Dialogue systems

Speech Recognition

Luděk Bártek

Laboratory of Searching and Dialogue, Fakulty of Informatics, Masaryk University, Brno

spring 2023

Speech Recognition

- Continuous speech recognition – transforms continuous speech to a textual form.
- Command recognition.
- Recognition principle:
  1. using a short term signal analysis acquire the feature vector,
  2. try to classify the signal using the vector from previous step.

Command Recognition

- Used to recognize either commands or words (commands) distinctly separated by silence on both ends.
- There is no problem to identify the start and the end of the word in continuous utterance.
- Usually user depended systems.
  - There is a need to train the recognizer,
  - limited size of used vocabulary.
- Command recognition problems:
  - Identifying the start and the end of the command:
    - how to distinguish a noise and sibilants,
    - distinguishing a random sound excitation (click, tapping, ...) and plosives including a short pause,
    - possible infra sounds,
    - ...

Command Recognition

Classifiers Types

- DTW based classifiers.
  - Tries to find maximum correspondence between recognized word na words in database.
- Statistical methods based classifiers – speech modelling using Hidden Markov Models:
  - simulates the speech generation process.
- Two phase classifiers:
  1. speech segmentation to segments and phonetic decoding of segments
  2. word recognition based on decoded segments.
- Artificial Neural Networks based solutions – see:
  - Hinton, O., Teh - A Fast Learning Algorithm for Deep Belief Nets, in Neural Computation, 2006
  - Bengio, L., Popovici, L. - Greedy Layer-Wise Training of Deep Networks, in NIPS' 20016
  - Speech recognition - Lecture 14: Neural Networks

Dynamic Time Warping (DTW)

- Method is used to compare two series of numbers – two parts of speech (two words).
- Input:
  - acoustic vectors sequence acquired using some of the short term signal analysis methods
  - database of acoustic vectors for recognized words.
- Output – recognized word or command.

DTW

Basic principle

- Let's create database of recognized words (reference sequences of acoustic vectors)
  - Usually several sequences for each word, corresponding to several manners of word pronunciations.
- Recognized word is transformed into the corresponding acoustic vectors sequence.
- Using DTW we find the reference sequence with maximum conformity.

DTW

Formalization

- DTW algorithm search for parametrizations f and g:

$$f, g : i = f(k), j = g(k), k \in < 1, K >$$

that minimizes expression:

$$D(A, B) = \sum_{i=1}^{K} d(a_{f(i)}, b_{g(i)})$$

  - d – acoustic vectors distance (i.e.. Euklid's metric)
  - $a_{f(i)}$, $b_{g(i)}$ – reference and recognized word/command.

DTW

Constraints

- f,g – non-descending function
- Local coherence and steepness:
  - $0 \leq f(k) - f(k-1) \leq I^*$
  - $0 \leq g(k) - g(k-1) \leq J^*$
  - mostly $I^*, J^* = 1, 2, 3$
  - Too steep function increase may lead to inappropriate correspondence between too short segment of a and too long segment of b
- Boundary points restriction:
  - $f(1) = 1, f(K) = I$, where I is the count of the samples of the word a.
  - $g(1) = 1, g(K) = J$, where J is the count of the samples of the word b.

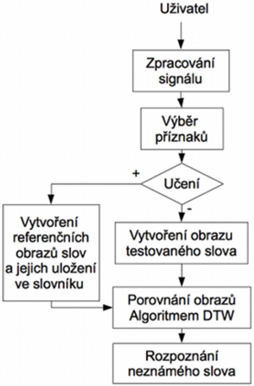
- DTW function growth global limits:
  - limits to maximum and minimum of the line first derivation defining the allowed area of the DTW function, where the boundary points constraints must be filled:

$$1 + \alpha[i(k) - 1] \leq 1 + \beta[i(k) - 1]$$

  - α – minimal line first derivation defining the allowed area
  - β – maximal line first derivation defining the allowed area.

DTW – Word Classifier Realization

Block schema



Obrázek: Block schema of the word classifier

DTW – Word Classifier Realization

Training

- General Algorithm:
  1. Either speaker or group of speakers pronounces each word of required vocabulary. It is done either once or repeatedly.
  2. Words on input are digitized and transferred by selected method of short-term signal analysis into the corresponding feature vectors.
  3. Word boundaries detection:
     - May be difficult due to the background noise for example.
     - Incorrect word boundaries deteriorates the recognition success rate.
     - Methods used to reduce the background sound influence increases the computational complexity.
  4. Creating reference words database..

DTW – realization

Methods used to create a reference word database

- Direct use of the words training set as the reference database – DTW does not require the reference word samples to be same length, but it is useful to perform the time normalization to be able to apply additional criteria.
- Creating average sample for each word *w* class
  - the linear and dynamic averaging methods are used.
- Creating sample words by clustering.
  - Words recordings are divided into clusters that each cluster contains "similar" word recording. Different clusters contains "different" word records.
  - Clustering can be done interactively (semi-automatic – chain map method, ISODATA algorithm), automatically (algorithms based on McQueen algorithm). See Mgr. J. Kučera final thesis.

DTW

Computational and Memory Complexity Reduction

- DTW Disadvantages – high memory and computing complexity can make real-time classification difficult even with relatively small dictionary.
- Solution:
  - Brute force – usage of either parallel processors or custom circuits – may be expensive.
  - Effective reference and testing words parameters encoding. Can be used:
    - vector quantization – the number of different word samples is finite – they are stored in the codebook and we can use their indices instead.
    - codebook – all samples included in the signal values alphabet (the encoding is more effective than the PCM).

DTW

Computational and Memory Complexity Reduction

- Usage of spectral stationarity area – method of spectral trace segmentation.
  - Spectral trace – feature vectors boundaries connector.
  - Can be approximated – by linear segments for example.
- Nearest neighbour search optimization:
  - metric spaces search methods
  - distance used in DTW must be a metric.

DTW

Computational and Memory Complexity Reduction

- Reduction of the computational requirements using heuristics by comparison.
  - Multi-level decision-making procedure:
    1. comparison of utterance using reduced feature vectors set against entire vocabulary
    2. searching the result of previous step using standard DTW.
  - Rejection threshold:
    1. We calculate distance of a word and the reference word in each step.
    2. When the distance is bigger then the experimentally established threshold, reference word is rejected.

Hidden Markov Models – HMM

- Speech modelling using HMM is based on the following idea of speech production:
  - Speech tract on short-term interval is in one of a finite amount of articulation configurations – generates a voice signal.
  - The configuration changes then.
- This activity is based on statistics.
- We can achieve a finite amount of all model parameters by all parameters quantization.

HMM

Speech recognition usage principles

- Two together tied time sequences of random variables are generated:
  - support Markov chain – finite number of states sequence
  - a string of finite number of spectral patterns.
- Random function assigning probability to state-pattern relation.
- The left-to-right Markov models are most often used for speech recognition:
  - suitable for increasing time related process modelling.

HMM

Markov process

- Markov process G with HMM is quintuplet $G = (Q, V, N, M, \pi)$
  - $Q = q_1, \ldots, q_k$ – set of states
  - $V = v_1, \ldots, v_k$ – set of input symbols
  - $N = (n_{i,j})$ – transition matrix. Evaluates the probability of transition from state q_i on time t_1 to state q_j on time t_2.
  - $M = (m_{i,j})$ – matrix assigning the probability that the acoustic vector v_j in state q_i no matter what time is it.
  - $\pi = (\pi_i)$ – initial state probability vector (probability of that the state i is the initial one).
- Triplet $\lambda = (N, M, \pi)$ – forms speech segment model.
  - the Vintsjuk's word model – 40 — 50 states (based on average count of micro segments in a word; segment length is 10 ms).

HMM

Determining the Probability of Utterance

- The probability is marked $P(O|\lambda)$
- The utterance O is usually processed as a sequences $O = (o_1, \ldots, o_T)$
  - T – number of utterance micro segments
  - o_i – corresponds to output symbols.
- Calculation of $P(O|\lambda)$ – the methods using the recursive enumeration either from the front or from the behind generated sequence (forward-backward algorithm).

HMM

Utterance probability determination – calculation

- Forward-backward calculation:
  - α_i – probability of transition into the state q_i while generating the output sequence $\{o_1, \ldots, o_t\}(\alpha_i = P(o_1 \ldots o_t, q_i(t)|\lambda)$
  - Recursive calculation:
    1. Initialization: $\alpha_1(i) = \pi_i m_i(o_1), i \in < 1, N >$
    2. Recursive step for t=1,... T-1:

       $$\alpha_{i+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) n_{i,j}] m_j(o_{i+1})$$

       for $j \in < 1, N >$, $m(o_t)$ is equal to notation $m_i(l)$, when $o_t = v_l$.
    3. Resulting probability:

       $$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

HMM

Alternative way of $P(O|\lambda)$ calculation

- Previous method disadvantages:
  - the result includes probabilities of all possible states sequences of length T.
- Solution:
  - calculation of maximum probable sequence of states Q.
- Calculation realized using the Viterbi algorithm:
  - the problem is solved recursively using dynamic programming techniques.

HMM

Training the model $\lambda = (N, M, \pi)$ parameters

- The procedure of training the model parameters must be determined.
- Training objectives:
  - maximization of the $P(O|\lambda)$ probability.
- Problem:
  - There is no analytical method to find the global maximum of a function of n variables.
- Solution:
  - Iterative algorithms for finding the local maximality can be utilized.
- The most used algorithm – Baum-Welch algorithm.
- Another problem while training the model:
  - finite training set problem:
    - The smaller training set is and the bigger the matrix M is, the higher probability that some elements in M will left 0 (the missing data problem).

HMM

Isolated word recognition decision rule

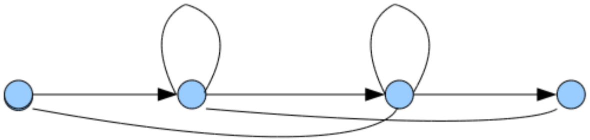
- The maximum credibility principle is used.
  1. For given word O and allλ:
     1. We calculate $P(O|\lambda)$.
  2. The result is the class with maximum value of $P(O|\lambda)$.

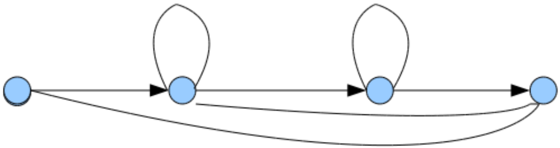
HMM

Implementation

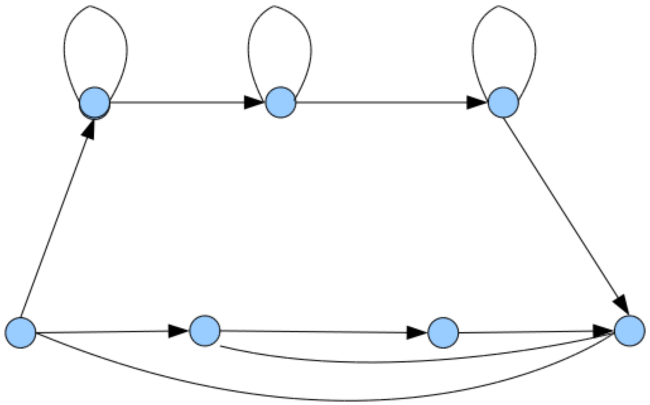
- Commands modelling:
  - Commonly the models with 4 — 7 states are used.
  - The tools for creating of HMM can be utilised during the modelling.
    - HTK – Hidden Markov Model Toolkit.
- Phoneme modelling:
  - 4 — 7 states usually
  - The word model – concatenation of phoneme models.
  - The real-time processing problems.
    - Can be solved using the special maximum $P(O|\lambda)$ searching algorithms.

Phoneme structure examples





Phoneme structure examples



Continuous speech recognition

- The principal differences to isolated word recognition:
  - the pattern database can not be created
  - the prosodic factors must be taken into the account
  - need to find word boundaries
  - the filler words/noises and speech errors must be processed.
- Solution – statistical approach:
  - language model
  - speaker model.
- Example: HMM returns the same probability of Czech words ,,máma" (mother) and ,,nána" (stupid girl) – the mother will be used – it's used more frequent.

Continuous speech recognition

Language models

- There are:
  - a word sequence (utterance) $W = (w_1, \dots, w_n)$
  - a sequence of acoustic vectors $O = (o_1, \dots, o_t)$.
- Our objective is to find W^* (set of all utterances), maximizing $P(W|O)$.
- According the Bayes' theorem:

$$P(W^*|O) = max\, P(W|O) = max \frac{P(W) * P(O|W)}{P(O)}$$

Continuous speech recognition

Language models – cont.

- We need to know following to find the $P(W^*|O)$ maximun:
  - a speaker model – $P(O|W)$
  - a language model – $P(W)$.
- The speaker model can be replaced by probabality of generating of W using the corresponding Markov model.
- The Trigram model:
  - Experimentally proven to be true:

$$P(w_n|w_1 \dots w_{n-1}) \cong P(w_n|w_{n-2}w_{n-1})$$

.

Continuous speech recognition

Topic recognition

- The speech recognition success rate is from aprox. 50 % — 99 % depending on the language, . . .
- The success rate can be improved by restricting the recognition domain:
  - topic recognition,
  - using the speech recognition grammar,
- When the topic is known:
  - the space state of trigrams and trigrams probabality can be changed:
    - For example stock market news – Was recognized the word "honey" or "money"?
  - more accurate language model can be created.

Speech recognition grammars

- The success rate of a general continuous speech recognition may drop to 50 %.
- It can be improved by limiting the recognition domain – by specification of allowed inputs for example.
- To limit allowed inputs the spech recognition grammars can be used:
  - context free grammars
- The possible ways of grammars notations:
  - using the logic programming methods
  - proprietary solutions
  - open standards – JSGF, W3C SRGS, ...

Speech recognition grammars

Java Speech Grammar Specification (JSGF)

- Textual grammar notation independent on platform and vendor.
- Design to be used in speech recognition.
- Part of the Java Speech API.
- It uses the Java style and conventions.
- Present veion 1.0 (říjen 1998).
- Used for example by the recognizer Sphinx-4, the VoiceXML interpreter VoiceGlue, ...
- More details in the 2nd half of semester on dialogue interfaces.

```
#JSGF
<koren> = I want to go by <what> .|
I want to go by <what> from <where> to <where> .|
I want to gou by <what> from <where> to <where> at
<when> .;
<what> = train| bus;
<where> = <city>;
<when> = <time>;
```

Speech recognition grammar

W3C Speech Recognition Grammar Specification (SRGS)

- W3C Standard.
- Current version 1.0 (March 2004).
- Defines the way of rules notation and referencing.
- Two possible notations:
  - XML
  - ABNF (Augmented BNF).
- In more detail on the 2nd half of the polovině semester (dialogue interfaces).

W3C SRGS Demo


```
#ABNF 1.0 UTF-8
root $greating;
language en-GB;
mode voice;
$greating = hello
```

```
<?xml version=”1.0”encoding=”utf-8”? >
<grammar root=”greating”xml:lang=”en-US”version=”1.0”>
<rule id=”greating”>
hello
< /rule>
< /grammar>
```