Joint Uncertainty Decoding for Noise Robust Speech Recognition

Jasha Droppo, Alex Acero, and Li Deng
Microsoft Research, One Microsoft Way, Redmond, Washington, USA
Outline

• What is robust?
• Feature-domain & model-domain
• Introduction to SPLICE
• What is uncertainty?
• Concept of uncertainty decoding
• Uncertainty Decoding with SPLICE
• Uncertainty with Joint Uncertainty Decoding
What is robust?

- We can say it is robust if it is hardly affected by extrinsic events.
  - Ex: A waterproof watch in water can still work as usual.

- For ASR
  - Speech recognition performance degrades in the presence of environmental noise, why?
    
    The answer is the mismatch between training and test condition.

- Solution
  - There are tow main direction using different aspect to cut into this problem.
Feature-domain & Model-domain

- Where is part of feature-domain or model-domain?

Feature-domain: The outputs from feature extraction are all restored feature vectors without any other information to adapt acoustic model.

Model-domain: The outputs from feature extraction may be not feature vectors but contains other information (ex: likelihood) to adapt acoustic model.
Introduction to SPLICE

- Stereo piecewise linear compensation for environment (SPLICE) takes advantage of seamlessly integrating into existing system, without a complete overhaul of existing code.

\[ \tilde{y}_t = \tilde{x}_t = E[x_t | y_t] = \sum_k p(k | y_t) E[x_t | y_t, k] \]

- Assuming that the difference between clean data and corrupted data can be compensated by each single Gaussian providing a linear compensation.

\[ E[x_t | y_t, k] \approx y_t + r_k \]

\[ \tilde{y}_t = y_t + \sum_k p(k | y_t) \cdot r_k \]

\[ \hat{k} = \arg \max_k p(k | y_t) \]

\[ p(k | y_t) = \frac{p(y_t | k) p(k)}{\sum_{k'} p(y_t | k') p(k')} \]

During training time:

- Probabilistic correction via the following equations:

\[ r_k = \frac{1}{T-1} \sum_{t=1}^{T-1} p(k | y_t) (x_t - y_t) \]

\[ \hat{k} = \arg \max_{\hat{k}} p(k | y_t) \]

\[ \hat{y}_t = y_t + r_{\hat{k}} \]
What is uncertainty?

- Feature compensation **without** uncertainty
  - The corrupted speech is restored by compensation and sent into decoder. The \( \hat{x} \) is viewed as the clean feature, is that right?

- Feature compensation **with** uncertainty
  It is intuitively reasonable to incorporate with **uncertain observation**.

Noticing here, it is the key idea adjusted by SPLICE and JUN uncertainty decoding to make process efficient.
Concept of uncertainty decoding

- Model-compensation
  - Renewing acoustic model for the specific noise.
  - The input is either the corrupted speech data or the data combined clean and corrupted speech to achieve this goal.

- Computationally expensive
How to design uncertainty?

- Noise robustness DBN

- Corrupted speech likelihood given by

\[ p(y_t|\tilde{M}, \theta_t) = \int p(y_t|x_t, M) p(x_t|M, \theta_t) dx_t \quad (1) \]

\[ p(y_t|x_t, M) = \int p(y_t|x_t, n_t) p(n_t|\tilde{M}, \theta^n_t) dn_t \quad (2) \]

- Only \( p(y_t | x_t, \tilde{M}) \) depend on noise.

- Efficient approximation emerges from above formulation.
  - Independent of clean model complexity.
  - Appropriate form for integration.
Appendix A for (1)

marginalise

\[ p(A \mid D) = \int p(A, B \mid D) d_B \]

\[ p(y_t \mid M, \tilde{M}, \theta_t) = \int p(y_t, x_t \mid M, \tilde{M}, \theta_t) d_{x_t} \]

\[ = \int p(y_t \mid x_t, M, \tilde{M}, \theta_t) p(x_t \mid M, \tilde{M}, \theta_t) d_{x_t} \]

\[ = \int p(y_t \mid x_t, \tilde{M}) p(x_t \mid M, \theta_t) d_{x_t} \]
Appendix B for (2)

\[ P(y_{t} \mid x_{t}, \tilde{M}) = \int p(y_{t}, n_{t} \mid x_{t}, \tilde{M})d_{n_{t}} \]

\[ = \int p(y_{t} \mid n_{t}, x_{t}, \tilde{M})p(n_{t} \mid x_{t}, \tilde{M})d_{n_{t}} \]

\[ = \int p(y_{t} \mid n_{t}, x_{t})p(n_{t} \mid \tilde{M}, \theta_{t}^{n})d_{n_{t}} \]
What’s difference of decoding between SPLICE & JUD

- Passing conditional probability to decoding
- Tow form of uncertainty decoding
  - Splice with uncertainty \( p(y_t|x_t,\tilde{M}) \) by Bayes’ rule
  - Joint distribution \( p(y_t|x_t,\tilde{M}) \) by joint probability
- Both are based on Gaussian mixture model
  - Using different approximation to make process efficient
Uncertainty decoding with SPLICE

Splice with uncertainty decoding uses Bayes’ rule to write GMM as

\[ p(y_t \mid x_t, \tilde{\mathcal{M}}) = \sum_{n=1}^{N} \left( \frac{p(x_t \mid y_t, \tilde{s}_n, \tilde{\mathcal{M}}) p(y_t \mid \tilde{s}_n, \tilde{\mathcal{M}}) \tilde{c}_n}{p(x_t \mid \tilde{\mathcal{M}})} \right) \]

- \( p(x_t \mid y_t, \tilde{s}_n, \tilde{\mathcal{M}}) \) related to standard SPLICE estimate
- Denominator \( p(x_t \mid \tilde{\mathcal{M}}) \) is a GMM – simplify using a single Gaussian
Appendix C for (3)

\[ p(y_i | x_i, \bar{M}) = \sum_{n=1}^{N} p(y_i | \bar{s}_n, x_i, \bar{M}) p(\bar{s}_n | x_i, \bar{M}) \]

\[ = \sum_{n=1}^{N} \frac{p(x_i, y_i | \bar{s}_n, \bar{M}) p(\bar{s}_n | x_i, \bar{M})}{p(x_i | \bar{s}_n, \bar{M})} \]

\[ = \sum_{n=1}^{N} \frac{p(x_i | \bar{s}_n, y_i, \bar{M}) p(y_i | \bar{s}_n, \bar{M}) p(\bar{s}_n | x_i, \bar{M})}{p(x_i | \bar{s}_n, \bar{M})} \]

\[ = \sum_{n=1}^{N} \frac{p(x_i | \bar{s}_n, y_i, \bar{M}) p(y_i | \bar{s}_n, \bar{M}) p(\bar{s}_n | x_i, \bar{M})}{p(x_i | \bar{M})} \]

\[ = \sum_{n=1}^{N} \frac{p(x_i | \bar{s}_n, y_i, \bar{M}) p(y_i | \bar{s}_n, \bar{M}) c_n}{p(x_i | \bar{M})} \]
Uncertainty with SPLICE

• Standard SPLICE uses

\[ \hat{x}_t = \mathbb{E}[x_t | y_t] = \sum_k P(k | y_t) \mathbb{E}[x_t | y_t, k] \]

Replace \( k \) with \( s \)

\[ = \sum_{n=1}^{N} P(\tilde{s}_n | y_t, \tilde{M}) \int_{x_t} x_t P(x_t | y_t, \tilde{s}_n, \tilde{M}) \, dx_t \]

• Uncertainty with SPLICE uses Bayes’ rule to write GMM as:

\[
P(y_t | x_t, \tilde{M}) = \sum_{n=1}^{N} \left( \frac{P(x_t | y_t, \tilde{s}_n, \tilde{M}) P(y_t | \tilde{s}_n, \tilde{M}) c_n}{p(x_t | \tilde{M})} \right) \]

\[
p(y_t | x_t, \tilde{M}, \tilde{s}_n) = f(y_t, \tilde{s}_n) N(A^{(n)} y_t + b^{(n)}; x_t, \sum_b^{(n)}) \]

\[
\tilde{s}_n = \arg \max \left( \frac{c_n p(y_t | \tilde{s}_n, \tilde{M})}{\sum_{i=1}^{N} c_i p(y_t | \tilde{s}_i, \tilde{M})} \right) \]

\[
p(y_t | M, \tilde{M}, \theta) \propto \sum_{m \in \theta_t} c_m N(A^{(n^*)} y_t + B^{(n^*)}; \mu^{(m)}, \sum^{(m)} + \sum_b^{(n^*)}) \]
Uncertainty decoding with JUD

- Joint distribution $p(x, y)$

When SNR high, the conditional is deterministic.
When SNR low, the conditional is Gaussian
Uncertainty decoding with JUN

- GMM is a standard approach to handle complex distribution
  - It's simple to marginalise two Gaussians
- Using approximation front-end compensation model $\tilde{M}$

\[
p(y_t | x_t, \tilde{M}) \approx \sum_{n=1}^{N} p(\tilde{s}_n | x_t, \tilde{M}) p(y_t | x_t, \tilde{s}_n, \tilde{M})
\]

- Only $\tilde{M}$ is a function of noise.

- Some issues need to be handled with
  - Component posterior $p(\tilde{s}_n | x_t, \tilde{M})$ is a function of clean speech
  - Component compensation parameters $p(y_t | x_t, \tilde{s}_n, M)$
  - Direct use increases number of components
Uncertainty decoding for JUD

- Joint uncertainty decoding uses the GMM directly,

\[
p(y_t \mid x_t, \tilde{M}) = \sum_{n=1}^{N} p(y_t \mid \tilde{s}_n, x_t, \tilde{M}) p(\tilde{s}_n \mid x_t, \tilde{M})
\]

\[
= \sum_{n=1}^{N} \frac{p(x_t, y_t \mid \tilde{s}_n, \tilde{M}) p(\tilde{s}_n \mid x_t, \tilde{M})}{p(x_t \mid \tilde{s}_n, \tilde{M})}
\]

but

- Approximates the component posterior of clean speech, using the corrupted speech:

\[
p(\tilde{s}_n \mid x_t, \tilde{M}) \approx p(\tilde{s}_n \mid y_t, \tilde{M})
\]

- This decouples the front-end distribution from being dependent on the acoustic model through the clean speech variable

- conditional probability derived from the joint distribution

\[
p(x_t, y_t \mid \tilde{s}_n, \tilde{M}) = N\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix} ; \begin{bmatrix} \tilde{\mu}_x^{(n)} \\ \tilde{\mu}_y^{(n)} \end{bmatrix} , \begin{bmatrix} \sum_{xx}^{(n)} & \sum_{xy}^{(n)} \\ \sum_{xy}^{(n)} & \sum_{yy}^{(n)} \end{bmatrix} \right)
\]

covariance matrix is usually made diagonal for efficiency
Uncertainty decoding for JUD

- Both uncertainty decoding schemes yield same decoding form:

\[ p(y_y \mid M, \tilde{M}, \theta_t) \approx \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha^{(mn)} N(A^{(n)} \gamma_t + b^{(n)}; \mu^{(m)}, \Sigma^{(m)} + \Sigma^{(n)}) \]

- Form of \( A^{(n)}, b^{(n)} \) and \( \Sigma^{(n)} \) differ in the two cases

- For JUN that would be:

\[
\begin{align*}
A^{(n)} &= \sum_x^{(n)} \sum_{yx}^{(n)-1} \\
b^{(n)} &= \mu_x^{(n)} - A^{(n)} \mu_y^{(n)} \\
\Sigma_b^{(n)} &= A^{(n)} \sum_y^{(n)} A^{(n)T} - \sum_x^{(n)}
\end{align*}
\]

- To improve efficiency only a single front-end component selected, for Joint based on \( p(\tilde{s}_n \mid y_t, \tilde{M}) \)

- Compared to model-based compensation computational cost is:
  - only a function of the \( N \),
  - Not the number of components in clean speech model through variance bias must be applied