Bayesian Tuning of DSP Algorithms Based on User Preference Data

Iman Mossavat

April, 2008

Contents

1 Introduction

A large array of digital signal processing (DSP) systems are dedicated to improving the human perception of signals, e.g., audio, speech, image, and video signals. To name a few, we may think of audio, image and video noise-reduction algorithms. Another example is the image editing algorithms that improve the 'perceived' quality of an image by adjusting contrast, sharpness etc. An important example is compression algorithms that are used in hearing aids (HA) to restore the hearing of hearing-impaired patients. Most of these algorithms possess **tuning parameters**, i.e. parameters that are required to (preferred to) be set according to **subjective human judgement** of the system output. If only a few parameters are available and if we are able to tune each one independently, then tuning would be a relatively easy task and a simple trial-and-error process is usually enough because the right setting is readily identifiable for the user. However, if these parameters need to be tuned simultaneously, the problem soon become tedious for humans as the number of tuning parameter combinations increases. Hence, parameter tuning in applications such as HA, where a few dozen of interactive parameters exist remains a challenge. In my PhD research I am looking forward to developing a principled way to tune DSP algorithms, in particular we are interested in audio and speech processing algorithms such as noise reduction algorithms and compression algorithms used in HA.

While design of DSP systems attracts a great of attention, there is a limited research on principled ways of tuning these systems. We view Tuning as the decision problem of choosing the optimal parameter value. This view

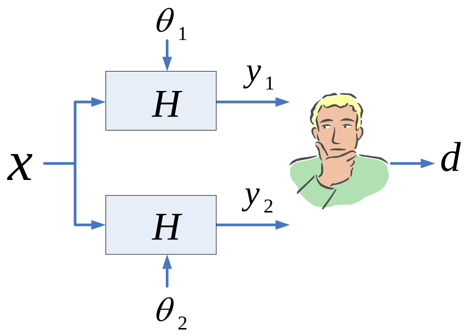

Figure 1: Pairwise comparison tests: Algorithm H processes the input signal x with tuning parameters θ_1 and θ_2. The user indicate its preference between the outputs y_1 and y_2 by the binary signal d.

calls for an optimality or decision criterion. In our approach, we rely on a quantitative representation for user satisfaction rating that we call the user **utility function** to formalize the optimality criterion. We use the **expected expected utility** as the decision criterion [1].

The utility of users are assumed to be unknown, our major research focus is to statistically infer the user utility function from their preferences using Bayesian methods. In our approach, which is illustrated in Figure 1, we do not explicitly ask the user to express his idea about the DSP system output. Instead, the user is presented with a set of alternative value combinations for the tuning parameters, and is supposed to choose the combination that results in the most desirable output. In other words, we learn the preference of users from their choice behavior.

In order to learn the utility function, we have parametric and nonparametric methods at our disposal. The parametric approach uses a few speech features combined in a specific functional form that is governed by a small number of parameters whose values are to be inferred from the data set. An alternative way to construct probabilistic models for the utility function is the nonparametric approach where no explicit parametrization of the utility function is assumed. Instead, we directly define a probability distribution over functions.Speech quality metrics are proposed as an instance of parametric utility model while in this document we propose Gaussian Processes as an instance of a nonparametric utility model. Gaussian Processes are used by Wu and Ghahramani to learn the preferences of users [3].

Tuning problem can pose different challenges based on the DSP system under question; However, the most common yet difficult challenges are as follows:

1. Unreliable, limited or costly data collection from human users.
2. Important individual differences in user preferences.
3. Incorporating prior knowledge such as expert knowledge, human intuition etc in the learning process.

We will discuss the cons and pros of parametric and nonparametric methods considering these challenges. In general, a major disadvantage of parametric approaches is that they rely heavily on the features incorporated in them and they may give a poor model of the data if features do not represent the important perceptual aspects of the signals well. Proper feature selection is a challenging problem which usually requires 'large' data sets.

A main challenge for tuning some DSP systems is that obtaining user data is limited. For example, in the HA domain the data from listening tests are limited in both quantity and quality, considering the fact that patients may become fatigued during a session. Hence, it is of fundamental importance to tune the parameters using as few listening tests as possible. An important point to notice is that we are not interested in modeling the utility function for all the combinations of tuning parameter values. We are primarily interested in the combination that leads to the highest utility value. From this perspective, our problem is very similar to the well-studied problem of optimizing black-box functions that are expensive to sample. Such optimization procedures, as one of our major inspirations, have been developed for Gaussian Process-based learning of the utility functions [2].

Another important challenge in our research is developing probabilistic utility models that are capable of representing user differences while exploiting their similarities. In certain applications like tuning HA parameters this variation among users is of primary interest. Hearing impaired patients need their HA device to compensate for their specific hearing loss which varies greatly from person to person. In parametric utility models, absorbing the user data in a few parameters may potentially limit the capability of the model to reflect variation of personal preferences. On the contrary, Gaussian Processes are known for their flexibility in modeling smooth functions.

Bayesian learning of parametric utility function leads us to the controversial problem of noninformative priors as the parameters of these models do not usually have a physical interpretation. On the contrary, most hyperparameters of the Gaussian Processes have meaningful interpretations that comes in handy when we want to incorporate prior knowledge. For example, there exist valuable expert knowledge for tuning HA parameters. We will illustrate a simple way of incorporating this knowledge using Gaussian

Processes.

In Section 2 we will introduce the concept of expected expected utility as our criterion of optimality. Furthermore, we will present the outline of the utility-based approach to tuning DSP algorithms parameters using preference data. We will present the behavioral model that is going to be the basis of likelihood functions for Bayesian inference of the utility. In Section 3 we will review Bayesian inference, parametric as well as nonparametric approaches to utility modeling. Gaussian Processes (GPs) are going to be discussed in more detail as the heart of our nonparametric utility modeling. In Section 4 we will show the power of nonparametric modeling in taking advantage of expert knowledge in a principled way. In Section 5 we give a detailed account of limitations of quality metrics as utility functions. Finally, in Sections 6 and 7 we will present the broad range of question that we propose to study and the concrete plan for our next step respectively.

2 Preliminaries

In this section we first describe the concept of **expected expected utility** (EEU) which is the decision metric or optimality criterion used for choosing the values of tuning parameters. Afterwards, we present the **behavioral model** that relates the preference data to the latent user utility. This model is used for the Bayesian inference of the utility based on the preference data.

2.1 Expected Expected Utility

let us describe our decision criterion called the **expected expected utility** (EEU). Starting from the basic scenario where for a given user i, and a given input signal x, we wish to set the tuning parameters θ to values $\theta_i^\star(x)$ that maximize the user's utility function. Mathematically put

$$\theta_i^\star(x) = \arg\max_\theta U_i(H(x,\theta)),$$

where we use the personalized utility function $U_i(H(x,\theta))$ to model the satisfaction rating of user i for the signal $y = H(x,\theta)$. Since $U_i(y)$ is not known (or partially known), we maximize the 'expected' utility instead,

$$\theta_i^\star(x) = \arg\max_\theta \int u\, p(U_i(H(x,\theta)) = u)\, du. \tag{1}$$

In some applications such as hearing aids (HAs), we do not allow $\theta^\star$ to be directly dependent on the input signal x. One way to tackle this problem is to use a classifier for input signals x. Members of class c are processed using a single parameter vector $\theta^\star_{ci}$. Vector $\theta^\star_{ci}$ is chosen according to the EEU criterion written as

$$\theta^\star_{ci} = \arg\max_\theta \sum_{x\in\mathcal{X}} p(x|c) \int u\, p(U_i(H(x,\theta)) = u)\, du, \tag{2}$$

where $\mathcal{X}$ is the set of all relevant input signals, and $p(x|c)$ is the probability of signal x being a member of class c. To recapitulate, in the EEU formalism we would like to maximize the 'average' utility with respect to our uncertainty about the utility function as well as with respect to variations in the input signal. These two averaging operations are responsible for the name of this formalism.

How to construct the classifier? In the HA literature, environmental classifiers are designed to identify classes such as speech and music apart. If the DSP algorithm gives its best performance for music and speech using the same tuning parameter vector, then it is not necessary to identify these two classes. Choosing Equation (2) as the optimality criterion, a more appropriate way of designing this classifier is based on the policy which seeks to minimize the variation of $\theta_i^\star(x)$ (Equation (1)) in each class. Hence, the design of this classifier depends on the DSP algorithm H and new classes can be defined according to the concept of minimization of the variance within classes.

2.2 Behavioral Model

Consider a DSP algorithm $y = H(x, \theta)$, where x and y are input and output signals respectively and θ is a vector of tuning parameters. Let us denote the set of all tuning parameter vectors by Θ. In a pairwise comparison test we will process the signal x using $\theta_1 \in \Theta$, and $\theta_2 \in \Theta$ to obtain y_1 and y_2. Next, we ask the user to indicate his preference for one of the two signals y_1 and y_2. We use the notation $y_1 \succ y_2$ to state that y_1 is preferred to y_2. Let us assume that there is a latent utility function value $U(y)$ associated with each signal y, such that the function values preserve the preference relations observed in pairwise comparison tests. In other words, we have

$$p_{\text{ideal}}(y_1 \succ y_2 | U(y_1), U(y_2)) = \begin{cases} 1 & \text{if } U(y_1) \geq U(y_2); \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This assumes that the preference data should exhibit transitivity to be consistent i.e. $y_1 \succ y_2$ and $y_2 \succ y_3$ implies $y_1 \succ y_3$. In experimental data, however, it is common to observe inconsistency. In order to model this inconsistency, we assume that the utility function is contaminated by independent identically distributed noise. Hence the following probabilistic model for the data is proposed

$$\begin{aligned} & p(y_1 \succ y_2 | U(y_1), U(y_2)) = \\ & \iint p_{\text{ideal}}(y_1 \succ y_2 | U(y_1) + n_1, U(y_2) + n_2)\, p_\epsilon(n_1)\, p_\epsilon(n_2)\, dn_1\, dn_2, \end{aligned} \tag{4}$$

where p_ϵ is the noise distribution. We call the map from utilities to the preference data in Equation (4) the 'behavioral model' (or choice model). If we assume the noise to be Gumbel distributed [1] with probability distribution function

$$p_\epsilon(n) = \text{Gum}(n|\mu, \lambda) = \lambda\, \zeta\, e^{-\zeta}, \tag{5}$$

[1]In [3] Wu and Ghahramani assumed zero-mean Gaussian noise with variance σ^2.

where $\zeta = \exp(-\lambda(n-\mu))$, $-\infty < n < \infty$, with mode μ and 'scale' parameter $\lambda > 0$, then Equation (4) evaluates to the 'logit choice model' [8] as,

$$\begin{aligned} p(y_1 \succ y_2 | U(y_1), U(y_2), \lambda) &= \frac{1}{1 + \exp -\lambda(U(y_1) - U(y_2))} \\ &= g(z). \end{aligned} \tag{6}$$

where $g(z) = \frac{1}{1+\exp -z}$ and $z = -\lambda(U(y_1) - U(y_2))$. Equation (6) is also called the Bradley-Terry model. As we will see later the behavioral model is the basis for the likelihood functions used in Bayesian inference of the latent utility function.

3 Bayesian Inference of the Utility Function

In Section 2 we introduced the behavioral model that describes the relation between the observed preference data and the unobserved (latent) utility. Also, we introduced EEU as a measure for choosing the optimal tuning parameter. Now, let us briefly describe Bayes rule that is the heart of our probabilistic inference method. After that we will describe parametric and nonparametric probabilistic utility models.

3.1 The Bayes rule

Suppose that we intend to do Bayesian inference about the variable ω given the observations in the data set $\mathcal{D}$ assuming a model $\mathcal{M}$ that relates the data to the variable. In the Bayesian framework, we calculate the conditional probability $p(\omega|\mathcal{D}, \mathcal{M})$, which we call the **posterior probability** of ω, using

Bayes rule

$$p(\omega|\mathcal{D},\mathcal{M}) = \frac{p(\mathcal{D}|\omega,\mathcal{M})p(\omega|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})}, \tag{7}$$

where we formulate our prior knowledge about the variable of interest ω using the probability distribution called the **prior distribution** or simply the prior $p(\omega|\mathcal{M})$. The second ingredient of Bayesian inference is $p(\mathcal{D}|\omega,\mathcal{M})$, that is the probability of observing data $\mathcal{D}$ given ω assuming model $\mathcal{M}$. The term $p(\mathcal{D}|\omega,\mathcal{M})$ is called the **likelihood function** when it is considered as a function of ω for fixed $\mathcal{D}$. Finally, the term $p(\mathcal{D}|\mathcal{M})$ is called the evidence, the probability of observing the data given the model $\mathcal{M}$. Unless model selection is a concern, we simply treat the evidence as a normalizing term and we write

$$p(\omega|\mathcal{D},\mathcal{M}) \propto p(\mathcal{D}|\omega,\mathcal{M})p(\omega|\mathcal{M}). \tag{8}$$

Note that the prior for inference is the description of what is known about a variable in the *absence* of data $\mathcal{D}$. Priors can be used to incorporate expert knowledge or knowledge from other data sets or previous experiments in our inferences. Choosing a prior that correctly represents our knowledge (or ignorance) is a challenge when using Bayesian methods. One way to assign prior probabilities is to use the principle of 'maximum entropy', where one chooses the prior to be the density that maximizes the Shannon entropy while meeting the set of constraints that defines the set over which the density is defined[2].

[2]This is motivated by the Shannon entropy of the probability distribution which measures the amount of information contained in the distribution. The larger the entropy, the less information is provided by the distribution. Thus, by maximizing the entropy over a suitable set of probability distributions, one finds that distribution that is least informative in the sense that it contains the least amount of information consistent with

For example, the maximum entropy prior on a discrete space, given only that the probability is normalized to 1, is the prior that assigns equal probability to each state. In the continuous case, the maximum entropy prior given that the density is normalized with mean zero and variance σ^2 is the zero-mean normal distribution with variance σ^2 [4]. A prior that assumes no knowledge about the scale of a variable is $p(\omega) \propto 1/\omega$ which is improper, i.e., it does not integrate to one [5]. Choosing the right prior has been a source of philosophical controversy as well as practical problems[3].

3.2 Parametric Vs. Nonparametric Utility Models

Here we are going to discuss two major approaches towards probabilistic utility modeling. Using these models, we can perform Bayesian inference to update our knowledge about the user utility function in the light of preference data. We name these approaches as **parametric** and **nonparametric** according to how MacKay categorizes regression methods in Chapter 45 of his book [6]. Note that the utility function is unobserved as the data $\mathcal{D}$ is comprised of user choices (preferences) not the function values. Following the machine learning literature, we treat the utility as a **latent** function. This is the major difference between our problem and regression, i.e., fitting a function through noisy data.

We will describe how to construct likelihood functions for both **para-**

the constraints that define the set.

[3]Philosophical problems associated with uninformative priors are associated with the choice of an appropriate metric, or measurement scale. Suppose we want a prior for the running speed of a runner who is unknown to us. We could specify, say, a normal distribution as the prior for his speed, but alternatively we could specify a normal prior for the time he takes to complete 100 meters, which is proportional to the reciprocal of the first prior. These are very different priors, but it is not clear which is to be preferred.

metric and **nonparametric** approaches according to the behavioral model proposed in Equation (6) of Section 2.2

In the parametric approach, the utility of signal y corresponding to user i is modeled as

$$U_i(y; \omega_i) = f(v_1(y), \ldots, v_N(y); \omega_i), \tag{9}$$

where the signal y is represented by features $v_1(y)$ to $v_N(y)$ that are combined using the functional form $f(.)$ which is governed by parameters ω_i. We sometimes refer to parametric modeling as **feature-based** modeling depending on the context. In this context, we assume that each person is characterized by the parameter vector ω_i and the uncertainty about the utility $U_i(y)$ is presented by the PDF $p(\omega_i)$. Having chosen the parametrization, we then infer the function $U_i(y)$ by inferring the parameters ω_i. Given the observation $y_1 \succ y_2$, the posterior probability of the parameters is

$$p(\omega_i, \lambda | y_1 \succ y_2) \propto p(y_1 \succ y_2 | \omega_i, \lambda) p(\omega_i) p(\lambda), \tag{10}$$

where λ is the choice model parameter. We should use the maximum entropy criterion under constraints dictated by function $f(.)$ to specify the priors $p(\omega_i)$ and $p(\lambda)$. We use the behavioral model in Equation (6) to specify the likelihood used in Equation (10). In order to find the posterior PDF for the utility parameters we marginalize Equation (10) with respect to the scale parameter λ as

$$p(\omega_i | y_1 \succ y_2) = \int_0^\infty p(\omega_i, \lambda | y_1 \succ y_2) \, \mathrm{d}\lambda. \tag{11}$$

Since λ is the scale parameter of the Gumbel distribution as Equation 5 suggests, we may use the noninformative scale-invariant prior such as $1/\lambda$, which is improper.

If we assume the features to combine linearly with the gain vector ω_i, then we write

$$\begin{aligned} U_i(y) &= \sum_l \omega_{i,l} v_l(y) \\ &= \omega_i^T \mathbf{v}(y), \end{aligned} \tag{12}$$

where $\omega_{i,l}$ is the lth element in vector ω_i and $\mathbf{v}(y)$ is the vector containing features $v_1(y)$ to $v_N(y)$. Using the behavioral model in Equation (6) we can relate ω_i to the preference data

$$p(y_1 \succ y_2 | \omega_i, \lambda) = \frac{1}{1 + e^{-\lambda(\omega_i^T \mathbf{v}(y_1) - \omega_i^T \mathbf{v}(y_2))}}. \tag{13}$$

It is important to note that in parametric methods priors are usually chosen to be noninformative.

The parameterized utility described in previous paragraphs makes no assumption about its input argument y. In order to construct our nonparametric utility model, we will explicitly consider y to be the output of the DSP algorithm $H(x, \theta)$. Let us start by considering the restricted case where we fix the input signal x and we infer $U(H(x, \theta))$ for different θ. Let us denote $U_i^\dagger(\theta; x) = U_i(H(x, \theta))$ and use shorthand $U_i^\ddagger(\theta)$ to refer to $U_i^\dagger(\theta; x)$ when the fixed input signal x is understood from the context. If we omit i from our notation, this either means that the data comes from one user or the user differences are considered as insignificant. We will always make this clear in the context.

By focusing on inferring $U^{\ddagger}(\theta)$ we will be able to incorporate human expert knowledge in our prior, instead of using noninformative priors. In this way, we can use well-studied tuning rules and update them in the light of data using the principled Bayesian approach. This is a significant advantage as noninformative priors may lead to practical difficulties. In addition, using noninformative prior leads to more demand for large data sets to perform inference.

In the nonparametric approach, there is no explicit parametrization of the unknown utility function; Consider an algorithm with N tuning parameters each taking one of k possible values. This leads to k^N combinations of tuning parameters. $U^{\ddagger}(\theta)$ lives in the k^N-dimensional space of all functions of θ. Instead of placing a prior on the utility model parameters, we place a prior $p(U^{\ddagger}(\theta))$ directly on the space of functions.

The key idea in generating a nonparametric model for $U^{\ddagger}(\theta)$ is assuming smoothness. We can define a distance metric, like Euclidean distance, in the multidimensional Θ space. We assume that if θ_1 and θ_2 are two tuning parameter vectors that are close, then $U^{\ddagger}(\theta_1)$ and $U^{\ddagger}(\theta_2)$ are close too. The Gaussian Process (GP) is a strong tool in dealing with smoothing problems. Zero-mean GP were used by Wu and Ghahramani to learn the preferences of users [3]. Also using a GP, an active learning method was developed in [2] to minimize the number of questions asked in order to locate a subject most preferred parameter setting in an image processing task. Based on these successful results in preference modeling and active learning, we propose to use GPs to learn the utility function $U^{\ddagger}(\theta)$.

3.3 Gaussian Processes as Priors over Functions

Let us now use the terminology defined in Section 3.1 to introduce Gaussian Processes. We use Gaussian Processes to define distributions over functions. We will follow Rasmussen [7] to define the concepts related to Gaussian Processes.

Definition 1 *A Gaussian Process is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions.*

In order to specify a Gaussian Process we only need to specify the mean function $m(x)$ and covariance function $k(x, x')$. We will write

$$f \sim \mathcal{GP}(m, k), \tag{14}$$

meaning: "the function f is distributed as a Gaussian Process with mean function m and covariance function k". In typical machine learning applications we are not able to fully specify mean and covariance function a priori. As Rasmussen [7] states:

> In the light of typically vague prior information, we use a hierarchical prior, where the mean and covariance functions are parameterized in terms of hyperparameters. For example, we could

use a generalization of Equation 14

$$
\begin{aligned}
f &\sim \mathcal{GP}(m, k), \\
m(x) &= ax^2 + bx + c, \\
k(x, x') &= \sigma_y^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) + \sigma_n \delta[x - x'],
\end{aligned} \tag{15}
$$

where $\delta[x - x']$ is nonzero only at the origin where it takes on the value 'one'. We have introduced **hyperparameters** $\alpha = \{a, b, c, \sigma_y, \sigma_n, l\}$. The purpose of this hierarchical specification is that it allows us to specify vague prior information in a simple way. For example, we have stated that we believe the function to be close to a second order polynomial, but we have not said exactly what the polynomial is, or exactly what is meant by 'close'. In fact the discrepancy between the polynomial and the data is a smooth function plus independent Gaussian noise, but again we do not need exactly to specify the characteristic length scale l or the magnitudes of the two contributions. We want to be able to make inferences about all of the hyperparameters in the light of the data.

4 Incorporating Expert Knowledge in Utility Training

In this section we will describe how using Gaussian Processes provides a principled way to incorporate expert knowledge in optimization of tuning parameters. We consider two examples to illustrate our idea.

The first example is tuning a hearing aid device. Knowing the audiogram of the patient, audiologists set the tuning parameters to values prescribed by 'fitting rules'. While in the parametric utility modeling there is no clear way for incorporate this knowledge in the utility function, in a Gaussian Process-based utility modeling framework, it is relatively straightforward to do so as we will discuss in the following. Assume that we would like to learn the utility function $U_i^{\ddagger}(\theta)$ using a Gaussian Process with mean function $m(x)$ and covariance function $k(x, x')$ for patient I with audiogram a_i. Further, assume that the fitting rules prescribe θ_i to be the appropriate setting for the tuning parameters. Given a distance metric $d(\theta_i, \theta)$ in the Θ space, we propose the following mean function for the GP

$$m(\theta) = \exp(-\kappa \, d(\theta_i, \theta)), \tag{16}$$

where κ is a hyperparameter that we can learn from data. Equation (16) 'guides' the learning process towards values as prescribed by audiology experts while the flexibility of Gaussian Processes allows to improve the tuning quality according to listening tests.

The second example is tuning a noise reduction algorithm called perceptual noise reduction (PNR) which is a spectral-subtraction-based noise reduction algorithm, i.e. an estimated noise spectrum is subtracted from the signal spectrum. Because the noise estimation is not perfect, processing artifacts are present in the algorithm output. In order to improve the sound quality, we multiply the estimated spectrum by a gain $(1 - \mathrm{g}_{\min})$ before subtraction is performed. The parameter $\mathrm{g}_{\min}$ has a significant perceptual impact. A small $\mathrm{g}_{\min}$ renders the processing artifacts imperceptible while only a poor noise reduction is achieved; on the other hand, a large value of $\mathrm{g}_{\min}$ reduces noise significantly while it results in perceptible processing artifacts in the output. The optimal value for $\mathrm{g}_{\min}$ depends on signal-to-noise ratio (SNR) and noise statistics such as its spectrum and distribution. We assume that the individual differences among users are insignificant. We would like to learn $U^{\dagger}(\theta; x)$ where $x = s + \sigma_n n$ where s is the clean signal and σ_n is the noise standard deviation and n is the noise signal. For a given noise spectrum and distribution we believe that $\mathrm{g}^{\star}_{\min}$ increases as SNR increases i.e. σ_n increases. To incorporate this knowledge, we can model the utility $U^{\dagger}(\mathrm{g}_{\min}; x)$ by a Gaussian Process with mean function

$$m(\mathrm{g}_{\min}) = \exp(-\kappa(\mathrm{g}_{\min} - g(\sigma_n))^2) \tag{17}$$

where κ is a hyperparameter and $g(\sigma_n)$ is an increasing function of σ_n with a possible floor at low SNRs. Using Equation (17) we give a hint that the utility function is unimodal and as SNR decreases, the optimal value for $\mathrm{g}_{\min}$ shifts towards higher values until a saturation occurs.

5 Objective Speech Quality Metrics as Utility Models

Since we are primarily interested in speech applications, the initial idea to construct utility models was to choose a speech quality metric. These metrics are meant to provide an objective estimate of the subjective speech quality by mapping different amounts of various distortions on the 'perceived' speech quality. These metrics essentially compute a few speech features that are meant to reflect the effects of various speech distortions and background noise. In a large class of speech quality metrics, the overall estimate of the speech quality is computed as in Equation (12). The data from large data sets of listening tests are absorbed into a few model parameters. The original motivation of developing these metrics is evaluation of quality in telephony and Voice over IP applications where normal-hearing individuals are the primary users. Since there is not much meaningful variation among normal hearing individuals, these models for subjective quality assessment of users are based on the assumption that the individual differences are insignificant. To train the parameters of these models data from multiple subjects are gathered into one large database in order to learn the average ratings of multiple individuals. From this viewpoint, the variation in the individual subjective evaluation is pure noise. A particular quality metric called the coherence-based speech intelligibility index (CSII) was proposed to take the role of utility.

Contrary to speech quality metrics, individualization of user utility function is of fundamental importance when we deal with tuning of the hearing aid (HA) tuning parameters. A hearing-impaired patient need his HA to compensate for his specific hearing loss which may vary greatly from other patients. The primary idea for constructing individualized utility functions based on speech quality metrics was to use to assign personalized gains for each patient in order to characterize him. In this proposal the community data was used to generate a prior distribution for learning this gains. Using CSII as the utility metric implicitly 'assumes' that it is possible to characterize each individual by the few model parameters that the CSII offers us with sufficient accuracy needed for individualized tuning of the parameters. However, following the discussion above, there is doubt whether this is a reasonable assumption.

It is possible to extend CSII for hearing-impaired patients by modifying the speech features according to the audiogram of the patient. However, hearing-impaired patients with similar audiograms usually vary greatly on how they want their HA device to be tuned. Thus an audiogram does not represent a patient sufficiently for tuning parameters. One may think of achieving individualization by a combination of audiogram-based feature adaptation and individualized parameter learning. The shortcoming of this approach is how to form priors from community data for learning the individual parameters (since features also change from person to person). It is important to note that, if we plan to use a feature-based model like CSII, we also need a model for individualization. This model has to map the community data into some prior knowledge for training the parameters of each individual model.

Last but not least, we should note that tuning a hearing aid is not only a matter of speech quality. For example, one of the HA main objectives is to restore the speech intelligibility. The quality metric may fail to properly tell us how a certain setting of HA tuning parameter is favored over another setting of tuning parameters. Hence, one may think of several sub-utilities combined together. This necessitates models for combination of these submodes and needs extensive amount of data to train and evaluate these models. In my idea, the root of these problems is our limited knowledge about human perception of sound and lack of knowledge about proper feature selection. Both these problems are out of the scope of our research.

In summary, I believe that using CSII as the utility function for tuning HA is questionable. In order to use a parametric model, we need to choose speech features that reflect the relevant information contained in the preference data. Furthermore, we should know if we can personalize the model based on individual differences. Is a linear model capable of representing different individuals properly? The traditional models for speech quality come with an array of assumptions; some of them hardly fit in the context of tuning HA devices.

6 Broad Research Perspective

Parametric methods for generating utilities lead us towards the difficult problem of feature selection and potentially result in limited capability for individualization. Nonparametric approaches such as Gaussian Processes, on the contrary, allow us to deal with modeling the utility in a straight manner where no features are needed. Furthermore, their flexibility provides enough room for learning the individual characteristics of the utility function.

Gaussian Processes were successfully used by Wu and Ghahramani [3] in simple scenarios of preference modeling. Using Gaussian Processes to learn $U^{\ddagger}(\theta)$ offers a natural way to incorporate expert knowledge. As the first step to extend the work of Wu and Ghahramani [3] is to construct priors according to expert knowledge. Developing active learning similar to [2] in such an extended model is an interesting problem.

While the approach of Wu and Ghahramani [3] is sufficient to solve the problem of learning $U^{\ddagger}(\theta)$, our problem i.e. learning $U_i^{\dagger}(\theta; x)$ is more complex since the utility function may vary for different users and different input signals. To do so, one possible approach is to develop hierarchical models that relate the utility corresponding to signal x_1 i.e. $U_i^{\dagger}(\theta; x_1)$ to utility corresponding to x_2 i.e. $U_i^{\dagger}(\theta; x_2)$. Another issue that we need to address is learning from the community data. If there are several users, the utility of each is learned independently using techniques of Wu and Ghahramani [3]. Hence, their method achieves individualization naturally. The natural question here is that whether we can benefit more from the community data, i.e. data from the pool of users. Putting it in a different way, we are interested to know how to relate the utility corresponding to the ith individual i.e. $U_i^{\dagger}(\theta; x)$ to the utility corresponding to the jth individual i.e. $U_j^{\dagger}(\theta; x)$. Again, developing active learning similar to [2] in such models is an interesting problem.

7 A Concrete Proposal for Using GP as Utility

Since there is less variability in the normal hearing subjects and they are readily available to perform listening tests, we consider the problem of tuning the PNR noise reduction algorithm for normal hearing subjects in order to get started. To start, we propose tuning PNR using a Gaussian Process is a reasonable step, our research objectives are:

1. Incorporate human knowledge by using a mean function similar to Equation (17). This allows to model how optimal PNR tuning parameters vary as SNR varies.

2. Instead of using the simple covariance function as that used by Wu and Ghahramani [3], use a more complex function with hyperparameters controlling the correlation in different directions. Sensitivity of PNR with respect to different tuning parameters vary. Using the community data of multiple users, we can learn the hyper-parameters of the distance metric and quantify the relative perceptual significance of parameters. This may lead to a principled way of studying an important concept in auditory research called just noticeable difference.

3. Develop active learning method for finding the optimal tuning parameter for the entire SNR range in as few listening tests as possible.

Furthermore, we are going to construct artificial data sets to study different computational issues and test our methods and codes accordingly. To generate these data sets we construct functions according to a set of constraints inspired by real world utilities. We use the generative model proposed in Equation 6 to generate artificial reference data. We will infer the utility function using the artificial data.

References

[1] C. Boutilier. On the foundations of expected expected utility. *Proc. of the 4th Int. Joint Conf. on on Artificial Intelligence*, pages 285–290.

[2] E. Brochu, N. de Freitas, and A. Ghosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems*, 2007.

[3] W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. *Proc. ICML*, 2005.

[4] E. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227–241, 1968.

[5] R. Kass and L. Wasserman. The Selection of Prior Distributions by Formal Rules. *Journal of the American Statistical Association*, 91(435), 1996.

[6] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003. available from http://www.inference.phy.cam.ac.uk/mackay/itila/.

[7] C. Rasmussen. Gaussian Processes in Machine Learning. *Advanced Lectures On Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003: Revised Lectures*, 2004.

[8] K. Train. *Discrete Choice Methods with Simulation.* Cambridge University Press, 2003.