공학석사학위논문

Speech Enhancement using Gaussian Process and Relevance Vector Machine

가우시안 프로세스와 Relevance Vector Machine을 이용한 데이터주도 기반 음성 향상

2014 년 8 월

서울대학교 대학원

전기·컴퓨터 공학부

수 카 냐

Speech Enhancement using Gaussian Process and Relevance Vector Machine

가우시안 프로세스와 Relevance Vector Machine을 이용한 데이터주도 기반 음성 향상

지도교수 김 남 수

이 논문을 공학석사 학위논문으로 제출함

2014 년 08 월

서울대학교 대학원

전기·컴퓨터 공학부

수 카 냐

수카냐의 공학석사 학위논문을 인준함

2014 년 08 월

위 원 장 김 성 철 (인)

부위원장 김 남 수 (인)

위 원 조 남 익 (인)

Abstract

This thesis presents a novel data-driven approach to single channel speech enhancement employing Gaussian process (GP) and relevance vector machine (RVM). The residual gain is defined as the difference between the optimal gain and that obtained from the minimum mean square error log-spectral amplitude (MMSE-LSA) estimator, the latter being one of the most popular spectral enhancement approaches. GP and RVM are applied to model and learn the relationship between the input features, which are the a priori and a posteriori signal-to-noise ratios (SNRs), and the outputs corresponding to the residual gains. The residual gain is predicted for each frequency bin separately using a different GP or RVm model. The proposed approach consists of two stages. In the first stage, the gain of the MMSE-LSA estimator is calculated in conjunction with the SNR features. In the second stage, the residual gains are estimated through GP or RVM and they are used to further enhance the output of the MMSE-LSA module. Experimental results show that the proposed approach produces better speech quality than not only the MMSE-LSA enhancement module but also the other data driven technique. We also extend our setting to the multi-task case where the residual gain is estimated jointly for a group of frequency bins. As expected, in the multi-task case, the enhancement performance is better than the case where the residual gain is estimated for each frequency bin using GP or RVM.

Keywords: Speech enhancment, Data-driven process, Gaussian process (GP), Relevance vector machine (RVM), Multi-task GP

Student Number: 2012-23955

Contents

List of Figures

Chapter 1

Introduction

Statistical model-based speech enhancement techniques have been widely applied to enhance the quality and intelligibility of the input speech corrupted by background noises [1, 2, 3]. Recently, a number of data-driven approaches have been proposed to further improve the performance of the traditional statistical model-based techniques [4, 5, 6]. For example, Fingscheidt et al. [5] proposed applying a look-up table indexed by the a priori and a posteriori signal-to-noise ratio (SNR) values to determine the weighting rules for noisy speech spectral amplitudes. Jin et al. [6] applied a codebook to predict the residual gain which they defined as the log-difference between the optimal gain and the gain derived from a statistical model-based algorithm. Park et al. [7] proposed a time domain approach employing Gaussian process (GP) regression to estimate the clean speech samples based on the past and present noisy samples.

In most of the proposed data-driven approaches, the a priori and a posteriori SNRs, which turn out to be important parameters in determining the gain in statistical model-based speech enhancement, are used as input features. Based on this, we

can treat the problem of finding an optimal gain in a data-driven speech enhancement technique as a regression task where the gain is predicted conditioned on the given a priori and a posteriori SNRs. In this respect, the conventional statistical model-based technique can be thought of as a feature extractor for the subsequently applied regressors.

In this work we present a data-driven approach towards speech enhancement which is based on predicting the optimal gain as a function of the SNRs. In the next section we will show that the task of finding the optimal gain is equivalent to that of finding the residual gain, which we define as the difference between the optimal gain and the gain derived from a statistical model-based algorithm. We call the latter as the preliminary gain. Our problem statement is thus reformulated as predicting the residual gain using the SNRs as input features.

Our proposed approach consists of two stages. In the first stage, the feature vector relevant to the SNRs is extracted and the preliminary gain is calculated. In the second stage, the residual gain is predicted based on the feature vector extracted from the first stage. The final gain is then obtained by adjusting the preliminary gain derived from the first stage with the predicted residual gain. For predicting the residual gain, we adopt two prevalent regression techniques: GP [9] and relevance vector machine (RVM) [10] which are powerful supervised learning approaches extensively used for regression problems in a wide range of areas [19, 20]. Both methods are kernel-based Bayesian regression algorithms which allow the data to be mapped into a high-dimensional space thereby capturing the relationship between the input and output variables in a more efficient manner. Experimental results show that the proposed method produces better speech quality than the conventional enhancement techniques. We also extend the our setting to the multi-task case where the residual gain is estimated jointly for a group of frequency bins. As expected, in the multi-task case, the enhancement performance is better than the case where the residual gain is

estimated for each frequency bin using GP or RVM.

Chapter 2

Residual Gain based Speech Enhancement System

2.1 Residual Gain

Let $X(k,l)$, $Y(k,l)$ and $D(k,l)$ denote the short term Fourier transform (STFT) coefficients of the clean speech, noisy speech and the background noise, respectively for a frequency index k and time-frame l. If we assume that the noise is additive and uncorrelated with the clean speech, then we have

$$Y(k,l) = X(k,l) + D(k,l). \tag{2.1}$$

The conventional statistical model-based speech enhancement techniques assume a family of parametric models for the distribution of the clean speech and noise spectra. They then find a gain $\hat{G}(k,l)$ which is optimal under some criterion such that the clean speech estimate $\hat{X}(k,l)$ can be derived by

$$\hat{X}(k,l) = \hat{G}(k,l)Y(k,l). \tag{2.2}$$

The minimum mean square error log-spectral amplitude (MMSE-LSA) estimator [1] is one such statistical approach which is most popular. The approach is based on minimizing the mean-square error of the log-spectra, assuming a Gaussian statistical model for both speech and noise. The gain in this case is given by

$$\hat{G}(k,l) = \frac{\xi(k,l)}{1+\xi(k,l)} \exp\left(\frac{1}{2}\int_{\nu(k,l)}^{\infty} \frac{e^{-t}}{t} \mathrm{d}t\right) \tag{2.3}$$

where $\nu(k,l) = \frac{\gamma(k,l)\xi(k,l)}{1+\xi(k,l)}$ with $\xi(k,l)$ and $\gamma(k,l)$ denoting the a priori and a posteriori SNRs, respectively. It should be noted that in (2.3), the gain is given as a function of $\xi(k,l)$ and $\gamma(k,l)$. Even though this estimator is optimal in the mean square sense, its optimality can be easily broken due to mismatches and inaccuracies in distribution modeling, noise estimation or SNR estimation.

Let us call the gain $\hat{G}(k,l)$ as the **preliminary gain**. Also let $G(k,l)$ denote the **optimal gain** such that the actual clean speech spectrum $X(k,l)$ turns out to be

$$X(k,l) = G(k,l)Y(k,l). \tag{2.4}$$

As mentioned previously, due to the modeling and estimation inaccuracies, $\hat{G}(k,l)$ in (2.3) usually deviates from $G(k,l)$. Let the **residual gain** $H(k,l)$ be defined as

$$H(k,l) = G(k,l) - \hat{G}(k,l). \tag{2.5}$$

$H(k,l)$ thus measures the deviation of $\hat{G}(k,l)$ from $G(k,l)$. *A positive $H(k,l)$ implies that $\hat{G}(k,l)$ under-estimates the corresponding speech component, while a negative $H(k,l)$ results in an over-estimated speech component.*

This is depicted in Figure 2.1 which shows the spectrograms of a clean utterance, the noisy utterance enhanced by the MMSE-LSA estimator at 15 dB white noise environment and the corresponding residual gain matrix thresholded at zero. The bright areas in the residual gain matrix depict a positive value of the residual gain while

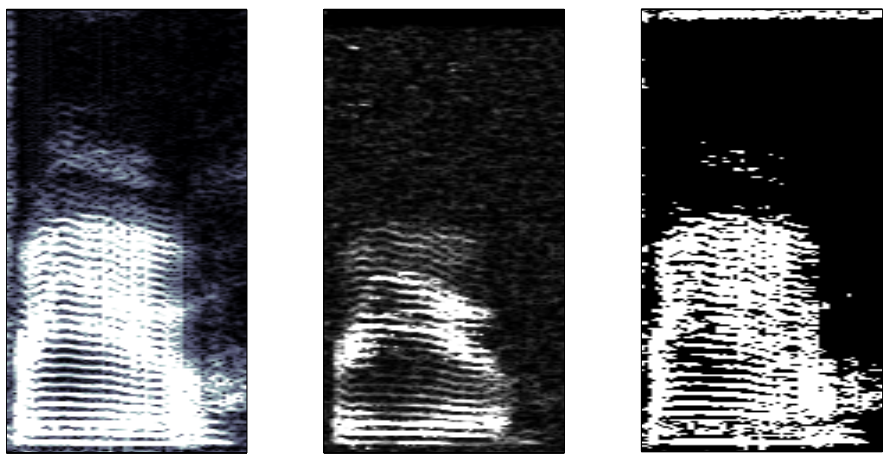

Figure 2.1 Spectrogram of a clean utterance (left). Spectrogram of the noisy utterance enhanced by MMSE-LSA estimator at 15 dB white noise environment (middle). The corresponding residual gain matrix thresholded at zero (right).

the dark areas depict a negative residual gain. From the figure we can observe that for the low and mid frequency bins, the MMSE-LSA estimator under-estimates the speech components and speech distortion occurs which is why the corresponding residual gain is mostly positive. In the high frequency bins residual noise exists in the MMSE-LSA enhanced speech due to which the residual gain values in these bins are mostly negative.

If we combine (2.4) and (2.5), $X(k,l)$ can be expressed in terms of $H(k,l)$ through the following relation

$$X(k,l) = [H(k,l) + \hat{G}(k,l)]Y(k,l). \tag{2.6}$$

To estimate $X(k,l)$, the task of predicting $G(k,l)$ thus reduces to the task of predicting $H(k,l)$. The approach thus being an error-driven approach has 2 stages : In the first stage we estimate the preliminary gain using the conventional statistical tech-

nique while in the second stage we estimate the residual gain and finally combine these two gains using (2.6) to estimate the clean speech.

In this work we regard the task of estimating the residual gain as a regression task while treating the a priori and a posteriori SNRs as input features. We use regression to predict the residual gain rather than deriving a closed form expression for the same because the latter involves several modeling assumptions. The regression technique on the other hand does not make much assumptions about the distribution of speech and noise data and finds the unknown function that best describes the relationship between the residual gain and the SNR features. The regression techniques we use are GP and RVM regression.

2.2 Feature Extraction and Pre-processing

The feature extraction process is depicted in Figure 2.2. To construct the feature vector $\tilde{\mathbf{z}}(k,l)$ corresponding to a point (k,l) in the frequency-time grid, the a priori and a posteriori SNRs are each collected over a rectangular spectro-temporal window which incorporates frequency and temporal components with their respective indexes varying from $k-M_w$ to $k+M_w$ and $l-N_w+1$ to l as in [6]. This renders $\tilde{\mathbf{z}}(k,l)$ as

$$\begin{aligned}\tilde{\mathbf{z}}(k,l) = \big[&\xi(k-M_w, l-N_w+1)\ldots\xi(k-M_w,l)\\ &\ldots\xi(k+M_w,l-N_w+1)\ldots\xi(k+M_w,l)\\ &\gamma(k-M_w,l-N_w+1)\ldots\gamma(k-M_w,l)\\ &\ldots\gamma(k+M_w,l-N_w+1)\ldots\gamma(k+M_w,l)\big]^T\end{aligned} \tag{2.7}$$

where the dimension of $\tilde{\mathbf{z}}(k,l)$ is $2(2M_w+1)N_w$ and the superscript T denotes matrix or vector transpose.

The grouping of the neighboring SNR features in (2.7) takes into account the high spectral and temporal correlations inherent in speech signals. The components of the vector $\tilde{\mathbf{z}}(k,l)$ are thus highly correlated. This allows us to further reduce the dimen-

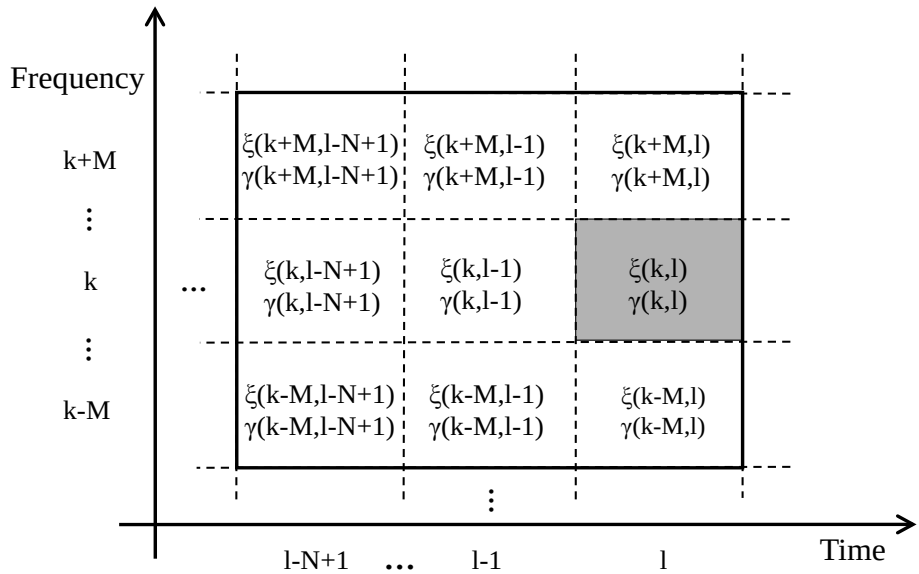

Figure 2.2 Feature extraction process for a point (k,l) in the time-frequency grid. The a priori and a posteriori SNR features are collected over a rectangular window of size $(2M_w + 1)N_w$.

sion of $\tilde{\mathbf{z}}(k,l)$ without much loss of information leading to a comparatively compact statistical representation. For this, we apply principal component analysis (PCA) to $\{\tilde{\mathbf{z}}(k,l)\}$ which results in the compact features $\{\mathbf{z}(k,l)\}$ with lower dimensionality. In this work, the dimension is reduced from $2(2M+1)N$ to d which determines the input dimensionality of the GP. In the remaining part of this paper, for simplicity, we will replace the notations $\mathbf{z}(k,l)$ and $H(k,l)$ with $\mathbf{z}_{kl}$ and H_{kl} respectively.

2.3 Overall System

Finally, the proposed speech enhancement system is described using a block diagram in Figure 2.3. For each frequency bin, the SNR feature vectors of the training examples are clustered into N_c clusters in the training phase. This is done by using

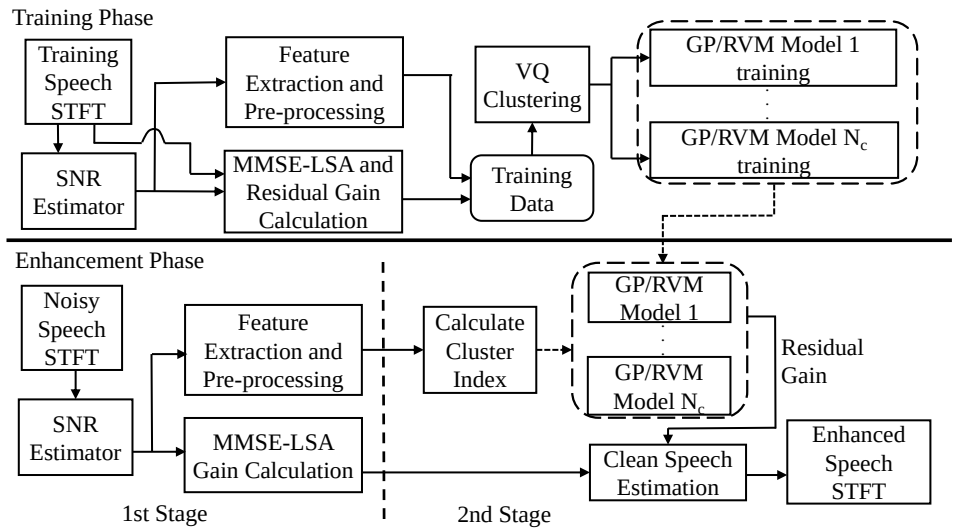

Figure 2.3 A block diagram of the proposed speech enhancement system using GP and RVM.

Vector Quantization (VQ). Then for each cluster, a regressor is trained by treating the residual gain values corresponding to the SNR feature vectors in the cluster, as the target for prediction. During the enhancement phase, a test feature vector for each frequency bin is first assigned to one of the N_c clusters in the same way as the training data is clustered. Finally, the corresponding residual gain is predicted by using the regressor belonging to the assigned cluster.

In our work, the values of M_w and N_w in (2.7) were respectively set as 1 and 5, which resulted in 30-dimensional feature vectors. This original dimension was further reduced to $d = 10$ with the help of PCA. These feature vectors for each frequency bin were then clustered into $N_c = 64$ clusters using VQ technique. In the clustering technique using VQ, the codebook of N_c codewords was learned and the SNR feature vectors for each frequency bin were clustered by applying the Linde-Buzo-Gray(LBG) algorithm.

Let $D_k^m = \{(\mathbf{z}_{ki}^m, H_{ki}^m) \mid i = 1, \ldots, N\}$ denote the training set corresponding to

the k^{th} frequency bin assigned to the m^{th} cluster. Both inputs and outputs are aggregated into vectors $\mathbf{Z}_k^m = [\mathbf{z}_{k1}^m \cdots \mathbf{z}_{kN}^m]^T$ and $\mathbf{H}_k^m = [H_{k1}^m \cdots H_{kN}^m]^T$, respectively. We assume, without loss of generality, that during the enhancement phase the test feature $\mathbf{z}_{kl}^*$ is assigned to the m^{th} cluster. This implies that the GP or RVM trained for the m^{th} cluster is used to predict the test output H_{kl}^*. In the following chapters dedicated to the review of GP and RVM, we will denote the input $\mathbf{Z}_k^m$ by $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]^T$ and the output $\mathbf{H}_k^m$ by $\mathbf{Y} = [y_1 \cdots y_N]^T$ for a better understanding of each method described using more general terminology.

Chapter 3

Speech Enhancement using Gaussian Process (GP) Regression

3.1 GP Model

Assuming that D_k^m is drawn from a noisy process, the signal model using GP is defined as

$$y_i = f(\mathbf{x}_i) + \eta_i \tag{3.1}$$

where η_i is a zero-mean Gaussian random variable with variance σ^2 and $f(\cdot)$ is an unknown latent function. A GP imposes a Gaussian prior over the unknown latent function f. Using the noise term, the joint distribution of the training output $\mathbf{Y}$ and the latent function value f^* at the test input $\mathbf{x}^*$ under the GP prior can thus be written as

$$\begin{bmatrix} \boldsymbol{Y} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I} & K(\mathbf{X},\mathbf{x}^*) \\ K(\mathbf{x}^*,\mathbf{X}) & K(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix}\right) \tag{3.2}$$

where $\mathbf{I}$ denotes the identity matrix and the $N \times N$ matrix $K(\mathbf{X},\mathbf{X})$ is the matrix of covariances evaluated at all pairs of training examples $\mathbf{X}$. Each element of $K(\mathbf{X},\mathbf{X})$

is given by

$$(K(\mathbf{X}, \mathbf{X}))_{ij} = \text{Cov}(f(\mathbf{x}_i), f(\mathbf{x}_j))$$

where $\text{Cov}(\cdot, \cdot)$ indicates the covariance. In a similar way, the N-length row vector $K(\mathbf{x}^*, X)$ represents the covariance between $\mathbf{x}^*$ and $\mathbf{X}$.

3.2 Predictions using GP

The GP predicts the function value for $\mathbf{x}^*$ by performing Bayesian inference as follows:

$$\mu^* = K(\mathbf{x}^*, \mathbf{X})\big[K\big(\mathbf{X}, \mathbf{X}\big) + \sigma^2\mathbf{I}\big]^{-1}\mathbf{Y} \tag{3.3}$$

where μ^* is the mean of the posterior distribution of f at $\mathbf{x}^*$. The above equation can also be written as

$$\sigma^* = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}^*, \mathbf{x}_i) \tag{3.4}$$

where $\boldsymbol{\alpha} = \big[K\big(\mathbf{X}, \mathbf{X}\big) + \sigma^2\mathbf{I}\big]^{-1}\mathbf{Y}$. Thus another way to look at this equation is that the posterior mean is given by the linear combination of N kernel points, each centered on a training point.

The test output y^* corresponding to the input $\mathbf{x}^*$ is then given by $y^* = \mu^*$. The GP also predicts the covariance σ^* of the posterior distribution of f at $\mathbf{x}^*$ which is given by

$$\sigma^* = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{X})\big[K\big(\mathbf{X}, \mathbf{X}\big) + \sigma^2\mathbf{I}\big]^{-1}K(\mathbf{X}, \mathbf{x}^*) \tag{3.5}$$

A GP is completely specified in terms of its mean and covariance functions. The mean function as described above is usually assumed to be zero without causing serious performance degradation. For the covariance function, we apply an isotropic

squared exponential kernel given by

$$\mathrm{Cov}(f(\mathbf{x}), f(\mathbf{x}')) = \delta^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')}{2l^2}\right) \tag{3.6}$$

where we need to specify two hyper-parameters: the signal variance δ and the scale parameter l. It should be noted that the scale parameter is the same for all the dimensions of the feature vector which avoids over-fitting for high dimensional features.

3.3 GP training

The hyper-parameters $\boldsymbol{\theta} = [\sigma\ \delta\ l]$ are trained by minimizing the negative log marginal likelihood of the training data, i.e. $-\log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})$ which is given by

$$-\log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{Y} + \frac{1}{2}\log|\mathbf{K}| + \frac{N}{2}\log 2\pi \tag{3.7}$$

where $\mathbf{K} = K\big(\mathbf{X}, \mathbf{X}\big) + \sigma^2\mathbf{I}$ and $|\cdot|$ denotes the matrix determinant. The three terms of the marginal likelihood in 3.7 have readily interpretable roles: the first term involves the data targets and is called the *data fit* term, the second term is the *negative complexity penalty* term depending only on the covariance function and the third term is a normalization constant.

The scale parameter l has a direct effect on the model complexity. A larger value of l results in the covariance function having a smaller curvature as the exponential term diminishes at a slower rate. Thus the model complexity decreases as it looses its flexibility and vice-versa. If we thus observe the effect of the three terms of the log marginal likelihood on l, we can observe from (3.7) that the data-fit term decreases while the negative complexity penalty term increases with l. This is depicted in Figures 3.1, 3.2 and 3.3 where the prediction is observed for three different values of l. From the figures we can observe that when l is too short then the GP prediction is more wiggly while if l is too big then the prediction is less flexible.

Intuitively this also makes sense because with increase in l the model becomes

less flexible which means that the model cannot fit the data well thereby resulting in the decrease in the data-fit term. On the other hand, a less flexible model implies a less complex model which has a low complexity penalty which results in the negative complexity penalty term being big.

In order to find the optimal hyper-parameters by maximizing the marginal likelihood, we seek the partial derivatives of the marginal likelihood with respect to the hyper-parameters. This involves inverting the $\boldsymbol{K}$ matrix which has a $\mathcal{O}(N^3)$ complexity unfavourable for large data-sets. It is this that the clustered approach comes handy. As explained earlier, we cluster our data and model a GP for each cluster. Thus effectively the number of training examples presented to the GP of 1 cluster is quite small and the computational burden is taken care of.

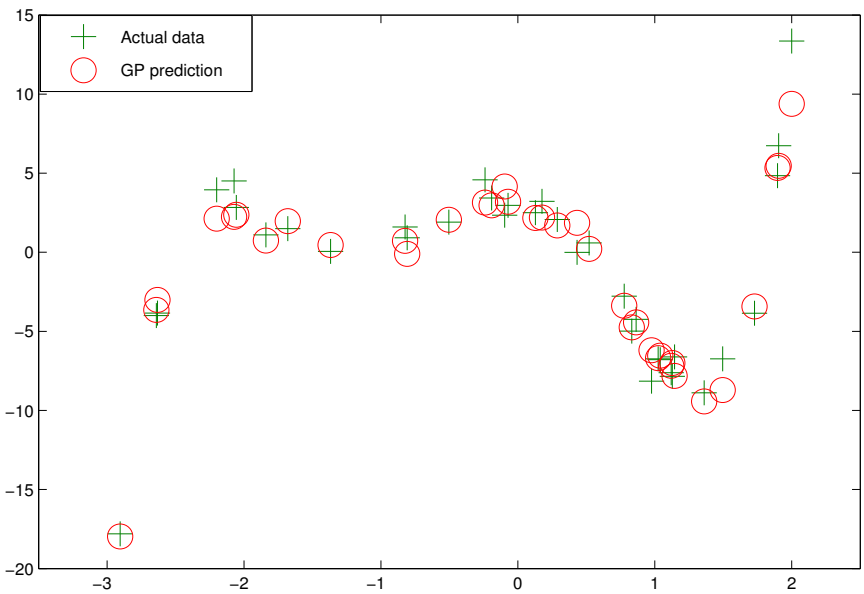

Figure 3.1 GP prediction with a scale parameter $l = 0.047$.

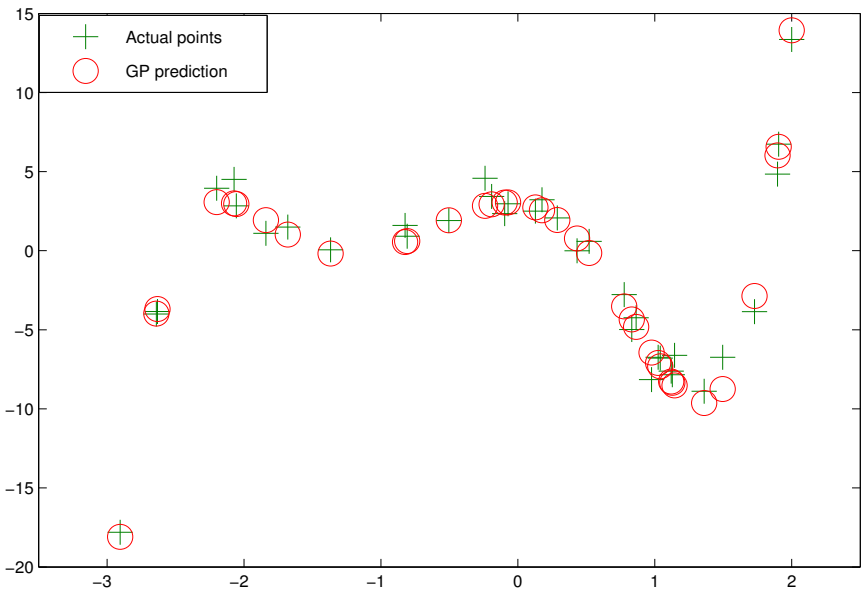

Figure 3.2 GP prediction with a scale parameter $l = 1.832$.

3.4 Experimental Setup

In this work, we implemented the GP algorithm using the GPML toolbox [16], which learns the GP hyper-parameters $\boldsymbol{\theta}$ and computes the posterior mean. The GP training in the toolbox was performed by maximizing the marginal likelihood using the method of conjugate gradients. The number of kernel functions involved is equal to the number of training examples N. The computational complexity for the a posteriori mean prediction is thus $\mathcal{O}(N)$ provided $\boldsymbol{K}$ is computed already. The experimental results obtained are presented in the following chapter.

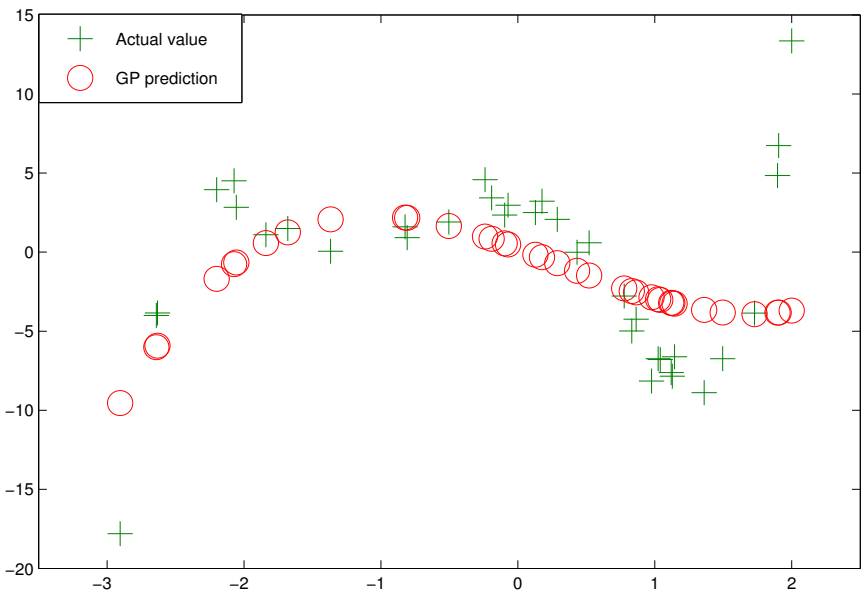

Figure 3.3 GP prediction with a scale parameter $l = 36.817$.

Chapter 4

Speech Enhancement using Relevance Vector Machine

4.1 RVM Model

The RVM assumes a finite linear model which is given by

$$y_i = \sum_{j=1}^{L} w_j \phi_j(\mathbf{x}_i) + \eta_i, \;\; i = 1, \ldots, N \tag{4.1}$$

where $\{\phi_j : \mathbb{R}^d \rightarrow \mathbb{R}\}$ are the basis functions, with L being their total number, w_j is the weight associated with each ϕ_j and η_i is an i.i.d. noise with variance σ^2. As described in [10], an independent Gaussian prior is assumed on each w_j

$$p(w_j|\alpha_j) = \mathcal{N}\big(0, \alpha_j^{-1}\big) \tag{4.2}$$

where α_j means the precision (inverse variance) for w_j. It is interesting to observe that a large value of α_j (close to ∞) implies that the corresponding w_j resets to 0 as the corresponding inverse variance approaches 0 thereby modifying the Gaussian prior to be an impulse-like function at 0 in accordance to the definition in 4.2.

4.2 RVM training

Optimizing the marginal likelihood $p(\mathbf{Y}|\alpha_j, \sigma^2)$ with respect to the $\{\alpha_j\}$ leads to a significant number of the precision values tending towards infinity which renders the corresponding weights to be confined to zero and thus enables us to discard the corresponding basis functions from the model. This scheme is often called the automatic relevance determination (ARD) in which the basis functions that do not contribute to explaining the data are removed and the resulting model becomes sparse.

4.3 Predictions using RVM

As shown in [8], the RVM is a special case of GP where the covariance function is given by $K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{L} \frac{1}{\alpha_j} \phi_j(\mathbf{x}) \phi_j(\mathbf{x}')$. In our work, we specify the basis function by means of a kernel representation as follows:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\left(\mathbf{x} - \mathbf{x}_j\right)^T \left(\mathbf{x} - \mathbf{x}_j\right)}{2l^2}\right) \tag{4.3}$$

and set L to N, the total number of training data.

4.4 Experimental Setup

In this work we implemented the RVM using the code obtained from *http:// www.miketipping.com/sparsebayes.htm*. In the case of RVM, the number of kernel functions involved could be reduced to the number of relevance vectors M which is always a fraction of N. Thus the mean prediction in this case is faster than that of GP with computational complexity given by $\mathcal{O}(M)$. During our experiments, we found that the average number of relevance vectors per RVM model was around 3 $\%$ of the training examples. The experimental results obtained are presented in the following chapter.

Chapter 5

Enhancement using Multi-task Gaussian Process

5.1 Introduction

In the previous chapters the residual gain for each frequency bin was estimated independently using a regressor for each frequency bin. Since the speech signals posses spectral correlations it might be beneficial to jointly estimate the residual gains for a group of frequency bins that are highly correlated. Exploring the issue of jointly estimating residual gains is what we explore in this chapter.

5.2 Multi-Task Learning

Multi-task learning (MTL) is an approach to machine learning that learns a problem together with other related problems at the same time, using a shared representation. This often leads to a better model for the main task, because it allows the learner to use the commonality among the tasks. Therefore, multi-task learning is a kind of inductive transfer. The goal of MTL is to improve the performance of learning algo-

rithms by learning classifiers for multiple tasks jointly. This works particularly well if these tasks have some commonality. The performance improvement achieved by learning the tasks together can be attributed towards the fact that as the tasks share some 'correlation' with each other, the training examples of one task play a role in deciding the decision region of the other task, for the better. Thus effectively, the number of training examples 'seen' by one task is greater than the number of training examples assigned to it as the examples of other tasks are also involved in deciding the decision region of the current task.

In this work, we carry forward the idea of multi-task learning to jointly estimate the residual gain for a group of frequency bins rather than estimating the residual gain for each frequency bin independently. How to group the frequency bins such that the residual gains for the corresponding bins within a group have a higher degree of correlation remains an interesting problem which is discussed in the nest section.

5.3 Frequency Bin Grouping

In order to measure the degree of correlation between the residual gains of the different frequency bins, we collected the residual gain data for the clean speech corrupted by white noise at 5 and 10 dBs. The clean speech data was obtained from all the speakers in the TIMIT database. Thus the total noisy data obtained after corrupting the clean speech with white noise at the above mentioned dBs amounted to 10 hours of data. Figure 5.1 shows the correlation coefficient between the residual gains of the frequency bins varying from 1 to 257 as the DFT size is 512.

From the figure we can see that the adjacent frequency bins are highly correlated while the bins far off are loosely correlated. Thus we use this knowledge to group the frequency bins for multi-task learning. In our frequency grouping scheme, we group

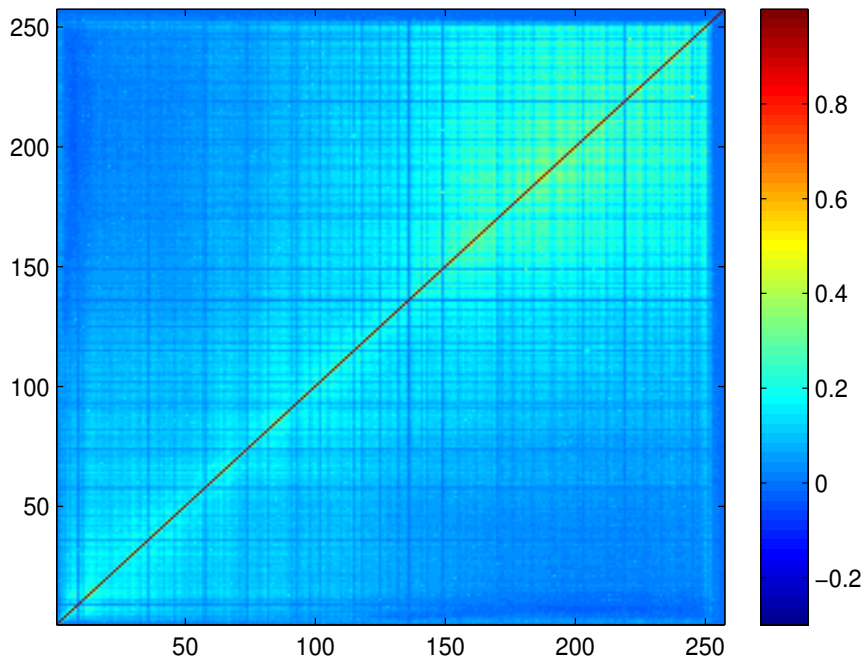

Figure 5.1 Correlation coefficient between the residual gains of the frequency bins varying from 1 to 257. The FFT size is 512.

3 adjacent bins into one single group. This can be expressed as

$$G_i = \{3i - 1,\ 3i,\ 3i + 1\} \tag{5.1}$$

where G_i is the set of frequency bins denoting the i^{th} group. The residual gains of the frequency bins within each group is estimated jointly. It could be noted that the grouping starts from bin number 2 to bun number 357 and the first bin is left out. Thus we have 85 multi-task regressor models (256 frequency bins) and each multi-task regressor estimates the residual gain for the corresponding group.

5.4 Multi-Task GP

For the multi-task regression using GP, the authors in [26] assume the tasks share the same input features.

5.4.1 Model

Given the input set $\boldsymbol{X}$ of N inputs $\boldsymbol{X} = [\boldsymbol{x}_1, \dots \boldsymbol{x}_N]$ the complete set of responses for M tasks is defined as $\boldsymbol{y} = [y_{11}, \dots, y_{N1}, \dots, y_{1M}, \dots y_{NM}]^T$, where y_{il} is the response for the l^{th} task on the i^{th} input $\boldsymbol{x}_i$. The GP prior is defined over the latent function as follows

$$Cov\big(f_l(\mathbf{x}), f_k(\mathbf{x}')\big) = K^f_{lk} k^x(\mathbf{x}, \mathbf{x}') \tag{5.2}$$

$$y_{il} \sim \mathcal{N}(f_l(\boldsymbol{x}_i), \sigma_l^2) \tag{5.3}$$

where K^f is a positive semi-definite (PSD) matrix that specifies the inter-task similarities, k^x is a covariance function over inputs, and σ_l^2 is the noise variance for the l^{th} task. Here k^x is the correlation function same as the covariance function defined in the GP section. The matrix K^f on the other hand is a symmetric positive definite matrix and is parametrized differently.

5.4.2 Inference

Using standard GP formulae, the predictive mean μ^* is found at $\boldsymbol{x}^*$ as

$$\mu^* = (\boldsymbol{k}_l^f \otimes \boldsymbol{k}_*^x)^T (K^f \otimes K^x + D \otimes I)^{-1} \boldsymbol{y} \tag{5.4}$$

where $\otimes$ denotes the Kronecker product, $\boldsymbol{k}_l^f$ selects the lth column of K^f , k_*^x is the vector of covariances between x_* and $\boldsymbol{X}$, K^x is the matrix of covariances between all pairs of training points, D is an $M \times M$ diagonal matrix in which the $(l, l)^{th}$ element is σ_l^2

5.4.3 Experimental Setup

In this work, we implemented the Multi-task GP algorithm using the code provided by the authors, which learns the GP hyper-parameters $\boldsymbol{\theta}$ and computes the posterior mean. The computational complexity for the a posteriori mean prediction is thus $\mathcal{O}(MN)$ provided the inverse is computed already.

Chapter 6

Experimental Results

In order to evaluate the performance of the proposed approaches, we performed experiments on speech enhancement where the clean speech data were drawn from TIMIT database [12]. For training, we used utterances spoken by 50 speakers (25 male and 25 female) and those from other 10 speakers were used for performance evaluation. Waveforms were sampled at 16 kHz and a Hamming window of length 512 samples (32 ms) was applied with a frame shift of 128 samples (75 % overlap). In order to compute the preliminary gain and extract SNR features, we applied the MMSE-LSA algorithm presented in [1]. For the purpose of performance comparison, we also implemented the VQ-based speech enhancement algorithm which is a data-driven technique proposed in [6].

For the first phase of our experiments, we considered the case of 'matched conditions' where the noise types of the training and test data are the same. Three different noise types, taken from Noisex92 database [11] were used in this experiment: white Gaussian, F16 and factory noises. During training, for each of the three noise types the clean speech signals in the training database were artificially degraded by the ad-

ditive noise while varying the SNR in the range (-10 dB, 30 dB). The total length of the training data for each noise type was 5364 seconds.

Performance was measured in terms of four metrics: segmental SNR (SegSNR) [14] improvement, perceptual evaluation of speech quality (PESQ) [15], log-likelihood ratio (LLR) and cepstral distance (CD) [17]. Figures 6.1 and 6.2 plot the four metrics obtained at four different SNR levels: -5, 0, 5 and 10 dBs. In this experiment, we compared the performances of four different approaches: MMSE-LSA, VQ-based (VQ), GP-based (GP) and RVM-based (RVM) speech enhancement algorithms.

From the results shown in Figures 6.1 and 6.2, we can see that the proposed GP and RVM methods produced better metric scores than MMSE-LSA and VQ across all the SNRs with the GP method producing the best results. Especially in high SNR conditions, our proposed methods showed significant improvements over the compared baseline methods.

Figure 6.3 shows example spectrograms of noisy speech and speech enhanced by MMSE-LSA, VQ and GP methods. The enhancement is performed in white noise environment at 10 dB SNR. As seen in the figure, the speech enhanced by GP method has lower residual noise than the speech enhanced by MMSE-LSA and VQ methods.

In the second phase of our experiments, we considered the case of 'mismatched conditions', where the noise types of training and test data are different. During training the models, we used the speech data corrupted only by the white noise. The test data for enhancement were obtained by degrading the clean speech signals with four types of noises different from the white noise: F-16, factory, airport and train noises. The last two types of noises were taken from Aurora -2 database [13].

In this experiment we compared the PESQ scores and Segmental SNR improvement of the input noisy speech with those of the enhanced speech obtained from the MSE-LSA, VQ, GP and RVM approaches. Figure 6.4 shows the average PESQ scores for the four different noise types while Figure 6.5 shows the average Segmental SNR

improvement results for the four different noise types. From the results, we can see that the proposed methods produce better speech quality as compared to the baseline approaches. Overall, it can be concluded that the GP and RVM methods outperform the other enhancement approaches in mis-matched conditions.

In the third and final phase of our experiments, we explored the effect of the form of the residual gain function on the enhancement performance. The residual gain in our work is the difference between the optimal and the statistical gain in the linear-domain. The residual gain in [6] is defined as the difference between the optimal and the statistical gain in the log-domain.

To evaluate the sensitivity of the performance to residual gain being log or linear, we applied GP and VQ methods to predict both the log and linear residual gain. The speech enhanced by estimating these log and linear residual gains using GP had very similar PESQ scores. The same was observed for the speech files enhanced by the VQ method. Thus the residual gain type did not affect the performance substantially. In these comparisons, we used white noise for both the training and testing phases.

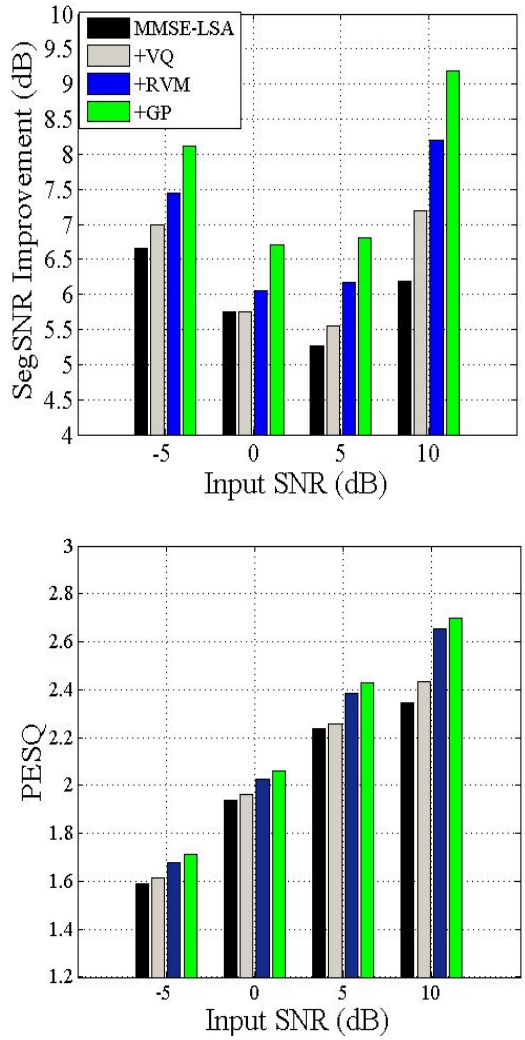

Figure 6.1 Average SegSNR improvement (upper left) and PESQ (upper right) results for MMSE-LSA, VQ, GP and RVM methods in the matched case setting at different SNRs across three noise types.

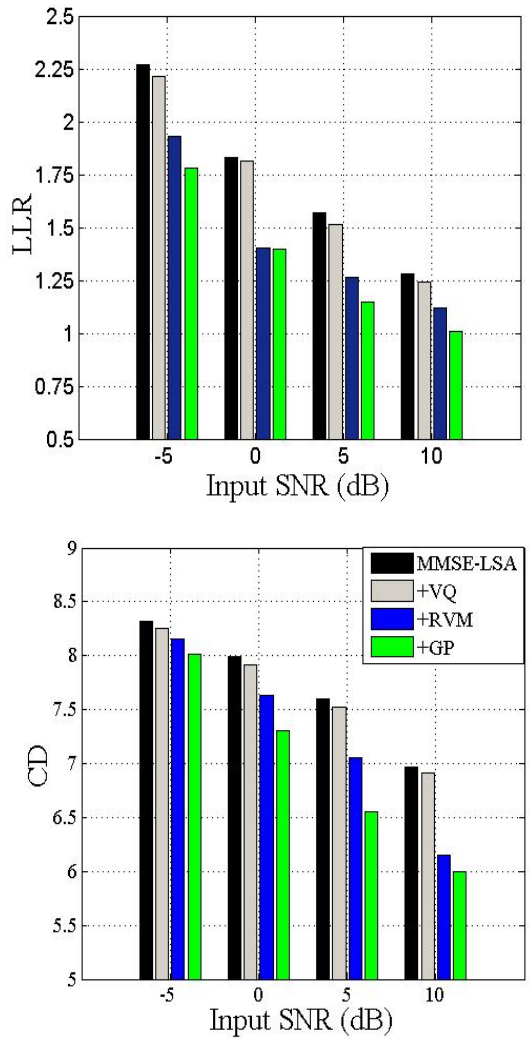

Figure 6.2 Average LLR (bottom left) and CD (bottom right) results for MMSE-LSA, VQ, GP and RVM methods in the matched case setting at different SNRs across three noise types.

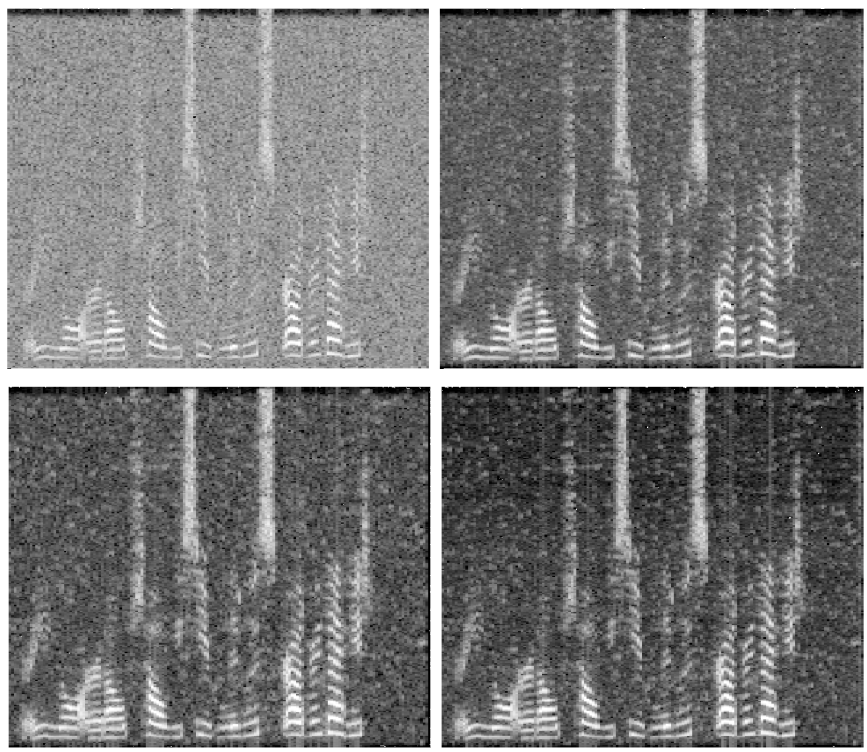

Figure 6.3 Example spectrograms of Noisy speech (upper left) and speech enhanced by MMSE-LSA (upper right), VQ (bottom left), and GP methods (bottom right). The enhancement is performed in white noise environment at 10 dB SNR.

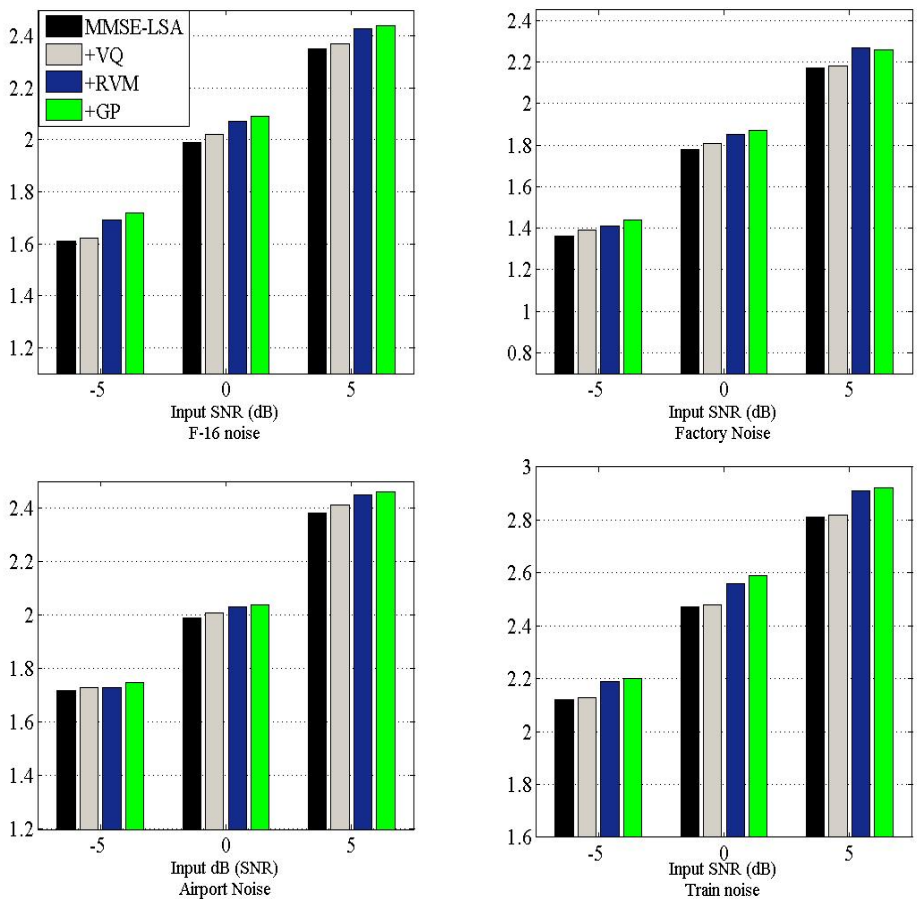

Figure 6.4 PESQ results in the mis-matched case setting at different SNRs for (a) F-16 (b) Factory (c) Airport (d) Train noise types.

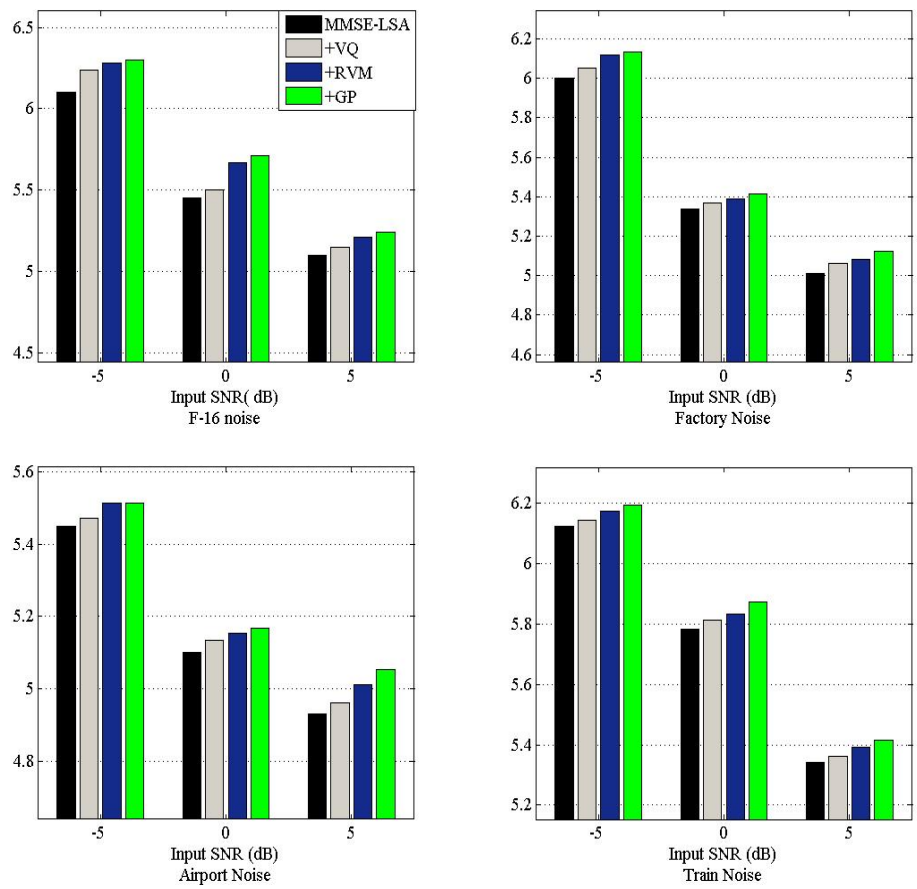

Figure 6.5 Segmental SNR improvement results in the mis-matched case setting at different SNRs for (a) F-16 (b) Factory (c) Airport (d) Train noise types.

Chapter 7

Conclusion and Future Work

In this thesis, we have proposed a novel data-driven approach for speech enhancement by treating the estimation of the residual gain as a regression problem. GP and RVM regression models are employed to estimate the residual gain based on the a priori and a posteriori SNRs as the input. A clustering scheme using VQ clustering is also applied to make the model training tractable. The experimental results have shown that our approach improves the performance of the conventional statistical model-based speech enhancement technique in both the matched and mis-matched noise conditions.

We also extend the our setting to the multi-task case where the residual gain is estimated jointly for a group of frequency bins. As expected, in the multi-task case, the enhancement performance is better than the case where the residual gain is estimated for each frequency bin using GP or RVM. From the experiments in the matched case we can see that our system performs significantly at high SNR conditions. However at low SNR conditions, the performance improvement is comparatively lesser. In the case of mis-matched conditions, however, there still remains room for performance

improvement.

For the case of multi-task GP, we grouped the frequency bins sequentially by observing the degree of correlation among the corresponding residual gains. However in order to get a more better grouping we can also possibly retort to clustering techniques like spectral clustering, constrained clustering etc. This can provide valuable insights regarding the implicit structure of the residual gain distribution with respect to the frequency bins. Using the grouping obtained from the clustering approach, applying multi-task regression to estimate residual gain can also help deliver better speech enhancement performance.

Bibliography

[1] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoustic Speech Signal Processing*, vol. ASSP-33, pp. 443-445, Apr. 1985.

[2] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, no. 11, pp. 2403-2418, Nov. 2001.

[3] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 11, no. 5, pp. 466-475, Sept. 2003.

[4] J. Erkelens, J. Jensen and R. Heusdens, "A data-driven approach to optimizing spectral speech enhancement methods for various error criteria," *Speech Communication*, vol. 49, no. 7-8, pp. 530-541, July-Aug. 2007.

[5] T. Fingscheidt, S. Suhadi and S. Stan, "Environment-optimized speech enhancement," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 16, no. 4, pp. 825-834, May 2008.

[6] Y. G. Jin, N. S. Kim and J. H. Chang, "Speech Enhancement Based on Data-Driven Residual Gain Estimation," *IEICE Trans. Information and Systems*, vol. 94, no. 12, pp. 2537-2540, Dec. 2011.

[7] S. Park and S. Choi, "Gaussian process regression for voice activity detection and speech enhancement," *IEEE Int. Joint Conf. on Neural Networks* pp. 2879-2882, June 2008.

[8] C. E. Rasmussen and C. K. I. Williams,"Gaussian processes for machine learning," the MIT Press, 2006.

[9] R. M. Neal, "Regression and classification using gaussian process priors," *Bayesian statistics*, vol. 6, New York: Oxford University Press.

[10] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, 1, pp. 211-244, June 2001.

[11] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247-251, July 1993.

[12] J. S. Garofolo, "Getting started with the DARPA TIMIT CD-ROM: an acoustic phonetic continuous speech database," National Institute of Standards and Technology, (prototype as of December 1988).

[13] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Int. Speech Communication Association Workshop ITRW ASR*, pp. 181-188, Sept. 2000.

[14] J. Hansen and B. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," *Int. Conf. Spoken Language Process*, vol. 7, pp. 2819-2822, Dec. 1998.

[15] “Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” Tech. Rep. ITU-T P.862, 2001.

[16] C. E. Rasmussen and H. Nickisch, “Gaussian processes for machine learning (gpml) toolbox,” *Journal of Machine Learning Research*, 11, pp. 3011-3015, Dec. 2010.

[17] S. Quackenbush, T. Barnwell and M. Clements, *Objective Measures of Speech Quality*, Englewood Cliffs, NJ: Prentice-Hall, 1988.

[18] Y. Hu and P. C. Loizou, “Evaluation of objective quality measures for speech enhancement,” *IEEE Trans. Acoust. Speech Signal Processing*, vol. 1, no.1, pp. 229-238, Jan. 2008.

[19] D. Gu, “Spatial Gaussian process regression with mobile sensor networks,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 23, pp. 1279-1290, Aug. 2012.

[20] L. Wei, Y. Yang, R. M. Nishikawa, M.N. Wernick, and A. Edwards, “Relevance vector machine for automatic detection of clustered microcalcifications,” *IEEE Trans. Med. Imag.*, vol. 24, no. 10, pp. 1278-1285, Oct. 2005.

[21] J. Sohn, N. S. Kim, and W. Sung, “A statistical model based voice activity detection,” *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1-3, Jan. 1999.

[22] J. W. Shin, J. H. Chang, and N. S. Kim,“Voice activity detection based on statistical models and machine learning approaches,” *Computer Speech and Language*, vol. 24, no. 3, pp. 515-530, Jul. 2010.

[23] Q. H. Jo, J. H. Chang, J. W. Shin, and N. S. Kim, “Statistical model-based voice activity detection using support vector machine,” *IET SIGNAL PROCESSING*, vol. 3, no. 3, May 2009.

[24] J. H. Chang, Q. H. Jo, D. K. Kim, and N. S. Kim, "Global soft decision employing support vector machine for speech enhancement," *IEEE Signal Processing Letters*, vol. 16, vo. 1, pp. 57-60, Jan. 2009.

[25] J. W. Shin, H. J. Kwon, S. H. Jin, and N. S. Kim, "Voice activity detection based on conditional MAP criterion," *IEEE Signal Processing Letters*, vol. 15, pp. 257-260 , Feb. 2008.

[26] E. V. Bonilla, K. M. A. Chai and C. K. I. Williams, "Multi-task Gaussian Process Prediction," *Neural Information Processing Systems*, vol. 20, no. 2007

국문초록

본 논문에서는 가우시안 프로세스 (GP) 와 relevance vector machine (RVM)을 활용한 데이터 구동(data-driven) 방식의 단일 채널 음성 향상을 소개한다. 이 방식에서의 잔여 이득은 스펙트럼 향상에 널리 사용되는 minimum mean square error log spectral amplitude (MMSE-LSA) 추정기로부터 구한 이득과 최적 이득의 차이로 정의한다. GP와 RVM을 적용함으로써 사전 (a priori) 및 사후 (a posteriori) 신호대 잡음비 (SNR) 와 같은 입력 특징들과, 출력 값인 잔여 이득과의 관계를 학습할 수 있다. 이 방식은 크게 두 단계로 나뉜다. 첫 번째 단계에서는 SNR 특징과 MMSE-LSA 추정기의 이득을 계산한다. 두 번째 단계에서는 GP나 RVM을 통하여 잔여 이득을 추정하고, 이는 MMSE-LSA 모듈의 출력을 향상시키는데 사용된다. 실험 결과를 통하여 MMSE-LSA 방식과 다른 데이터 구동 방식의 음성 향상에 비하여 음질이 훨씬 개선된 것을 확인 할 수 있었다. 더 나아가 본 논문에서는 멀티 태스크 가우시안 프로세스 (multi-task GP)를 이용하여 주파수 빈들의 집단에 대하여 연대적으로 잔여 이득을 추정하는 멀티 태스크 (multi-task) 환경으로까지 실험을 확장시켰다. 예상대로 일반 GP나 RVM을 사용하여 각 주파수 빈들의 잔여 이득을 구하는 음성 향상에 비해 멀티 태스킹 환경에서의 음성 향상이 성능이 좋은 것을 확인 할 수 있었다.

주요어: relevance vector machine, 가우시안 프로세스,데이터 구동 , 멀티 태스크 가우시안 프로세스 ,음성 향상을

학번: 2012-23955