Gaussian Process Experts for Voice Conversion

Nicholas Pilkington, *Heiga Zen*, *Mark J. F. Gales* (Toshiba Research Europe Ltd)

TOSHIBA
Leading Innovation

1. Introduction

GMM-based voice conversion (VC)

· Overview
- Train GMM on joint PDF using parallel speech corpus
- Derive conditional PDF given source feats
- Predict tgt feats from conditional PDF

· Limitations
- Oversmoothing ⇒ Poor modeling
- Overfitting ⇒ Poor generalization

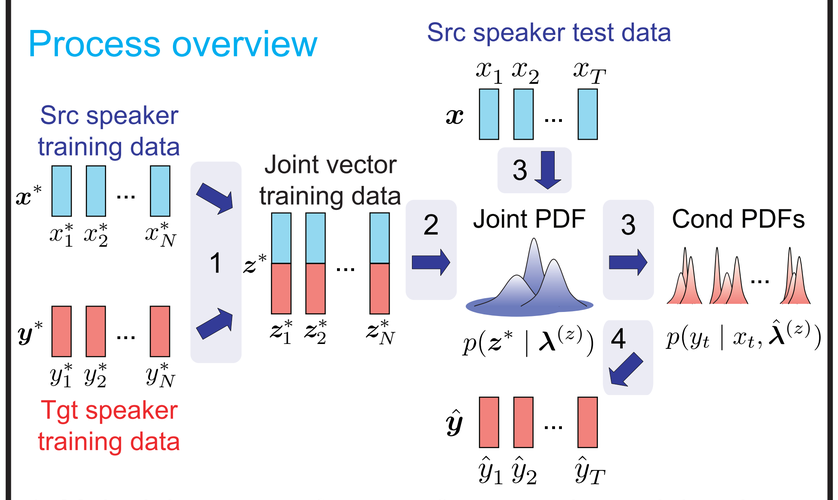
Gaussian process (GP)-based VC

· Gaussian Process [Rasmussen;'06]
- Non-parametric Bayesian model
- Many advantages over parametric approaches
  * Flexibility
  * Robust against overfitting

⇒ Can address limitations in GMM-based VC

2. GMM-based VC [Kain;'98]

Process overview



1. Make joint vectors from src & tgt speaker training data
2. Model joint PDF between src & tgt data by GMM

$$p(\boldsymbol{z}_t \mid \boldsymbol{\lambda}^{(\boldsymbol{z})}) = \sum_{m=1}^{M} w_m \, \mathcal{N}(\boldsymbol{z}_t \,;\, \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)}) \quad \boldsymbol{\mu}_m^{(z)} = \left[\mu_m^{(x)}, \mu_m^{(y)}\right]^\top$$

$$\hat{\boldsymbol{\lambda}}^{(z)} = \arg\max_{\boldsymbol{\lambda}^{(z)}} \; p(\boldsymbol{z}^* \mid \boldsymbol{\lambda}^{(z)}) \qquad \boldsymbol{\Sigma}_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{bmatrix}$$

3. Derive cond PDF given $\boldsymbol{x}$ from joint PDF

$$p(y_t \mid x_t, \hat{\boldsymbol{\lambda}}^{(z)}) = \sum_{m=1}^{M} P(m \mid x_t, \hat{\boldsymbol{\lambda}}^{(z)}) p(y_t \mid x_t, m, \hat{\boldsymbol{\lambda}}^{(z)})$$

$$p(y_t \mid x_t, m, \boldsymbol{\lambda}^{(z)}) = \mathcal{N}(y_t \,;\, \mu_m^{(y|x_t)}, \Sigma_m^{(y|x_t)})$$

$$\mu_m^{(y|x_t)} = \mu_m^{(y)} + \Sigma_m^{(yx)} \Sigma_m^{(xx)^{-1}} (x_t - \mu_m^{(x)})$$

$$\Sigma_m^{(y|x_t)} = \Sigma_m^{(yy)} - \Sigma_m^{(yx)} \Sigma_m^{(xx)^{-1}} \Sigma_m^{(xy)}$$

4. Predict tgt speech feats from cond PDF by MMSE

$$\hat{y}_t = \sum_{m=1}^{M} P(m \mid x_t, \hat{\boldsymbol{\lambda}}^{(z)}) \, \mu_m^{(y|x_t)}$$

⇒ Discontinuity due to frame-by-frame mapping

3. GMM-based VC w/ dyn feats [Toda;'07]

Use static & dynamic *experts* for conversion

1. Both static & dynamic feats in joint vectors

$$\Delta x_t = x_{t+1} - x_t \qquad \Delta y_t = y_{t+1} - y_t$$
$$\boldsymbol{z}_t = [x_t, y_t, \Delta x_t, \Delta y_t]^\top$$

2. Model joint PDF between src & tgt data by GMM

$$\boldsymbol{\mu}_m^{(z)} = \left[\mu_m^{(x)} \quad \mu_m^{(y)} \quad \mu_m^{(\Delta x)} \quad \mu_m^{(\Delta y)}\right]^\top$$

$$\boldsymbol{\Sigma}_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} & 0 & 0 \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} & 0 & 0 \\ 0 & 0 & \Sigma_m^{(\Delta x \Delta x)} & \Sigma_m^{(\Delta x \Delta y)} \\ 0 & 0 & \Sigma_m^{(\Delta y \Delta x)} & \Sigma_m^{(\Delta y \Delta y)} \end{bmatrix}$$

3. Derive static/dynamic *experts* given $\boldsymbol{x}$ from joint PDF

Static: $\mathcal{N}(y_t \,;\, \mu_{\hat{m}_t}^{(y|x_t)}, \Sigma_{\hat{m}_t}^{(y|x_t)})$

Dynamic: $\mathcal{N}(\Delta y_t \,;\, \mu_{\hat{m}_t}^{(\Delta y|\Delta x_t)}, \Sigma_{\hat{m}_t}^{(\Delta y|\Delta x_t)})$

4. Predict tgt speech feats while satisfying both experts

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \prod_{t=1}^{T} \Big\{ \mathcal{N}(y_t \,;\, \mu_{\hat{m}_t}^{(y|x_t)}, \Sigma_{\hat{m}_t}^{(y|x_t)}) \; \mathcal{N}(\Delta y_t \,;\, \mu_{\hat{m}_t}^{(\Delta y|\Delta x_t)}, \Sigma_{\hat{m}_t}^{(\Delta y|\Delta x_t)}) \Big\}$$

4. Gaussian Processes

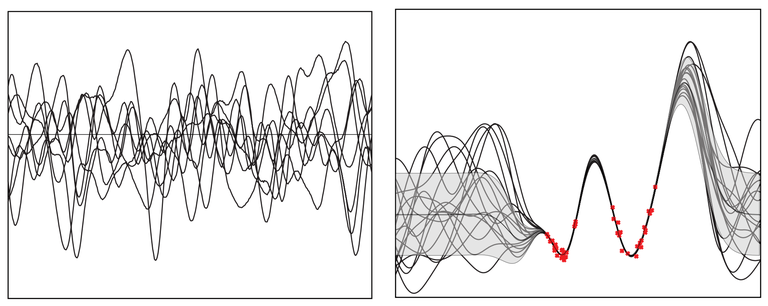
Typical form for parametric prediction models

$$y_t = f(x_t; \boldsymbol{\lambda}) + \epsilon \qquad \epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$
$$p(y_t \mid x_t, \boldsymbol{\lambda}) = \mathcal{N}\left(y_t \,;\, f(x_t; \boldsymbol{\lambda}), \sigma^2\right)$$

⇒ Instead, use non-parametric kernel-based model

· Gaussian Process (GP)
- Probabilistic Bayesian model
- Distribution over functions
- Cov matrix is given by Gramian matrix via kernel func
- Non-parametric model

⇒ Unnecessary to set parametric form a priori



Samples from GP prior — Samples from GP posterior

· GP regression
- Mapping function is now a sample from GP

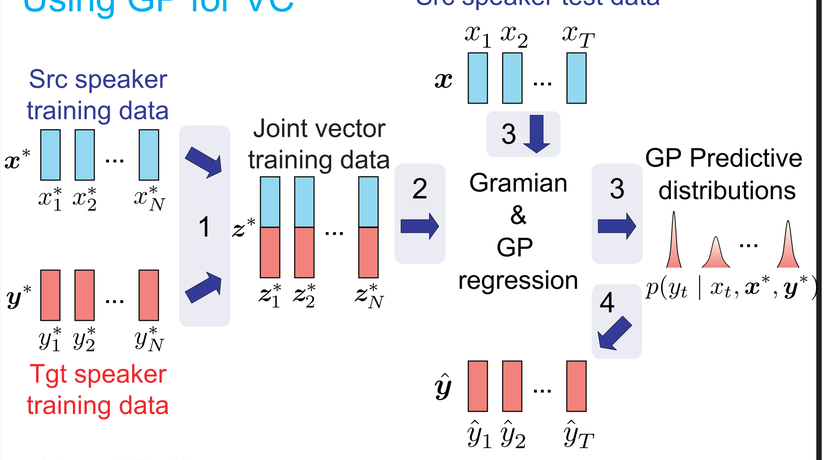
$$f(x; \boldsymbol{\lambda}) \sim \mathcal{GP}\left(m(x), k(x, x')\right)$$
$m(x)$: mean function
$k(x, x')$: covariance (kernel) function

- GP predictive distribution

$$p(y_t \mid x_t, \boldsymbol{x}^*, \boldsymbol{y}^*) = \mathcal{N}\left(y_t \,;\, \mu(x_t), \Sigma(x_t)\right)$$
$$\mu(x_t) = m(x_t) + \boldsymbol{k}_t^\top [\boldsymbol{K}^* + \sigma^2 \boldsymbol{I}]^{-1} (\boldsymbol{y}^* - \boldsymbol{\mu}^*)$$
$$\Sigma(x_t) = k(x_t, x_t) + \sigma^2 - \boldsymbol{k}_t^\top [\boldsymbol{K}^* + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{k}_t$$
$$\boldsymbol{\mu}^* = \left[m(x_1^*) \quad m(x_2^*) \quad \ldots \quad m(x_N^*)\right]^\top$$
$$\boldsymbol{K}^* = \begin{bmatrix} k(x_1^*, x_1^*) & k(x_1^*, x_2^*) & \ldots & k(x_1^*, x_N^*) \\ k(x_2^*, x_1^*) & k(x_2^*, x_2^*) & \ldots & k(x_2^*, x_N^*) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N^*, x_1^*) & k(x_N^*, x_2^*) & \ldots & k(x_N^*, x_N^*) \end{bmatrix}$$
$$\boldsymbol{k}_t = \left[k(x_1^*, x_t) \quad k(x_2^*, x_t) \quad \ldots \quad k(x_N^*, x_t)\right]^\top$$

5. GP-based VC

Using GP for VC



· Use GP directly
- MMSE estimates of tgt speech feats

$$\hat{y}_t = \mu(x_t)$$

⇒ Discontinuity due to frame-by-frame mapping

· Use static & dynamic GP experts
- GP experts for static & dynamic feats

Static: $y_t \sim \mathcal{N}(\mu(x_t), \Sigma(x_t))$

Dynamic: $\Delta y_t \sim \mathcal{N}(\mu(\Delta x_t), \Sigma(\Delta x_t))$

- Predict tgt speech feats

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \prod_{t=1}^{T} \{\mathcal{N}\left(y_t \,;\, \mu(x_t), \Sigma(x_t)\right) \; \mathcal{N}\left(\Delta y_t \,;\, \mu(\Delta x_t), \Sigma(\Delta x_t)\right)\}$$

- Predictive distribution is still Gaussian

⇒ Maximization can be solved efficiently using the speech parameter generation algorithm

6. Implementation

Covariance & mean functions

· Covariance (kernel) function
- Measure describing the local covariance
- e.g., linear, squared exponential, polynomial

· Mean function
- Describe mean characteristics
- e.g., $m(x)=0$, $m(x)=b$, $m(x)=ax$, $m(x)=ax+b$
- We used $m(x)=ax+b$

Partitioning input space

- Inverting Gramian matrix (N x N) is memory & computationally intractable if amount of data is large
- Partition input space by LBG (e.g., 32 sub-spaces)

7. Experiment

Conditions

Database	CMU ARCTIC speech database
Training data	Speakers CLB & SLT, 50 utterances (CLB⇒SLT) 34,664 frames
Test data	50 utterances not included in training data
Sampling freq.	16 kHz
Frame shift	5-ms
Feature vec	0--40 mel-cepstral coefficients, Δ & ΔΔ
Methods	GMM w/o dynamic feats, GMM w/ dynamic feats, trajectory GMM, GP w/o dynamic features, GP w/ dynamic features
Mapping	Spectrum: Mapping by probabilistic model F0: Linearly transformed in the log domain

Objective evaluation (mel-cepstral distortions)

GP-based VC w/ various covariance (kernel) functions

Covariance functions	w/o dyn.	w/ dyn.
Linear	3.96	**4.15**
Linear+ARD	**3.95**	**4.15**
Matern	4.96	5.99
Neural network	4.96	5.95
Polynomial	4.95	5.80
Piecewise polynomial isotoropic	4.96	6.00
Rational quadratic+ARD	4.96	5.98
Rational quadratic isotropic	4.96	5.98
Squared exponential+ARD	4.95	5.98
Squared exponential isotropic	4.95	5.98

Performance of conventional approaches

#mix	GMM w/o dyn.	GMM w/ dyn.	Trajectory GMM
2	5.97	5.95	5.90
4	5.75	5.82	5.81
8	5.66	5.69	5.63
16	5.56	5.59	5.52
32	5.49	5.53	5.45
64	5.43	5.45	5.38
128	5.40	5.38	**5.33**
256	**5.39**	5.35	5.35
512	5.41	**5.33**	5.42
1,024	5.50	5.34	5.64