Dynamic Time Warping

Dr Philip Jackson

- Acoustic features
- Distance measures
- Pattern matching
- Distortion penalties
- DTW speech recognizer

www.ee.surrey.ac.uk/Teaching/Courses/eem.ssr
Statistical modeling of speech

1. Recognizing patterns

2. Markov and hidden-Markov processes
   - Likelihood calculation
   - Decoding
   - Training

3. Gaussian pdfs
   - HMM-GMM

4. Practical issues
Recognizing patterns

Pattern recognition is an important enabling technology in many machine intelligence applications, e.g., automatic speech recognition. A pattern is a structured sequence of observations.

Various approaches have been employed:

- Rule-based heuristics
- Pattern matching
  - dynamic time warping (deterministic)
  - Hidden Markov models (stochastic)
- Classification
  - artificial neural networks (discriminative)
Distance-from-template pattern matching

- **Template**
  - the features of a typical example of the sequence to be recognized
  - e.g., filterbank, linear prediction, PLP, cepstrum, MFCC

- **Distance**
  - a measure of how well the features of a new test sequence match those of the reference template
  - e.g., Euclidean distance, Mahalanobis distance, Itakura distance
Sequences of speech features

Time-frequency representation of speech as observation sequences for a template and two test words: “match”, “match”, “dummy”
Dynamic time warping

DTW is a method of pattern matching that allows for timescale variations in sequences of the same class.

Two aligned instances of the same word (after Holmes & Holmes, 2001). Open circles mark permitted predecessors to the closed circle at \((t, i)\).
Dynamic programming for time alignment

Cumulative distance along the best path upto frame $N$ in template and $T$th test frame is:

$$D(T, N) = \min_{t,i} \sum_{t \in 1..T} \sum_{i \in 1..N} d(t, i)$$ (1)

where $d(t, i)$ is distance between features from $t$th frame of test utterance and those from $i$th frame of template.

Allowing transitions from current and previous frames only, we compute the cost recursively:

$$D(t, i) = \min [D(t, i - 1), D(t - 1, i - 1), D(t - 1, i)] + d(t, i)$$ (2)
Basic DTW algorithm

1. Initialise the cumulative distances for \( t = 1, \)

\[
D(1, i) = \begin{cases} 
  d(1, i) & \text{for } i = 1 \\
  D(1, i - 1) + d(1, i) & \text{for } i = 2, \ldots, N 
\end{cases}
\]  

(3)

2. Recur for \( t = 2, \ldots, T, \)

\[
D(t, i) = \begin{cases} 
  D(t - 1, i) + d(t, i) & \text{for } i = 1 \\
  \min \{ D(t, i - 1), D(t - 1, i - 1), D(t - 1, i) \} + d(t, i) & \text{for } i = 2, \ldots, N 
\end{cases}
\]  

(4)

3. Finalise, the cumulative distance up to the final point gives the total cost of the match:

\[
D(T, N)
\]
Sequences of speech features

Time-frequency representation of speech as observation sequences for a template and two test words: “match”, “match”, “dummy”
Inter-utterance distances

Euclidean distances between template and test word features
DTW question

Now we can efficiently compute the cost of matching any test utterance against our template, using the path of the best alignment.

How can we determine the route of the best path?
DTW with traceback to store the best path

1. Initialise distances and traceback indicator for $t = 1$,

$$D(1, i) = \begin{cases} 
  d(1, i) & \text{for } i = 1 \\
  d(1, i) + D(1, i - 1) & \text{for } i = 2, \ldots, N 
\end{cases}$$

$$\phi(1, i) = \begin{cases} 
  [0, 0] & \text{for } i = 1 \\
  [1, i - 1] & \text{for } i = 2, \ldots, N 
\end{cases}$$

2. Recur for cumulative distances at $t = 2, \ldots, T$,

$$D(t, i) = \begin{cases} 
  d(t, i) + D(t - 1, i) & \text{for } i = 1 \\
  d(t, i) + \min \left[ D(t, i - 1), D(t - 1, i - 1), D(t - 1, i) \right] & \text{for } i = 2, \ldots, N 
\end{cases}$$

$$\phi(t, i) = \begin{cases} 
  [t - 1, i] & \text{for } i = 1 \\
  \arg \min \left[ D(t, i - 1), D(t - 1, i - 1), D(t - 1, i) \right] & \text{for } i = 2, \ldots, N 
\end{cases}$$
3. Final point gives the total alignment cost $D(T, N)$ and the end coordinates of the best path $z_K = [T, N]$, where $K$ is the number of nodes on the optimal path.

4. Trace the path back for $k = K - 1, \ldots, 1$,
   
   $z_k = \phi(z_{k+1})$, and

   $Z = \{z_1, \ldots, z_K\}$. 


Alternative sets of predecessor nodes

How could we refine the search strategy to encourage linear alignment, meanwhile allowing some warping?

Permissible preceding nodes under various transition constraints.
Abbridded DTW with distortion penalty

1. Initialise distance scores for $t = 1$,

\[
D(1, i) = \begin{cases} 
    d(1, i) & \text{for } i = 1 \\
    d(1, i) + D(1, i - 1) + d_V & \text{for } i = 2, \ldots, N
\end{cases}
\]

2. Recur for $t = 2, \ldots, T$,

\[
D(t, i) = \begin{cases} 
    d(t, i) + D(t - 1, i - 1) + d_H & \text{for } i = 1 \\
    \min \left\{ 
        d(t, i) + D(t, i - 1) + d_V, \\
        2d(t, i) + D(t - 1, i - 1), \\
        d(t, i) + D(t - 1, i) + d_H
    \right\} & \text{for } i = 2, \ldots, N
\end{cases}
\]

where $d_V$ and $d_H$ are costs associated with vertical and horizontal transitions respectively.

3. Finalise the penalised alignment cost $D(T, N)$. 

D.15
Distortion penalty examples

Best paths (clockwise from top left) for basic DTW, for DTW with standard, low and high distortion penalties.
Practical methods for search pruning

• Reducing the search space:
  – by gross partitioning
  – by score pruning
Example 1: Isolated word recognition

Given templates for the words “yes” and “no”, we can devise a binary recognition task that has a grammar:

SENT-START ( YES | NO ) SENT-END

Grammar for a binary isolated word recognition (IWR) task
Example 2: Connected word recognition

Left: grammar for a connected word recognition task
Right: trellis diagram showing connected templates

Distance metric now extends across word boundaries:

\[ D(t, 1, n) = d(1, 1, n) + \min_m [D(t - 1, m, L(m))] \], \hspace{1cm} (5) \]

where \( n \) is the current template, \( m \) is the previous one, \( L(m) \) is its length, and \( V \) is the vocabulary size.
Example 3: Template end-pointing

An end-pointing procedure can be used for automated cropping of recorded audio files:

SENT-START ( SIL WILDCARD SIL ) SENT-END

- End-pointing
  - Word detection
  - Allowing for errors during enrolment

Grammar for an end-pointing task
Training a DTW recognizer

• Enrolment
  – training session with new user
  – recordings used to provide templates
  – silence template

• Reliable templates
  – time aligning examples
  – clustering features
  – end-point detection using wildcards
  – word-boundary segmentation

Grammar for connected-word segmentation task
Summary of Dynamic Time Warping

- **Distance measures**
  - features: filterbank, MFCC, PLP
  - metrics: Euclidean, Malhalanobis, Itakura

- **Isolated Word Recognition**
  - DTW time alignment
  - traceback
  - distortion penalties
  - pruning
  - end-pointing

- **Connected Word Recognition**
  - silence and wildcard templates
  - word segmentation

- **Training a DTW recognizer**
  - enrollment recordings
  - reliable templates
Summary of Dynamic Time Warping

The DTW approach allows efficient computation with limited flexibility in the alignment. It treats templates as deterministic with residual noise.

Problems:

1. How much flexibility should we allow?
2. How should we penalise any warping?
3. How do we determine a fair distance metric?
4. How many templates should we register?
5. How do we select the best ones?

Solution:

- Develop an inference framework to build templates based on the statistics of our data.
Characteristics of the desired model

1. evolution of sequence should not be deterministic
2. observations are coloured depending on class
3. cannot directly observe class
4. stochastic sequence + stochastic observations

Applications:
- automatic speech recognition
- optical character recognition
- protein and DNA sequencing
- speech synthesis
- noise-robust data transmission
- crytoanalysis
- machine translation
- image classification, etc.