Statistical Parametric Speech Synthesis

Vassilis Tsiaras
University of Crete
May 2017
Contents

- HMMs
- LDMs
- Linguistic to phonetic mappings
  - Decision tree clustering
  - Deep neural networks
- Global variance
HMMs: Example
HMMs: Example

Synthesis
HMMs: Example

- Synthesis
- Discontinuities
- Model the derivative
HMMs: Example

- Synthesis
- Discontinuities
- Model the derivative
- Use both models
Linear Dynamical Models

- LDMs are described by a hidden Markov chain

```
\begin{align*}
\text{State evolution Model: } & \quad p(x_t|x_{t-1}) = N(x_t; F \cdot x + g, Q) \\
\text{Observation Model: } & \quad p(y_t|x_t) = N(y_t; h(x_t), R) \\
\end{align*}
```

- $x_t$: Abstract state, Articulators, Sinusoidals, e.t.c.
- $y_t$: (mceps, F0, bap), Sinusoidal parameters, Raw speech
- $h(x)$ affine transformation or non-linear map. E.g., $h(x) = H \cdot x + \mu$
LDM vs HMM

LDM: $x_t$ continuous
LDM: $p(x_t|x_{t-1}) = N(x_t; Fx_{t-1} + g, Q)$

HMM: $x_t \in \{1, 2, \ldots, N\}$
HMM: $p(x_t|x_{t-1})$ arbitrary

Observation models

Diagram: Transition probabilities between states 1, 2, and 3 with arrows indicating transition probabilities 0.5, 0.2, 0.1, 0.9, 0.3, and 0.4.
Linear Dynamical Models

- An LDM is a generative model with a time-varying multivariate unimodal Gaussian output distribution.
- The LDM is specified by the following pair of equations:

  \[ x_t = Fx_{t-1} + g + w_t \quad w_t \sim N(0, Q) \quad x_t, w_t \in \mathbb{R}^n \quad F \in \mathbb{R}^{n \times n} \]

  \[ y_t = Hx_t + \mu + v_t \quad v_t \sim N(0, R) \quad y_t, v_t \in \mathbb{R}^m \quad H \in \mathbb{R}^{m \times n} \]

- We further assume that the initial state is Gaussian \( x_1 \sim N(g_1, Q_1) \)
Factor Analysis

- Factor analysis is a statistical method for modeling the covariance structure of high dimensional static data using a small number of latent (hidden) variables.

\[ x = w \quad w \sim N(0, I) \quad w \in \mathbb{R}^n \]

\[ y = Hx + v \quad v \sim N(\mu, R) \quad y, v \in \mathbb{R}^m \quad H \in \mathbb{R}^{m \times n} \]

- Example \( n = 1, m = 2 \)

\[ w \sim N(0,1) \]

\[ v \sim N\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \right) \]

\[ H = 5 \times \begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \end{bmatrix} \]
Factor Analysis

\[ x \rightarrow Hx \]

\[ x \rightarrow v \]

\[ v \rightarrow y \]
Factor Analysis

- Example $n = 1, m = 2$

\[
x = w \quad w \sim N(0, I) \quad w \in \mathbb{R}^n
\]

\[
y = Hx + v \quad v \sim N(\mu, R) \quad y, v \in \mathbb{R}^m \quad H \in \mathbb{R}^{m \times n}
\]

\[
w \sim N(0,1)
\]

\[
v \sim N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix})
\]

\[
H = 5\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}
\]

- $m+m+m\times n$ parameters

\[
R = \begin{bmatrix} 16.20 & -12.54 \\ -12.54 & 16.97 \end{bmatrix} \quad R \in \mathbb{R}^{m \times m}
\]

Covariance matrix: $m \times m$ parameters
Dynamics of LDMs

• Is the dynamic of the state evolution process of LDM rich enough?
  \[
x_t = Fx_{t-1} + g + w_t \quad w_t \sim N(0, Q) \quad \text{Gauss-Markov random process}
\]
  \[
y_t = Hx_t + \mu + v_t \quad v_t \sim N(0, R)
\]

• Does a second order recursion offer any advantage?
  \[
x_t = F_1x_{t-1} + F_2x_{t-2} + g + w_t \quad w_t \sim N(0, Q)
\]
  \[
y_t = Hx_t + \mu + v_t \quad v_t \sim N(0, R)
\]

• Second order systems model many physical systems
  (e.g. mass \times\text{ acceleration} = \text{spring force} + \text{friction})

• Any high order linear recursion can be converted into a first order recursion by increasing the number of states.
Linear Dynamical Models – State Evolution

- Based on MRI images of speech production, researchers at Haskins Laboratories developed differential equations that describe how the articulators move to produce a particular utterance.

- The motions of the articulators are calculated as a critically-damped spring-mass model

\[
\frac{d^2 x(t)}{dt^2} + 2S \frac{dx(t)}{dt} + S^2 (x(t) - u) = w
\]
LDMs and Autoregressive Models

- A p-order vector autoregressive (AR) model

\[ z_k = \sum_{i=1}^{p} A_i z_{k-i} + w_k \]

\[
\begin{bmatrix}
    z_k \\
    z_{k-1} \\
    \vdots \\
    z_{k-p+2} \\
    z_{k-p+1}
\end{bmatrix} =
\begin{bmatrix}
    A_1 & A_2 & \cdots & A_{p-1} & A_p \\
    I & 0 & \cdots & 0 & 0 \\
    0 & I & \cdots & 0 & 0 \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    0 & 0 & \cdots & I & 0
\end{bmatrix}
\begin{bmatrix}
    z_{k-1} \\
    z_{k-2} \\
    \vdots \\
    z_{k-p+1} \\
    z_{k-p}
\end{bmatrix} +
\begin{bmatrix}
    w_k \\
    0 \\
    \vdots \\
    0 \\
    0
\end{bmatrix}
\]

\[ x_{k+1} = Fx_k + w_k' \]

\[ y_k = x_k \]
A 1-dimension state can model exponential decay or growth or random walks.

With a state dimension of 2, the transform $F$ can be composed of oscillations + damped + exponentially increasing oscillations.
Main issues using an LDM

- **Evaluation problem.** Given the LDM $\theta$ and the observation sequence $Y=y_1 y_2 \ldots y_T$, calculate the probability that model $\theta$ has generated sequence $Y$.

- **Inference.** Given the LDM $\theta$ and the observation sequence $Y=y_1 y_2 \ldots y_T$, calculate the probability of hidden states $x_t$ that produced this observation sequence $Y$.

- **Learning problem.** Given some training observation sequences $Y=y_1 y_2 \ldots y_T$, determine LDM parameters $\theta$ that best fit training data.
Properties of Normal Distributions

The density of random variable \( x \sim N(\mu, \Sigma) \) is

\[
p(x) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]
\]
Properties of Normal Distributions

- The class of Gaussian distributions is closed under linear transformations.

- If $X$ is normally distributed with mean $\mu$ and standard deviation $\sigma$, then the variable $Y = aX + b$, for any real numbers $a$ and $b$, is also normally distributed, with mean $a\mu + b$ and standard deviation $|a|\sigma$.

- Also if $X_1$ and $X_2$ are two independent normal random variables, with means $\mu_1$, $\mu_2$ and standard deviations $\sigma_1$, $\sigma_2$, then their sum $X_1 + X_2$ will also be normally distributed, with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$. 
Properties of Normal Distributions

- Let \( x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \) be an n-dimensional random vector with distribution \( x \sim N(\mu, \Sigma) \)

- where \( x_1 \) and \( x_2 \) are two sub-vectors of respective dimensions \( p \) and \( q \), with \( p+q = n \).

- \( \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \)

- Note that \( \Sigma^T = \Sigma \) and \( \Sigma_{21}^T = \Sigma_{12} \)
Properties of Normal Distributions

- Theorem

- The marginal distributions of $x_1$ and $x_2$ are also normal with mean vector $\mu_i$ and covariance matrix $\Sigma_{ii}$ ($i=1,2$), respectively.

- The conditional distribution of $x_i$ given $x_j$ is also normal with mean vector

  $$\mu_{i|j} = \mu_i + \Sigma_{ij}\Sigma_{jj}^{-1}(x_j - \mu_j)$$

  and covariance matrix

  $$\Sigma_{i|j} = \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ij}^T$$
Evaluation and Inference

- The initial state $p(x_1)$, the transition, $p(x_t|x_{t-1})$, and the observation, $p(y_t|x_t)$, probabilities are Gaussians.

- Then the joint probability distribution $p(X,Y)$ is Gaussian.

- Also, the marginal probability distributions $p(x_t|y)$, $p(x_{t-1}, x_t|y)$ and $p(Y)$ are Gaussians.
Filtering & smoothing

- For online data analysis, we seek filtered state estimates given earlier observations:

\[ p(x_t \mid y_1, y_2, \ldots, y_t) \quad t = 1, 2, \ldots \]

- In other cases, find smoothed estimates given earlier and later of observations:

\[ p(x_t \mid y_1, y_2, \ldots, y_T) \quad t = 1, 2, \ldots, T \]
LDM: Filtering

\[ \alpha_t(x_t) \triangleq p(x_t | y_1, \ldots, y_t) \]

\[ \hat{\alpha}_t(x_t) = \frac{1}{c_t} p(y_t | x_t) \int p(x_t | x_{t-1} = z) \hat{\alpha}_{t-1}(z) dz \]

Normalization constant

Prediction: \( p(x_t | y_1, \ldots, y_{t-1}) \)

Update: \( p(x_t | y_1, \ldots, y_t) \)
LDM: Smoothing

\[ p(x_t \mid y) \propto p(x_t \mid y_1, \ldots, y_t) p(y_{t+1}, \ldots, y_T \mid x_t) \]

- The forward-backward algorithm updates filtering via a reverse-time recursion:

\[
\hat{\beta}_{t-1}(x_{t-1}) = \frac{1}{c_t} \int p(x_t = z \mid x_{t-1}) p(y_t \mid x_t = z) \hat{\beta}_t(z) dz
\]
LDM: Smoothing

- Backward recursion:

\[ \hat{\beta}_{t-1}(x_{t-1}) = \frac{1}{c_t} \int p(x_t = z | x_{t-1}) p(y_t | x_t = z) \hat{\beta}_t(z) dz \]

- Sequential recursion:

\[ \hat{\alpha}_{t-1}(x_{t-1}) \hat{\beta}_{t-1}(x_{t-1}) = \int p(x_{t-1} | x_t = z, y_{1:t-1}) \hat{\alpha}_t(z) \hat{\beta}_t(z) dz \]
Optimal state estimation

- Probabilities measure the posterior confidence in the true hidden states
- For the learning problem, the following marginal probabilities are inferred from the observations

\[
p(x_t | y) = \frac{1}{c_t} \hat{\alpha}_t(x_t) \hat{\beta}_t(x_t)
\]

\[
p(x_{t-1}, x_t | y) = \frac{1}{c_t} \hat{\alpha}_{t-1}(x_{t-1}) p(x_t | x_{t-1}) p(y_t | x_t) \hat{\beta}_t(x_t)
\]
Kalman Filtering and Smoothing

- Dynamics and Observation model
  \[ x_t = Fx_{t-1} + g + w_t, \quad w_t \sim N(0, Q) \]
  \[ y_t = Hx_t + \mu + v_t, \quad v_t \sim N(0, R) \]

- Kalman Filter:
  - Compute \( p(X_t \mid Y_1 = y_1, \ldots, Y_t = y_t) \)
  - Real-time, given data so far

- Kalman Smoother:
  - Compute \( p(X_t \mid Y_1 = y_1, \ldots, Y_T = y_T), \quad 1 \leq t \leq T \)
  - Post-processing, given all data
Optimal state estimation

Definitions

\[ p(x_t | y_{1:t-1}) = N(x_t; \hat{x}_{t|t-1}, \hat{\Sigma}_{t|t-1}) \]

\[ p(x_t | y_{1:t}) = N(x_t; \hat{x}_{t|t}, \hat{\Sigma}_{t|t}) \]

\[ p(x_t | y_{1:T}) = N(x_t; \hat{x}_{t|T}, \hat{\Sigma}_{t|T}) \]

\[ p(x_t, x_{t-1} | y_{1:T}) = N(x_t; \hat{x}_{t,t-1|T}, \hat{\Sigma}_{t,t-1|T}) \]

\[ \hat{R}_{t|T} = \hat{\Sigma}_{t|T} + \hat{x}_{t|T} \hat{x}_{t|T}^T \]

\[ \hat{R}_{t,t-1|T} = \hat{\Sigma}_{t,t-1|T} + \hat{x}_{t|T} \hat{x}_{t-1|T}^T \]
The set of Kalman Filtering Equations in Detail

### Prediction (Time Update)

1. Project the state ahead
   \[ \hat{x}_{t|t-1} = F\hat{x}_{t-1|t-1} + g \]
2. Project the error covariance ahead
   \[ \hat{\Sigma}_{t|t-1} = F\hat{\Sigma}_{t-1|t-1}F^T + Q \]

### Correction (Measurement Update)

1. Compute the Kalman Gain
   \[ K_t = \hat{\Sigma}_{t|t-1}H^T(H\hat{\Sigma}_{t|t-1}H^T + R)^{-1} \]
2. Update estimate with measurement \( x_t \)
   \[ \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H\hat{x}_{t|t-1} - \mu) \]
3. Update Error Covariance
   \[ \hat{\Sigma}_{t|t} = \hat{\Sigma}_{t|t-1} - K_tH\hat{\Sigma}_{t|t-1} \]
Algorithm 6: Kalman Filter

**Data:** Observations, $y_{1:T}$, and model parameters: $F, g, Q, H, \mu, R, g_1, Q_1$

**Result:** $\log L = \log(p(y_{1:T}))$ and statistics $\hat{x}_{t|t}$, $\hat{\Sigma}_{t|t}$, $t \in \{1, \ldots, T\}$,

$\hat{x}_{t|t-1}$, $\hat{\Sigma}_{t|t-1}$, $t \in \{2, \ldots, T\}$

/**< Initialization */
$\hat{x}_{t|t-1} = g_1; \quad \hat{\Sigma}_{t|t-1} = Q_1; \quad \log L = 0$

for $t = 1:T$ do

/**< Prediction */
if $t > 1$ then

$\hat{x}_{t|t-1} = F\hat{x}_{t-1|t-1} + g$
$\hat{\Sigma}_{t|t-1} = F\hat{\Sigma}_{t-1|t-1}F^\top + Q$

/**< Update */
$e_t = y_t - (H\hat{x}_{t|t-1} + \mu)$
$\hat{\Sigma}_{e_t} = H\hat{\Sigma}_{t|t-1}H^\top + R$
$K_t = \hat{\Sigma}_{t|t-1}H^\top\hat{\Sigma}_{e_t}^{-1}$
$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_te_t$
$\hat{\Sigma}_{t|t} = \hat{\Sigma}_{t|t-1} - K_tH\hat{\Sigma}_{t|t-1}$

/**< Compute $\log p(Y|\theta) = \sum_{t=1}^{T} \log(c_t)$ */
$l_{c_t} = \log \mathcal{N}(e_t; 0, \hat{\Sigma}_{e_t})$
$\log L = \log L + l_{c_t}$

end for
Algorithm 7: Kalman Smoother

Data: Statistics $\hat{x}_t|t$, $\hat{\Sigma}_t|t$, $\hat{x}_{t|t-1}$, $\hat{\Sigma}_{t|t-1}$ calculated from Kalman filter, and model parameter $F$

Result: Statistics $\hat{x}_t|T$, $\hat{P}_t|T$, $t \in \{1, \ldots, T\}$ and $\hat{P}_{t,t-1|T}$, $t \in \{2, \ldots, T\}$

$$\hat{P}_{T|T} = \hat{\Sigma}_{T|T} + \hat{x}_{T|T}\hat{x}_{T|T}^\top$$

for $t = T:-1:2$ do

$$J_t = \hat{\Sigma}_{t-1|t-1}F^\top\hat{\Sigma}_{t|t-1}^{-1}$$

$$\hat{x}_{t-1|T} = \hat{x}_{t-1|t-1} + J_t(\hat{x}_{t|T} - \hat{x}_{t|t-1})$$

$$\hat{\Sigma}_{t-1|T} = \hat{\Sigma}_{t-1|t-1} + J_t(\hat{\Sigma}_{t|T} - \hat{\Sigma}_{t|t-1})J_t^\top$$

$$\hat{\Sigma}_{t,t-1|T} = \hat{\Sigma}_{t|T}J_t^\top$$

$$\hat{P}_{t-1|T} = \hat{\Sigma}_{t-1|T} + \hat{x}_{t-1|T}\hat{x}_{t-1|T}^\top$$

$$\hat{P}_{t,t-1|T} = \hat{\Sigma}_{t,t-1|T} + \hat{x}_{t|T}\hat{x}_{t-1|T}^\top$$
Identification of Model Parameters

• The parameters of an autoregressive (AR) model can be specified by solving the Yule-Walker equations.

• However, there is no closed form solution to parameter identification in LDMs.
• Parameters can be estimated by minimizing the log likelihood:
  – Numerical Optimization Algorithms (e.g. Steepest Descent)
  – Expectation Maximization Algorithm
  – Subspace Algorithms (N4SID)
Likelihood

- **Probability**: predict unknown *outcomes* based on known *parameters*:
  - \( p(x | \theta) \)

- **Likelihood**: estimate unknown *parameters* based on known *outcomes*:
  - \( L(\theta | x) = p(x | \theta) \)

- **Likelihood function and auxiliary function**

\[
\mathcal{L}(\theta) = \log p(Y | \theta) = \log \left( \int_X p(X, Y | \theta) dX \right)
\]

\[
Q(\theta_i, \theta) = E[\log p(x_1 | \theta) | Y, \theta_i] + \sum_{t=1}^{T-1} E[\log P(x_{t+1} | x_t, \theta) | Y, \theta_i] + \sum_{t=1}^{T} E[\log P(y_t | x_t, \theta) | Y, \theta_i]
\]
Likelihood

- Auxiliary function

\[
Q(\theta_i, \theta) = \text{const} - \frac{1}{2} \log |Q_1| - \frac{1}{2} E \left[ (x_1 - g_1)^T Q_1^{-1} (x_1 - g_1) | Y, \theta_i \right] - \frac{T - 1}{2} \log |Q| \\
- \frac{1}{2} \sum_{t=2}^{T} E \left[ (x_t - F x_{t-1} - g)^T Q^{-1} (x_t - F x_{t-1} - g) | Y, \theta_i \right] \\
- \frac{T}{2} \log |R| - \frac{1}{2} \sum_{t=1}^{T} E \left[ (y_t - H x_t - \mu)^T R^{-1} (y_t - H x_t - \mu) | Y, \theta_i \right]
\] (54)
Maximize Likelihood

• The E-step of EM requires computing the expected log-likelihood

\[ L = E \left[ \log p(\{x_{1:T}\}, \{y_{1:T}\} | \theta) \right | \{y_{1:T}\}, \theta] \]

• This quantity depends on three expectations

\[ E[x_t | y_{1:T}] = \hat{x}_{1|T} \]
\[ E[x_t x_t^T | y_{1:T}] = \hat{\Sigma}_{t|T} + \hat{x}_{t|T} \hat{x}_{t|T}^T = \hat{R}_t \]
\[ E[x_t x_{t-1}^T | y_{1:T}] = \hat{\Sigma}_{t,t-1|T} + \hat{x}_{t|T} \hat{x}_{t-1|T}^T = \hat{R}_{t,t-1} \]
Sufficient statistics

\[ \zeta_1 = \sum_{t=1}^{T-1} \hat{x}_{t|T} \]
\[ \zeta_2 = \sum_{t=2}^{T} \hat{x}_{t|T} \]
\[ \zeta_3 = \sum_{t=1}^{T} \hat{x}_{t|T} \]
\[ \zeta_4 = \sum_{t=1}^{T} y_t \]
\[ \Gamma_1 = \sum_{t=1}^{T-1} \hat{R}_{t|T} \]
\[ \Gamma_2 = \sum_{t=2}^{T} \hat{R}_{t|T} \]
\[ \Gamma_3 = \sum_{t=1}^{T} \hat{R}_{t|T} \]
\[ \Gamma_4 = \sum_{t=2}^{T} \hat{R}_{t,t-1|T} \]
\[ \Gamma_5 = \sum_{t=1}^{T} y_t \hat{x}_t^T|T \]
\[ \Gamma_6 = \sum_{t=1}^{T} y_t y_t^T \]
### M-step

<table>
<thead>
<tr>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_1 = \hat{x}_{1</td>
</tr>
<tr>
<td>$Q_1 = \hat{R}_{1</td>
</tr>
<tr>
<td>$F = (\Gamma_4 - \frac{1}{T-1}\zeta_2 \zeta_1^T)(\Gamma_1 - \frac{1}{T-1}\zeta_1 \zeta_1^T)^{-1}$</td>
</tr>
<tr>
<td>$g = \frac{1}{T-1}(\zeta_2 - F \zeta_1)$</td>
</tr>
<tr>
<td>$Q = \frac{1}{T-1} \left( \Gamma_2 - F \Gamma_4^T - g \zeta_2^T \right)$</td>
</tr>
<tr>
<td>$H = (\Gamma_5 - \frac{1}{T}\zeta_4 \zeta_3^T)(\Gamma_3 - \frac{1}{T}\zeta_3 \zeta_3^T)^{-1}$</td>
</tr>
<tr>
<td>$\mu = \frac{1}{T}(\zeta_4 - H \zeta_3)$</td>
</tr>
<tr>
<td>$R = \frac{1}{T} \left( \Gamma_6 - H \Gamma_5^T - \mu \zeta_4^T \right)$</td>
</tr>
</tbody>
</table>
Training LDMs for Synthesis

Each utterance consists of segments of phones or subphones.
- Train an LDM for each label $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, \ldots$

- Problem: There are segments that consists of one only frame.
EM Algorithm

• Training an LDM for label $\phi_i$
• Initial guesses of $F, H, Q, R, g, \mu, g_1, Q_1$

• Kalman smoother (E-step):
  – Clear the sufficient statistics variables
  – For each example $y_{i1}, \ldots, y_{iT}$ in $\phi_i$
    • Compute distributions of $X_1, \ldots, X_T$
given data $y_{i1}, \ldots, y_{iT}$ and $F, H, Q, R, g, \mu, g_1, Q_1$.
    • Accumulate the sufficient statistics into global variables

• Update parameters (M-step):
  – Update $F, H, Q, R, g, \mu, g_1, Q_1$ based on sufficient statistics.

• Repeat until convergence (local optimum)
Training

Observation vectors
Cepstrum coefficients + F0

\[ y_{11}, y_{12}, \ldots, y_{1T_1} \]
\[ y_{21}, y_{22}, \ldots, y_{2T_2} \]
\[ \ldots \]
\[ y_{k1}, y_{k2}, \ldots, y_{kT_k} \]

LDM model

\[ x_1 \sim N(g_1, Q_1) \]
\[ x_t = Fx_{t-1} + g + w_t \]
\[ y_t = Hx_t + \mu + v_t \]

\[ w_t \sim N(0, Q) \]
\[ v_t \sim N(0, R) \]

Maximize likelihood to estimate the parameters

\[ F, H, Q, R, g, \mu, g_1, Q_1 \]

and the hidden states

\[ x_{11}, x_{12}, \ldots, x_{1T_1} \]
\[ x_{21}, x_{22}, \ldots, x_{2T_2} \]
\[ \ldots \]
\[ x_{k1}, x_{k2}, \ldots, x_{kT_k} \]

Parameters

\[ F, H, Q, R, g, \mu, g_1, Q_1 \]

Synthesis

Initial state

\[ x_1 = g_1 \]

State equation

\[ x_t = Fx_{t-1} + g \]

Hidden state vectors

\[ X_1, x_2, \ldots, x_T \]

Observation model

\[ y_t = Hx_t + \mu \]

Observation vectors

Cepstrum coefficients + F0

\[ y_1, y_2, \ldots, y_T \]

Speech

Duration of sub-phoneme
LDM Configurations

**Optimization of LDM training configurations:**

- The ideal state-space dimension is between 6 and 9
  - Low dimensional dynamics produce high dimensional observations (e.g., 40 cepstral coefficients)
- Matrices $Q$ and $R$ should be diagonal
- The parameter $\mu^o$ is necessary
- Stability constraints should be enforced to LDMs
- All models can have the same matrix $H$
The Edinburgh speech synthesis database was used

A baseline synthesizer based on HMM was trained

- Each full context model was represented by a five state left-to-right HMM
- Decision tree clustering was used to cluster similar HMM states
  - 340,885 states were tied to produce 965 states
- The observation vectors consisted of 40 mel-cepstral coefficients, delta and delta-delta.

The trained HMMs were used to segment the database at the state level

- The terminal nodes of the decision trees and their respective observations were used to train the LDMs

Histogram of the 4th mel-cepstral coefficient taken from the observations belonging to a given terminal node of the decision trees. This histogram follows approximately a Gaussian distribution.
A large number of LDM training configurations were performed

- The best configurations were identified

- The difference between natural and generated cepstra were measured with
  - mean value of the cepstral distance & b) raw PESQ

- Most of the considered LDM models produced cepstra closer to natural than HMMs

Trajectory of the 2-nd mel-cepstral coefficient. The dashed blue line shows the natural version, the black line shows the c(2) generated from HMMs, while the green line shows the c(2) generated from an LDM model
LDMs - Experiments

- Preliminary results using autoregressive models for speech synthesis
  - Autoregressive models are special cases of LDMs
  - They offer a paradigm of how to construct LDMs with state space dimension greater than the observation space dimension.
  - The quality of the produced speech is comparable to that of LDMs
- Discontinuities between neighbouring segments in synthesized speech
From linguistic specification to sequence of models

- Please call Stella
- sil p l ii z k oo l s t E l ax sil
- Context dependent linguistic units
  - pentaphones

```
0 2569160  sil^^sil^^sil^-p++l+=l
2569160 3467120  sil^^sil^^sil^-p++l+=ii
3467120 3766440  sil^^p--l++ii+=z
3766440 5163260  p^^l--ii++z+=k
5163260 6061220  l^^ii--z++k+=OO
6061220 6759640  ii^^z--k++OO+=l
6759640 8356010  z^^k--OO++l+=s
8356010 8755100  k^^OO--l++s+=t
8755100 9952380  OO^^l--s++t+=E
9952380 10551020  l^^s--t++E+=l
10551020 11349200  s^^t--E++l+=ax
11349200 12047620  t^^E--l++ax+=sil
12047620 12945580  E^^l--ax++sil+=sil
12945580 16038550  l^^ax--sil++sil+=sil
```
Linguistic-to-Acoustic Mappings

The simplest map is for each linguistic (phonetic and prosodic contextual unit) unit to assign an acoustic model (in our case an LDM). However, there are billions of linguistic units and less than a million training segments. Therefore there are not enough training samples to robustly train all models. Even worse, for some linguistic units there are not any training sample. Therefore we will end up with a situation like the following.

There are, for example, 42 examples (segments) in the database

<table>
<thead>
<tr>
<th>Linguistic unit 1</th>
<th>There is no example in the speech database</th>
<th>There is no example in the speech database</th>
<th>There are 3 examples in the database</th>
<th>There are 10 examples in the database</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDM 1</td>
<td>Cannot estimate any acoustic model</td>
<td>Cannot estimate any acoustic model</td>
<td>LDM 4</td>
<td>LDM 5</td>
</tr>
<tr>
<td></td>
<td>Robust estimate of parameters</td>
<td>Robust estimate of parameters</td>
<td>Not robust estimate of parameters</td>
<td>Marginally robust estimate of parameters</td>
</tr>
</tbody>
</table>
Linguistic-to-Acoustic Mappings

The problem arises when we want to synthesize speech (or acoustic) parameters from linguistic units 2, 3, or 4. Because the linguistic analysis of a text produced these units.

A solution is the following. Use the same LDM for more than one linguistic units. This trick is called clustering. Then the question is which linguistic units to assign to the same LDM. In our work we describe an algorithm of how to do clustering in an way that is close to optimal, using binary decision trees. Note that the problem is non-deterministic polynomial complete.
LDM: Decision Tree Clustering

- The LDM models are trained using full context labelling
  - Pentaphon context is used for mceps
  - Pentaphon context plus position in the sentence is used for F0
  - The context is independent of the number of states

Current phoneme

15 frames

Current phoneme

3 states in LDMs

5 states in HMMs
The LDM models are trained using full context labelling. One LDM models many pentaphons that have similar speech parameters. The training examples are clustered according to linguistic questions and how well they fit to LDM that models the examples of a cluster. Initially, all training examples are modelled with one LDM.

An LDM models an “average” trajectory of a set of example trajectories.

**Blue:** Trajectories of $c(0)$

**Red:** Synthetic trajectory of LDM
LDM: Decision Tree Clustering

- Hierarchical top-down clustering. Split if $L_y + L_n > L_p + \text{MDL\_threshold}$

**MDL\_threshold** = $\frac{1}{2} k \log N$

- $L_p$: Sum log-likelihood using LDM$_p$
- $L_n$: Sum log-likelihood using LDM$_n$
- Question: C_phone(notin)+continuant

Diagram:
- Yes branch
- No branch
LDM: Decision Tree Clustering Algorithm

- Create the root node of the decision tree, which contains all examples
  
```python
queue.put(rootNode)
```

- While(is_not_empty(queue))
  
```python
node = queue.pop()
```

- Find the question that has the largest $L_y + L_n$
  
```python
For each question  //Do this using Parallel Processing
- Split the examples associated with the current node
- Fit an LDM to “yes” examples and calculate $L_y$
- Fit an LDM to “no” examples and calculate $L_n$
- Check if $L_y + L_n > L_p + MDL_{-threshold}$ and store $L_y + L_n$
  ```

- If a (best) question is found
  
```python
Create tree node yesNode that contains the “yes” examples
Create tree node noNode that contains the “no” examples
queue.put(yesNode)
queue.put(noNode)
```
Application of LDMs to TTS – Clustering

- Part of the Decision Tree of F0

- C_phone(not in) - nasal
  - yes
  - L_phone(not in) - aspirated
    - 74908 training examples
    - 13419
    - C_phone(in) nasal _consonant
      - 3736
      - syllBGBw(=)12
        - 561
      - CuWrdPos(in) intransitive
        - 3175
      - R_phone(in) voiced _consonant
        - 6135
    - PhinWrdFwd(<=)5
      - 9683
      - L_phone(in) +strident
        - 3548
      - L_phone(in) +voiced
        - 5023
  - no
  - R_phone(in) voiced _consonant
    - 61489
    - CuWrdPos(in) intransitive
      - 39031
      - 22458
      - C_phone(in) vowel
        - 8025
      - L_phone(in) -strident
        - 17435
      - L_phone(in) voiced
        - 3548
      - L_phone(in) +strident
        - 5023
      - L_phone(in) +voiced
        - 17435
      - R_phone(in) voiced _consonant
        - 61489
      - C_phone(in) vowel
        - 31006
Application of LDMs to TTS – Clustering

- Part of the Decision Tree of mceps

```
Part of the Decision Tree of mceps

\[
\begin{align*}
\text{C}_\text{phone(notin)}^+\text{continuant} & \quad 74908 \text{ training examples} \\
\text{yes} & \\
\text{C}_\text{phone(in)}^+\text{consonantal} & \quad 26855 \\
\text{no} & \\
\text{C}_\text{phone(in)}^-\text{sonorant} & \quad 48053 \\
\end{align*}
\]

```

```
\[
\begin{align*}
\text{C}_\text{phone(in)}^-\text{nasal_consonant} & \quad 20667 \\
\text{LL}_\text{phone(in)}^\text{silence} & \quad 6188 \\
\text{C}_\text{phone(in)}^+\text{coronal_distributed} & \quad 12958 \\
\text{C}_\text{phone(in)}^-\text{dorsal_high} & \quad 35095 \\
\end{align*}
\]

```

```
\[
\begin{align*}
\text{L}_\text{phone(in)}^-\text{consonantal} & \quad 7231 \\
\text{C}_\text{phone(in)}^+\text{coronal_anterior} & \quad 13436 \\
\text{R}_\text{phone(notin)}^\text{consonant} & \quad 2836 \\
\text{L}_\text{phone(in)}^-\text{sonorant} & \quad 3352 \\
\text{C}_\text{phone(notin)}^-\text{aspirated} & \quad 3880 \\
\text{C}_\text{phone(in)}^+\text{coronal_anterior} & \quad 9078 \\
\text{C}_\text{phone(in)}^\text{front} & \quad 18856 \\
\text{C}_\text{phone(in)}^\text{consonant_manner} & \quad 16239 \\
\end{align*}
\]

```

74908 training examples

- 26855 examples with C_phone(notin) + consonantal
- 48053 examples with C_phone(in) - sonorant
- 20667 examples with C_phone(in) nasal_consonant
- 6188 examples with LL_phone(in) silence
- 12958 examples with C_phone(in) coronal_distributed
- 35095 examples with C_phone(in) dorsal_high
- 7231 examples with L_phone(in) consonantal
- 13436 examples with C_phone(in) + coronal_anterior
- 2836 examples with R_phone(notin) consonant
- 3352 examples with L_phone(in) sonorant
- 3880 examples with C_phone(notin) aspirated
- 9078 examples with C_phone(in) + coronal_anterior
- 18856 examples with C_phone(in) front
- 16239 examples with C_phone(in) consonant_manner ≧ approximant
Application of LDMs to TTS – Global Variance

- Global Variance (GV) is defined as an intra-utterance variance of a speech parameter trajectory and is modelled by a Gaussian distribution.
- The GV algorithm constrain the synthesized trajectories to have the same GV as the GV of the corresponding training samples.
- In speech parameter generation, the optimum parameter sequence is determined so as to maximize an objective function consisting of the LDM and GV log pdfs

\[
L = \frac{1}{T} \log P(Y \mid \bar{X}, \theta_{LDM}) + \log P(v \mid \theta_{GV})
\]

where \( \theta_{LDM} \) and \( \theta_{GV} \) are the parameters of the distributions of LDM and GV, \( Y \) are the trajectories of speech parameters (e.g., Cepstrum), vector \( v \) has the variances of \( Y \) trajectories, \( T \) is the duration of trajectories, and hidden state \( \bar{X} \) is

\[
\bar{X} = \arg \max P(X \mid \theta_{LDM})
\]

- The objective function \( L \) is maximized by a steepest decent algorithm.
GV has been applied both to traditional LDMs and to LDMs with critically dumped target-dynamics.

In informal subjective listening tests the volunteers preferred the GV LDM synthesized speech from the LDM synthesized speech.
Samples

Samples from the training set
HSMM duration. Synthesized Cepstrum, Band aperiodicity and F0

LDM

herald_264
herald_439
LDM GV

herald_264
herald_439
HSMM GV

Samples from the test set
HSMM duration. Synthesized Cepstrum, Band aperiodicity and F0

LDM

herald_413
herald_752
hvd_720
mrt_150
LDM GV

herald_413
herald_752
hvd_720
mrt_150
HSMM GV