis of pitch (jitter and shimmer)
- Articulation - changes in vocal tract shape
  - use source-system decomposition
  - analysis of voiced speech (vowel formants)



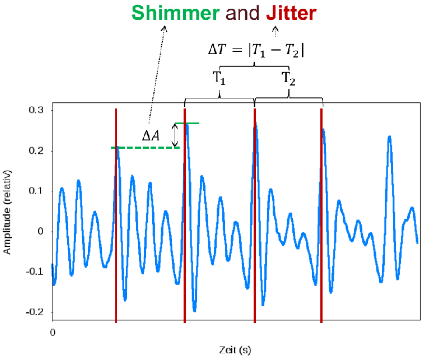
Respiration and pause patterns

- pitch
- loudness
- statistics of silent regions
- statistics of unvoiced regions per second (pseudo-rate)

Phonation and voice-quality features

Measuring irregular phonation

- Jitter: deviations of pitch from perfect periodicity
- Shimmer: deviation of energy of signal



- Harmonics-to-Noise Ratio (HNR):
  $HNR = 10 * \log \frac{ACF[T0]}{ACF[0] - ACF[T0]}$
  ACF denotes autocorrelation function and T_0 denotes pitch period.

Articulation features

Measuring vocal tract shaping

- Mel Frequency Cepstral Coefficients (MFCCs)
- Perceptual Linear Prediction Coefficients (PLPs)
- Vowel formant frequencies
- Spectral slopes and ratios
- ...

Feature sets

Traditional approach to paralinguistic problems:
Use large quantities of short-time acoustic features

Several version of feature sets, easily extractable with openSMILE [3]

- openSMILE feature set: 6773 LLDS and functionals
- eGeMAPS feature set: 88 LLDS

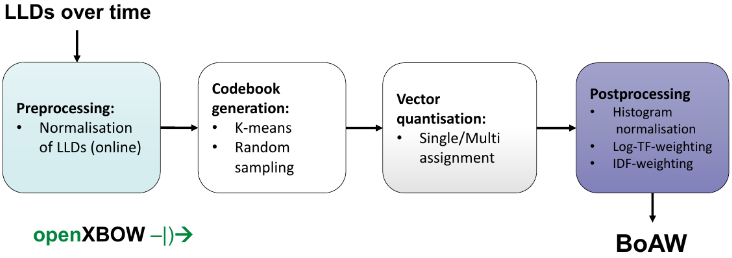
How to classify frame-level acoustic features?

1. Majority vote of frame-level feature vectors
2. Summarize frame-level feature vectors into per-utterance vectors

[3] https://www.audeering.com/opensmile/

Per-utterance representations

- acoustic features are summarized in statistical functionals (e.g. means, moments, extrema, percentile, slopes, regression lines, max, min)
- bag-of-audio–word (BoAW) representations derived from bag-of-word vectors used in NLP [4]



- robust, time-invariant, non-reconstructable (good for privacy)

[4] https://github.com/openXBOW/openXBOW

Classification

Choice of classifier is dictated by amount of data available

- k-Nearest Neighbors (kNN)
  - works well with small data sets
- Support Vector Machine (SVM)
  - works well with small data sets and high dimensions
- Random forest
  - works well with small data sets and high dimensions
- Boosting
  - works well with small data sets and high dimensions
- Neural Networks (NN)
  - May not suit well for small data sets

In some problems, classification is replaced by regression.

Evaluation

Common evaluation metrics:

- Regression
  - Pearson's correlation r: linear correlation coefficient
  - Spearman's cross-correlation ρ: non-parametric correlation coefficient
- Classification
  - Unweighted Average Recall (UAR): consider performance across non-balanced class distributions

Common evaluation schemes

- N-fold cross-validation
- leave-one-speaker-out cross-validation

Thank you for your attention!

Dr. Mathew Magimai Doss

Idiap Research Institute, Martigny, Switzerland