TR－A－0156

DISCRIMINATIVE FEATURE EXTRACTION

Shigeru Katagiri, Biing-Hwang Juang, Alain Biem

1992.11.27

ATR視聴覚機構研究所

〒619-02 京都府相楽郡精華町光台2-2 ☎07749-5-1411

ATR Auditory and Visual Perception Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1411
Facsimile: +81-7749-5-1408

DISCRIMINATIVE FEATURE EXTRACTION

*Shigeru Katagiri *, Biing-Hwang Juang**, and Alain Biem**
*ATR Auditory and Visual Perception Research Laboratories,
**AT&T Bell Laboratories

October 29th, 1992

Abstract

Pattern recognition consists of two main stages: feature extraction and classification. Needless to say, these two constituent processes should be designed systematically in a manner consistent with accurate recognition. However, such consistency has not yet been achieved in pattern recognition methods up to now. We thus propose in this paper a novel solution to this important long standing problem. The proposed method is mainly based on a recent discriminative learning theory, the Minimum Classification Error formalization and the Generalized Probabilistic Descent method, and referred to as Discriminative Feature Extraction. A key idea of Discriminative Feature Extraction is to embed both procedures of feature extraction and classification in a smooth functional form and consistently design both stages so as to reduce the number of misclassifications. An application of the method to speech recognition clearly shows the great promise of this new approach.

1 Introduction

For clarity of discussion, we assume that pattern recognition consists of two stages, feature extraction and classification. Feature extraction is usually executed based on knowledge specific to a given task or criterion that is not directly linked to the final classification goal. On the other hand, classification is generally performed by using statistical pattern classification of the resulting features. In this paper, we present a method which integrates the two stages so as to systematically perform the entire recognition process in a manner consistent with accurate classification.

The recent advent of Minimum Classification Error formalization (MCE)/ Generalized Probabilistic Descent method (GPD) provided a new theoretical ground for discriminative pattern classification that unifies both the feature extraction and classification stages [1-4]. The usefulness of MCE/GPD has been demonstrated in many experiments. However, the full potential of this new framework has not yet been revealed. The most important philosophy of MCE/GPD was to formalize the overall procedure in a given task in a *smooth* (at least first differentiable) functional form suited to the use of a practical gradient-based search algorithm. This concept is worth applying to many procedures besides classification. Our focus in this paper is to overcome the above-mentioned gap between feature extraction and classification by embedding a feature extraction process in an MCE/GPD-based classifier design. We call this discrimination-oriented feature extraction Discriminative Feature Extraction (DFE). This paper is intended to introduce this novel approach to pattern recognition.

The paper is organized as follows. In Section 2, MCE/GPD is described. In Section 3, we specifically focus on the idea of embedding the entire classification process in a smooth functional form. Section 4 is devoted to introducing DFE and examples of its application to speech recognition. The paper is summarized in Section 5.

2 MCE/GPD-Based Discriminative Learning

2.1 Discriminant function approach: background of MCE/GPD

Here, we consider an M-class classification task, $\{C_j\}_{j=1}^{M}$. A classifier consists of a set of trainable parameters Λ. Given a set of design pairs, a pattern sample $\mathbf{x}_n$ and its class C_k ($n = 1, ..., N$ and $k = 1, ..., M$), we aim at designing an optimal Λ. We also assume that an individual sample is already represented as a K-dimensional feature vector.

Bayes decision theory provides a fundamental guideline to design. This approach is based on the Bayes decision rule

$$C(\mathbf{x}) = C_i \quad \text{if} \quad P_\Lambda(C_i|\mathbf{x}) = \max_j P_\Lambda(C_j|\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is an arbitrary sample, $\mathbf{x} \in \mathcal{R}^K$, $C(\cdot)$ denotes a classification operation, and it is assumed that the true *a posteriori* probability has the parameterized form $P_\Lambda(C_j|\mathbf{x})$ and the precise values of Λ is known. An actual design procedure is to estimate the *a posteriori* probabilities or conditional probabilities. The above decision rule, if it can be used, represents the best classification situation, i.e., the Bayes minimum risk. Perfect execution of Eq. (1) guarantees realization of the Bayes minimum risk, and this rule can thus be considered a principle for statistical classifier design. However, the truth is that this approach suffers from the serious difficulty that the nature of the sample distributions, such as the form of the density function, is rarely known and it is then almost impossible to estimate desired probabilities in Eq. (1).

An alternative to the Bayes decision approach is represented in the following functional form classifier:

$$C : \mathcal{R}^K \rightarrow \{C_j\}_{j=1}^{M} \tag{2}$$

In this most general case, the decision rule is embedded in a functional form. This formalization is less practical, however. Thus, the classifier is usually reduced to a more practical version which is associated with the following decision rule:

$$C(\mathbf{x}) = C_i \quad \text{if} \quad g_i(\mathbf{x}; \Lambda) = \max_j g_j(\mathbf{x}; \Lambda) \tag{3}$$

where $g_j(\mathbf{x}; \Lambda)$ is referred to as a discriminant function and the classifier function $C(\cdot)$ is expressed in an operational form. The classification based on Eq. (3) and discriminant function designs is referred to as the discriminant function approach. This approach does not require assumptions about the form of the sample distributions. This allows one to execute the classification in a manner more flexible than the Bayesian approach. Any reasonable measure such as a distance can be used as the discriminant function. The computation of these measures is usually simple. Thus, this approach is quite practical. Recall that there are many well-studied examples. A classical linear discriminant function has long been used. The resurgence of discriminative learning by modern artificial neural networks is still fresh in memory. However even this attractive approach is not perfect. In particular, there was a big gap between actual design of discriminant functions and realization of the Bayes minimum risk. One solution to this serious difficulty is MCE/GPD.

In this section we review the background of MCE/GPD. Design of discriminant functions is usually characterized by two factors: 1) learning objective and 2) optimization of Λ (minimum search of the objective).

The ultimate way is perhaps to find the minimum of an error count objective by using Simulated Annealing. This objective can be represented as the average of discontinuous 2-state functions, each enumerating error one for misclassification, and zero for correct classification. Given an infinite run of adjustments, Simulated Annealing can find with probability one the minimum state of the objective, which corresponds to an optimal set of Λ. This property makes the method attractive, but the infinite training it requires is never realistic, and even practical implementations of the annealing process converge extremely slowly. Moreover, a simple execution of this method in a real situation where only a finite number of design samples are available means that even with an infinite training run, Simulated Annealing is still prone to the training robustness problem. Therefore, a more practical design method is required.

Thus, a main concern in developing MCE/GPD was to create a method satisfying the following conditions, 1) directly attaining the Bayes minimum risk, 2) learning efficiently, and 3) being highly practical. The key to the MCE/GPD solution was to embed the entire process of classification in a smooth functional form and design an at least locally optimal state of classifier parameters through gradient descent-based adaptive training. MCE and GPD are closely related to each other, and it is thus rather difficult to draw a boundary between both. In this paper, we specifically introduce them in the following categorization: MCE is a theoretical framework for discriminative learning aiming at minimum classification error [3, 4]; GPD is a practical, adaptive learning procedure suitable for discriminating various kinds of patterns in the sense of MCE [1, 2].

2.2 Minimum classification error formalization

Let us consider the situation that $\mathbf{x}_n$ is selected from the given design samples. We assume $\mathbf{x}_n \in C_k$. MCE formalization consists of 3 steps. The first step defines a discriminant function $g_j(\mathbf{x}_n;\Lambda)$ which represents the degree to which $\mathbf{x}_n$ belongs to C_j. As cited before, any reasonable measure can be used to define the function. Specifically, we assume that $g_j(\mathbf{x}_n;\Lambda)$ is a distance measure. The second step is the heart of MCE. A smooth misclassification measure is introduced here to simulate the operation in Eq. (3), i.e., comparison/decision among the competing classes. Among many possibilities,

$$d_k(\mathbf{x}_n;\Lambda) = g_k(\mathbf{x}_n;\Lambda) - \left[\frac{1}{M-1}\sum_{j,j\neq k}\left\{g_j(\mathbf{x}_n;\Lambda)\right\}^{-\mu}\right]^{-1/\mu} \tag{4}$$

is a typical definition, where the classification is expressed by decision on a scalar value, and μ is a positive number. $d_k(\mathbf{x}_n;\Lambda) > 0$ implies misclassification, and $d_k(\mathbf{x}_n;\Lambda) \leq 0$ means correct classification. Note here that varying the value of μ allows one to realize various decisions. The third step completes MCE by embedding the misclassification measure in a loss

$$\ell_k(\mathbf{x}_n;\Lambda) = \ell_k\big(d_k(\mathbf{x}_n;\Lambda)\big), \tag{5}$$

where $\ell_k()$ is a monotonically-increasing smooth function. The loss is introduced to evaluate a classification result.

We focus on a smooth classification error count loss

$$\ell_k(\mathbf{x}_n;\Lambda) = \frac{1}{1+e^{-\alpha\left(d_k(\mathbf{x}_n;\Lambda)+\beta\right)}}, \quad \alpha > 0, \tag{6}$$

where α and β are real numbers. It is now clearer that MCE can directly attain the minimum classification error situation. As a first step, let us assume that the discriminant function is selected so as to have the correct form of the *a posteriori* probability $P_\Lambda(C_i|\mathbf{x})$. The Bayes minimum risk is then expressed as

$$\mathcal{E} = \sum_{k=1}^{M} \int_{\mathcal{X}_k} P_\Lambda(\mathbf{x}, C_k) 1(\mathbf{x} \in C_k) d\mathbf{x}, \tag{7}$$

where $1(\mathcal{A}) = \begin{cases} 1, & \text{if } \mathcal{A} \text{ is true} \\ 0, & \text{otherwise} \end{cases},$

$\mathcal{X}_k = \left\{ \mathbf{x} \in \mathcal{X} \;\middle|\; P_\Lambda(C_k|\mathbf{x}) \neq \max_j P_\Lambda(C_j|\mathbf{x}) \right\}$, and

$\mathcal{X}$ is the entire observation space.

This can be approximated by the misclassification measure and the loss as follows:

$$\begin{aligned} \mathcal{E} &= \sum_{k=1}^{M} \int_{\mathcal{X}_k} P_\Lambda(\mathbf{x}, C_k) 1(\mathbf{x} \in C_k) 1\Big(P_\Lambda(C_k|\mathbf{x}) \neq \max_i P_\Lambda(C_i|\mathbf{x})\Big) d\mathbf{x} \\ &\simeq \sum_{k=1}^{M} \int_{\mathcal{X}_k} P_\Lambda(\mathbf{x}, C_k) 1(\mathbf{x} \in C_k) \ell_k\left(d_k(\mathbf{x}; \Lambda)\right) d\mathbf{x}. \end{aligned} \tag{8}$$

An important point here is the fact that the approximation accuracy of Eq. (8) can be arbitrarily increased by varying the smoothing constants in the MCE functions such as the loss. That is to say, MCE possesses the capability to approximate the Bayes minimum risk with arbitrarily high accuracy, in the extreme case of this discriminant function approach. This result proves that MCE potentially bridges the gap between the discriminant function approach and the Bayes minimum risk.

The smoothness of MCE has turned out to be extremely useful in various stages of analysis. In fact, the previous discussion already shows that the use of L_p-norm form greatly increases the generality of the classification rule formalism. In addition, the effect of smoothness on training robustness should be addressed. To describe this point, we consider an empirical classification error rate

$$L(\Lambda) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{M} \ell_k(\mathbf{x}_n; \Lambda) 1(\mathbf{x}_n \in C_k). \tag{9}$$

Since sample distributions are unknown, this sample average-form, empirical error is only one measurable objective in a real situation. If the loss is a real error count, i.e., a piece-wise linear 2-step function, this error rate has the shape of a piece-wise linear, multi-step surface. As the number of samples increases, the surface becomes smoother and goes to a continuous, curved surface. On the other hand, the use of a smooth loss makes the surface of this error rate smoother, even if the number of samples is not increased. This effect is equivalent to perturbing and increasing the effective size of design samples. If this perturbation is properly done around the original locations of given samples, the resultant situation can increase the robustness. Interestingly, it was demonstrated that the smoothness did not drastically change the shape of the empirical error based on piece-wise 2-value losses; i.e., it seems that the perturbation was locally effective. Therefore, it is probably true that this smoothness has a certain contribution to classifier robustness. This effect is evidently worth further investigating, and especially the relation between the loss smoothness and the samples finiteness (scatter property) should be an interesting topic.

2.3 Generalized probabilistic descent method

As suggested by the name, GPD is a modern, extended version of the classical probabilistic descent method [5]. GPD gives a rigorous form, suited for gradient search-based design, for classifying ***dynamic*** (variable-durational) patterns by various kinds of system structures.

We have assumed that the pattern sample is a fixed dimensional vector. However, many kinds of natural patterns such as speech signals are actually dynamic. For instance, it is obvious that segments of the same phoneme class can have different durations. A proper classification of these dynamic patterns requires a significant extension of the traditional classification methods. In fact, even the modern artificial neural networks can hardly overcome this difficulty, and as a result many hybrid structures incorporating hidden Markov models (HMMs) or Dynamic Time Warping (DTW) based on Dynamic Programming (DP) have been reported [6-12]. Let us assume in the remaining part of this section that all samples $\mathbf{x}_n$'s are dynamic. Moreover, let us assume that a DTW distance classifier assigning a dynamic reference pattern, denoted by $\mathbf{r}$, to each competing class is prepared to classify these dynamic patterns; $\mathbf{r}_k \subset \Lambda$ and $\mathbf{r}_k \in C_k$. These reference patterns are designed through the pursuit of the minimum classification error situation. Measuring different durational patterns requires the normalization of duration. As widely seen in speech recognition, DTW uses a discriminant function

$$g_k(\mathbf{x}_n;\Lambda) = \min_{\theta}\Big\{D_\theta(\mathbf{x}_n,\mathbf{r}_k)\Big\}, \tag{10}$$

where $D_\theta(\mathbf{x},\mathbf{r}_k)$ is a *path distance* accumulated along the θ-th best (smallest distance) path selected by the DP-matching between $\mathbf{x}_n$ and $\mathbf{r}_k$ among all the possible Θ paths. The operation searching the best normalization path associated with the minimum accumulated distance is obviously discontinuous in Λ. This is an impediment in the gradient descent method. A GPD solution to this problem is to replace the best path search operation (minimum distance search operation) by a smooth search function based on L_p-norm form

$$g_k(\mathbf{x}_n;\Lambda) = \left[\sum_{\theta=1}^{\Theta}\Big\{D_\theta(\mathbf{x}_n,\mathbf{r}_k)\Big\}^{-\xi}\right]^{-1/\xi}, \tag{11}$$

where ξ is a positive constant. Notice that Eq. (11) closely approximates Eq. (10), when ξ goes to infinity.

Similarly, the idea of smooth search operation is utilized to define a discriminant function

$$g_k(\mathbf{x}_n;\Lambda) = \left[\sum_{b=1}^{B_k}\Big\{D(\mathbf{x}_n,\mathbf{r}_k^b)\Big\}^{-\zeta}\right]^{-1/\zeta}, \tag{12}$$

where

$$D(\mathbf{x}_n,\mathbf{r}_k^b) = \left[\sum_{\theta=1}^{\Theta}\Big\{D_\theta(\mathbf{x}_n,\mathbf{r}_k^b)\Big\}^{-\xi}\right]^{-1/\xi},$$

$D(\mathbf{x}_n,\mathbf{r}_k^b)$ is a *reference distance* between $\mathbf{x}_n$ and the b-th best C_k reference $\mathbf{r}_k^b$, and B_k is the number of C_k references, for a classifier in which multiple reference patterns are assigned to each class. When ζ goes to infinity, Eq. (12) approximates a discriminant function which represents the corresponding class by the smallest reference distance of the class. Note that the idea of a smooth operation, underlying Eqs. (11) and (12), is conceptually the same with that in Eq. (4), i.e., the smooth comparison among the competing classes.

The entire process of classifying multi-class dynamic patterns is now formalized in a smooth functional form suited for gradient search. Consequently, one can design a distance classifier having at least a locally-minimum classification error situation, by using a smooth loss, e.g. Eq. (6). There are several versions of gradient search algorithms. The selection here is flexible. One major motivation for GPD is to be able to accomplish adaptive learning. It is highly desirable that a classifier always learns to refine itself given a new sample. It is probably even more desirable to be able to adaptively accomplish minimization of the expected classification error. The following probabilistic descent theorem provided a rigorous mathematical ground which satisfies these requirements [5].

[Probabilsitc Descent Theorem]

Given $\mathbf{x} \in C_k$, *if the classifier parameter adjustment* $\delta\Lambda(\mathbf{x}, C_k, \Lambda)$ *is specified as*

$$\delta\Lambda(\mathbf{x}, C_k, \Lambda) = -\varepsilon \mathbf{U} \nabla \ell_k(\mathbf{x}; \Lambda) \tag{13}$$

where $\mathbf{U}$ *is a positive-definite matrix and* ε *is a small positive real number, then*

$$E[\delta L(\Lambda)] \leq 0, \tag{14}$$

where

$$L(\Lambda) = \sum_{k=1}^{M} \int P_\Lambda(\mathbf{x}, C_k) 1(\mathbf{x} \in C_k) \ell_k(\mathbf{x}; \Lambda) d\mathbf{x}. \tag{15}$$

Furthermore, if an infinite sequence of random observations $\mathbf{x}_t$ *is presented for training and the parameter adjustment rule of (13) is utilized with a corresponding step size sequence* ε_t *which satisfies*

$$\text{i)} \quad \sum_{t=1}^{\infty} \varepsilon_t \to \infty \quad ; \text{and} \tag{16}$$

$$\text{ii)} \quad \sum_{t=1}^{\infty} {\varepsilon_t}^2 < \infty, \tag{17}$$

then the parameter sequence Λ_t *according to*

$$\Lambda_{t+1} = \Lambda_t + \delta\Lambda(\mathbf{x}_t, C_k, \Lambda_t) \tag{18}$$

converges with probability one to a Λ^* *which results in a local minimum of* $L(\Lambda)$.

The above smooth formalization and the probabilistic descent theorem thus complete the adaptive discriminative training for classifying dynamic patterns by the distance classifiers.

We have used a distance measure as our discriminant function. However, a probability measure is most likely a more useful discriminant function. To this end, [24] and [25] provide a detailed description of the method to design an HMM classifier, which is considered most useful for classifying dynamic patterns at present, in the MCE framework.

Notice that a fixed-dimensional vector is merely a special case of a dynamic pattern, and that no specific assumptions of the patterns were made in the previous discussion. It is thus evident that MCE/GPD can be applied to an extremely wide range of pattern classification.

2.4 Relations with other classification/training methods

We briefly refer to the relations between MCE/GPD and other training methods. The readers may notice that the probabilistic descent theorem shows the convergence principle of an adaptive form of Error Back-Propagation. In fact, a multi-layer feed-forward network, which is conventionally designed with minimum squared error criterion and Error Back-Propagation, can be designed in a manner more consistent with classification by using MCE/GPD.

There are several attempts to pursue the minimum classification error situation: e.g., a distance classifier using a traditional, piece-wise linear error rate function [13] and a multi-layer perceptron using the Classification Figure of Merit [14]. MCE/GPD is quite different from these in terms of both development philosophy and resulting formalization. On the other hand, Learning Vector Quantization (LVQ) too is a design method aiming at misclassification reduction, though it was intuitively developed, particularly without explicit measurement of error counts [15-17]. Interestingly, LVQ can be formalized as a simplified implementation, specially prepared for a multi-reference Euclidian distance classifier, of MCE/GPD. The detailed relation with LVQ is shown in [1, 18]. It is worth pointing out here that using LVQ is a useful implementation of MCE/GPD.

2.5 Applications

MCE/GPD has been vigorously applied to speech pattern classification, and its promising capability has clearly been demonstrated.

Applications to a multi-layer feed-forward network, particularly likelihood network and distance network, are described in detail in [18], where the effectiveness of MCE/GPD was observed on the Fisher iris task. Let us introduce here a corollary-like generalization of results in [18]; i.e., a mixture-distribution continuous HMM classifier can be formalized in a generalized form by assigning Markov states to an output node of a three-layer likelihood network.

Application to speech pattern classification was started in a somewhat limited way, using a DTW classifier [19, 20]. A limited implementation in a hybrid form was propsoed in [21] too. Full application to DTW systems were performed in [22, 23]. In particular, [22] studied the smoother case of the minimum search operation of Eq. (12), and demonstrated the effectiveness of using multiple normalization paths. Application to HMM systems was specially formalized as segmental GPD and showed great promise [24]. Successful results for HMM classifiers were also observed in [25]. It should be addressed that these HMM applications showed an important departure from a rather simple classification of isolated-mode speech utterances; i.e., they provided a training mechanism for applying MCE/GPD to classification of arbitrary speech segments such as subwords, words, and phrases. This extended idea has proved to be useful in a DTW classifier too [26]. Furthermore, [27], where speaker mapping was trained based on MCE, showed a new direction of the application. [28] showed the MCE/GPD superiority in a noisy speech classification. Application of MCE/GPD is still in the beginning stage, however results so far all clearly demonstrate its promising capability.

3 Task Formalization Using Smooth Functions

We have considered in this paper that the pattern recognition process consists of feature extraction and pattern classification. However, a real pattern recognition process is more complex. For example, in speech pattern recognition, the classification process should closely relate to a language process which may decide *a priori* probabilities. Moreover, although a simple speech classification scheme assumes that a sample, i.e., a speech segment, is extracted beforehand from continuous utterances, a real speech recognizer needs to include this segmentation process. It is certainly desirable that MCE/GPD can handle all these real situations properly; doing so may be a real goal of the MCE/GPD approach. MCE/GPD actually possesses a great potential which allows one to design recognizers that are even more general than in the above application studies. A recent study on minimum spotting error learning is showing signs of success in this new, advanced application [29].

Our simple 2-stage definition of pattern recognition suggests a straightforward extension of MCE/GPD application; i.e., an MCE/GPD design for both feature extraction and classification [30]. Here, the original sample is passed to the feature-extraction and classification stages in a consistent manner, directly aimed at the (locally) minimum classification error objective. We call this extended use of MCE/GPD Discriminative Feature Extraction (DFE).

DFE is essentially equivalent to MCE/GPD. Therefore, we don't need any new, specific formalization. One may embed a feature extraction process, conditioned by the given task and available resources, in the MCE/GPD functional form. MCE/GPD is mainly based on statistics, far from heuristics. However, expertise specific to a task is certainly useful in this mathematical approach. For example, applying DFE to acoustical speech utterances directly would be rather foolhardy, or rather, it would be more realistic to employ a power spectrum, which is prepared based on speech science knowledge, as input to the recognizer. Implementation of this new concept is thus task-dependent. DFE applications for speech recognition are described in detail in the next section.

4 Discriminative Feature Extraction for Speech Recognition

4.1 Various realizations

Mainly based on knowledge of hearing and speech perception, speech is usually represented, for the purpose of recognition, as a sequence of short-time power spectra or related parameter vectors. This kind of extraction, i.e., power spectrum sequence, is certainly a proper base for an effective DFE application.

A short-time power spectrum is generally computed by using FFT or autoregressive modeling. This is sometimes computed with a band-pass filter bank. Frequency scaling is usually linear, or Bark scale, or Mel scale. Spectrum intensity is often scaled logarithmically. The idea of weighting too is widely used to control feature sensitivity. As is well known, there are many conventional realizations of such sequences. However, most of these realizations are based on analysis of human capability, and thus are not necessarily directly applicable to statistically-designed machine recognition.

DFE attempts to accomplish extractions from the standpoint of minimizing misclassifications. In place of the Bark scale, a new frequency scaling could be found. A linear representation based on autoregressive modeling too could be extended to a discriminative non-linear version. ¿From among many possibilities, we specifically focus in this paper on cepstrum region design of power spectrum.

4.2 Application to lifter design

A short-time logarithmic power spectrum pattern is converted to a cepstrum vector through the Inverse Fourier Transform. In this conversion, frequency is mapped to quefrency, which corresponds to time. Let us consider a cepstrum pattern sequence as the recognizer input. It is well known that phoneme class identity, which is useful for speech recognition, locally exists in the low quefrency region. Therefore, a conventional speech recognizer selectively uses this narrow region cepstrum components as a feature for classification, by using a time window called a lifter. Notice that liftering (applying a lifter to a cepstrum vector) performs the feature extraction. A liftered cepstrum sequence is the pattern to be classified. Fig. 1 illustrates this recognition process, i.e., the recognizer structure consisting of a lifter and a post-end classifier.

A liftered cepstrum sequence pattern may represent (phoneme) class identity more properly than an unliftered cepstrum pattern. The question here is how to design a good lifter. Conventionally, the duration of lifter is chosen so as to suppress the cepstrum components due to glottal source. Usual lifter shapes are those of lag or time windows, e.g., Hamming window, whose properties have been extensively analyzed in spectrum estimation theory. In a somewhat advanced case, a lifter is designed over design samples so that cepstrum components relevant to classification can be emphasized [31]. However, clearly, these lifters, designed independently of the minimum classification error situation, are not guaranteed to be optimal.

DFE consistently designs both the lifter and the post-end classifier within the MCE/GPD framework. An arbitrary system structure can be used for the post-end classifier. By way of example, we use a multi-layer feed-forward network. Our recognizer is illustrated in Fig. 2. To simplify analysis of the lifter design, each node of the bottom lifter layer has only a vertical connection. The discriminant function here is each of the network outputs. MCE/GPD is then implemented accordingly.

As a preliminary evaluation, we conducted experiments on the task of classifying Japanese five-category vowels. We used speech data of 100 phonetically-balanced sentences, spoken by 5 speakers (3 males and 2 females) and recorded at 12 kHz sampling rate. The recognizer input was just a fixed-dimensional cepstrum vector which corresponds to a single time-windowed vowel segment; i.e., our sample was not a dynamic pattern. Each sample was prepared as follows: 1) A center segment of vowel was extracted by using a 42 msec Hamming window from the database. 2) The extracted speech signal was then converted to a 256-point cepstrum vector by using FFT/IFFT. We collected 3,500 samples in total; half for design and half for training.

The recognizer was investigated with different settings for experimental conditions such as recognizer size, and produced the highest accuracy, 96.8% on design data and 88.7% on testing data. For comparison, we also evaluated the conventional use of a rectangular lifter: The rectangular lifter was realized on the lowest lifter layer by assigning two constants, 1 and 0, to the connection weights: Only the post-end classifier was trained. This conventional way could not excel DFE in recognition accuracy. Although different lifter lengths were carefully tried, only 89.1% and 87.3% were attained on design and testing data, respectively.

A lifter example of the DFE design is shown in Fig. 3. This lifter clearly suppresses cepstrum components in a high quefrency region which is usually occupied by information irrelevant to phoneme classification. The lifter also suppresses extremely-low quefrency components, which probably correspond to speaker identity. It is likely that DFE successfully distinguished vowel class identity from other features such as speaker identity. Observing the spectrum domain helps understand the results. Fig. 4 shows two logarithmic power spectra of a single input: one was calculated without liftering, meaning that this corresponds to the input cepstrum vector, and one was calculated by using the lifter in Fig. 3. The smoothed, liftered spectrum removes the harmonic structure due to vocal source excitation, and brings out the spectrum envelope, which mainly corresponds to the phoneme class identity.

The experimental results in the above paragraph indicate the fundamental possibility that our new design method can be superior to conventional methods. The DFE learning method is an automatic and efficient way of extracting possible feature parameters.

The results over the testing samples may need further analysis. The difference in the DFE results between design and testing data should be studied from the viewpoint of training robustness. The fact that the lifter in Fig. 3 is not so smooth must relate to this big drop. Similar to highly-discriminative, nonlinear artificial neural networks, the smoothness of feature extraction should be carefully studied in our approach too.

5 Summary

In this paper, we summarized the new discriminative learning theory called MCE/GPD and also introduced Discriminative Feature Extraction as one of its extended applications. A motivation underlying the proposed method is to formalize the entire task at hand in a smooth functional form and efficiently find a practical solution on this form. Our approach provides a straight-forward and sound basis for the realization of long standing minimum classification error pattern recognition problem.

Acknowledgements
The authors would like to thank Mr. Erik McDermott and Mr. Takashi Komori for their insightful discussions and patient support in accomplishing this study.

References

[1] S. Katagiri, C.-H. Lee, and B.-H. Juang; *A Generalized Probabilistic Descent Method*, ASJ, Proc. of Fall Meeting, 2-p-6, Vol. 1, pp. 141-142 (1990. 9).

[2] S. Katagiri, C.-H. Lee, and B.-H. Juang; *New Discriminative Training Algorithms Based on the Generalized Probabilistic Descent Method*, IEEE, Neural Networks for Signal Processing, pp.299-308 (1991. 9).

[3] B.-H. Juang, and S. Katagiri; *Discriminative Learning for Minimum Error Classification*, to be published in IEEE, Trans. on SP (1992. 12).

[4] B.-H. Juang, and S. Katagiri; *Discriminative Training*, to be published in J. Acoust. Soc. Jpn. (E), (1992. 11).

[5] S. Amari; *A Theory of Adaptive Pattern Classifiers*, IEEE, Trans. on EC, Vol. EC-16, No. 3, pp. 299-307 (1967. 6).

[6] D. Howell; *The Multi-Layer Perceptron as a Discriminative Post Processor for Hidden Markov Networks*, FASE, Proc. of 7th FASE Symposium - Speech, pp. 1389-1396 (1988).

[7] H. Sakoe, R. Isotani, K. Yoshida, and T. Watanabe; *Speaker Independent Word Recognition Using Dynamic Programming Neural Networks*, IEEE, Proc. of ICASSP89, Vol. 1, pp.29-32 (1989. 5).

[8] H. Iwamida, S. Katagiri, E. McDermott, and Y. Tohkura; *A Hybrid Speech Recognition System Using HMMs with an LVQ-Trained Codebook*, ASJ, J. Acoust. Soc. Jpn. (E), Vol. 11, No. 5, pp. 277-286 (1990. 9).

[9] D. Kimber, M. Bush, and G. Tajchman; *Speaker-Independent Vowel Classification Using Hidden Markov Models and LVQ2*, IEEE, Proc. of ICASSP90, Vol. 1, pp.497-500 (1990. 4).

[10] G. Yu, W. Russell, R. Schwartz, and J. Makhoul; *Discriminative Analysis and Supervised Vector Quantization for Continuous Speech Recognition*, IEEE, Proc. of ICASSP90, Vol. 2, pp.685-688 (1990. 4).

[11] Y.-Q. Gao, T.-Y. Huang, and D.-W. Chen; *HMM-Based Warping in Neural Networks*, IEEE, Proc. of ICASSP90, Vol. 1, pp. 501-504 (1990. 4).

[12] S. Katagiri, and C.-H. Lee; *A New HMM/LVQ Hybrid Algorithm for Speech Recognition*, IEEE, Proc. of GLOBECOM90, 608.2, Vol. 2, pp.1032-1036 (1990. 12).

[13] A. Ando, and K. Ozeki; *A Clustering Algorithm to Minimize Recognition Error Function*, IEICE, Trans. of IEICE (A), Vol. J74-A, No. 3, pp. 360-367 (1991. 3)(in Japanese).

[14] J. Hampshire, and A. Waibel; *A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks*, IEEE, Trans. on NN, Vol. 1, No. 2, pp.216-228 (1990. 6).

[15] T. Kohonen; *Learning Vector Quantization for Pattern Recognition*, Helsinki University of Technology, Report TKK-F-A601 (1986. 11).

[16] E. McDermott; *LVQ3 for Phoneme Recognition*, ASJ, Proc. of Spring Meeting, 2-P-16, Vol. 1, pp. 151-152 (1990. 3).

[17] T. Kohonen; *The Self-Organizing Map*, IEEE, Proc. of IEEE, Vol. 78, No. 9, pp. 1464-1480 (1990. 9).

[18] S. Katagiri, C.-H. Lee, and B.-H. Juang; *Discriminative Multi-Layer Feed-Forward Networks*, IEEE, Neural Networks for Signal Processing, pp.11-20 (1991. 9).

[19] P.-C. Chang, and B.-H. Juang; *Design of Discriminant Functions for Distortion Sequences in Dynamic Pattern Matching for Speech Recognition*, ASA, J. Acoust. Soc. Am., Suppl. 1, 5SP4, Vol. 88, p. S102 (1990, 11).

[20] P.-C. Chang, S.-H. Chen, and B.-H. Juang; *Discriminative Analysis of Distortion Sequences in Speech Recognition*, IEEE, Proc. of ICASSP91, Vol. 1, pp.549-552 (1991. 5).

[21] W.-Y. Chen, and S.-H. Chen; *Word Recognition Based on the Combination of a Sequential Neural Network and the GPDM Discriminative Training Algorithm*, IEEE, Neural Networks for Signal Processing, pp. 376-384 (1991. 9).

[22] P.-C. Chang, and B.-H. Juang; *Discriminative Template Training for Dynamic Programming Speech Recognition*, IEEE, Proc. of ICASSP92, Vol. 1, pp. 493-496 (1992. 3).

[23] T. Komori, and S. Katagiri; *Application of a Generalized Probabilistic Descent Method to Dynamic Time Warping Based Speech Recognition*, IEEE, Proc. of ICASSP92, Vol. 1, pp. 497-500 (1992. 3).

[24] W. Chou, B.-H. Juang, and C.-H. Lee; *Segmental GPD Training of HMM Based Speech Recognition*, IEEE, Proc. of ICASSP92, Vol. 1, pp. 473-476 (1992. 3).

[25] D. Rainton, and S. Sagayama; *Minimum Error Classification Training of HMMs -Implementational Details and Experimental Results-*, ASJ, Tech. Report SP91-107, pp. 39-46 (1992. 1).

[26] E. McDermott, and S. Katagiri; *Prototype-Based Discriminative Training for Various Speech Units*, IEEE, Proc. of ICASSP92, Vol. 1, pp. 417-420 (1992. 3).

[27] M. Sugiyama, and K. Kurinami; *Minimal Classification Error Optimization for a Speaker Mapping Neural Networks*, IEEE, Neural Networks for Signal Processing II, pp. 233-242 (1992. 8).

[28] K. Ohkura, D. Rainton, and M. Sugiyama; *Noise-Robust HMMs Based on Minimum Error Classification*, ASJ, Proc. of Fall Meeting, 1-7-14, Vol. 1, pp. 73-74 (1992. 10)(in Japanese).

[29] T. Komori, and S. Katagiri; *GPD Training for Spotting*, ASJ, Proc. of Fall Meeting, 2-Q-12, Vol. 1, pp. 195-196 (1992. 10).

[30] A. Biem, and S. Katagiri; *Cepstrum Liftering Based on Minimum Classification Error*, ASJ, Tech. Report SP92-26, pp. 17-24 (1992. 6).

[31] Y. Tohkura; *A Weighted Cepstral Distance Measure for Speech Recognition*, IEEE, Trans. on ASSP, Vol. ASSP-35, No. 10, pp. 301-309 (1987. 11).

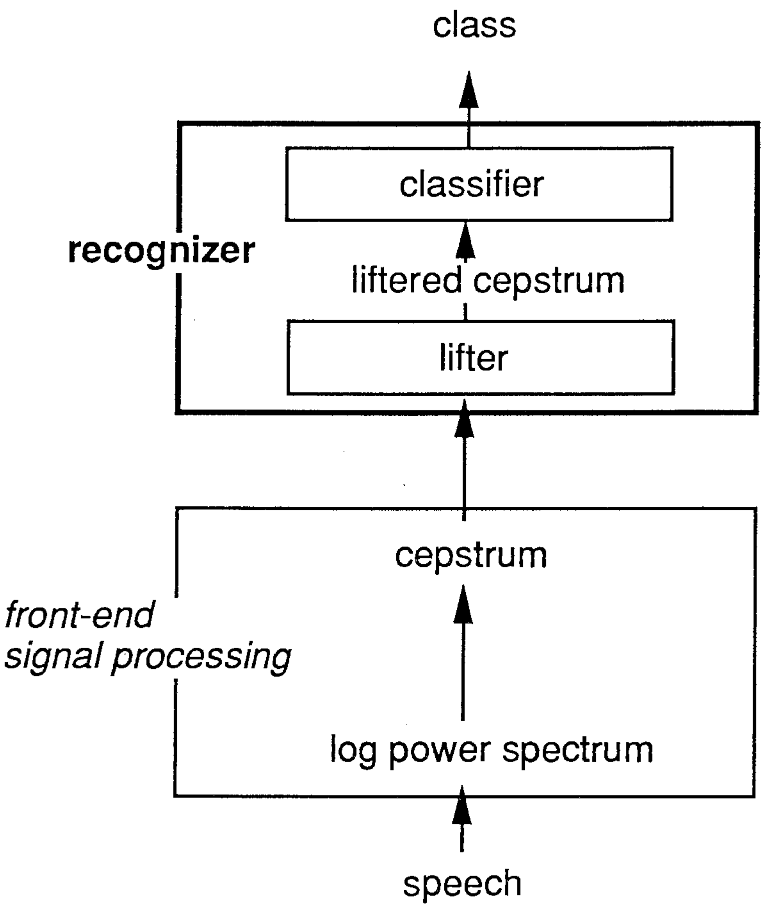

Figure 1. Speech recognition using cepstrum.

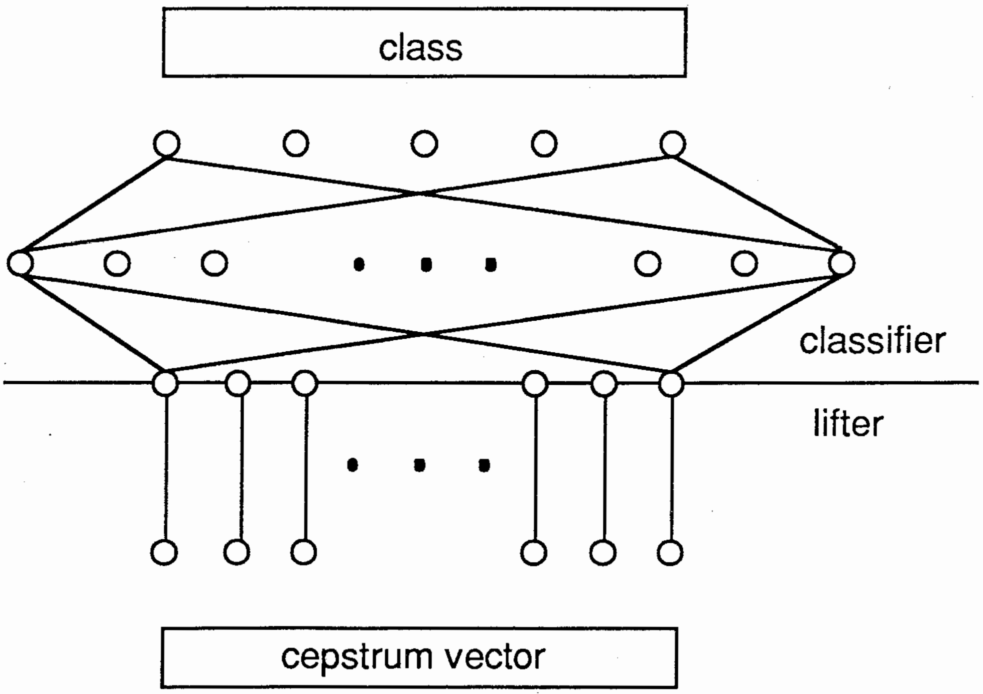

Figure 2. A four-layer feed-forward network recognizer including a lifter.

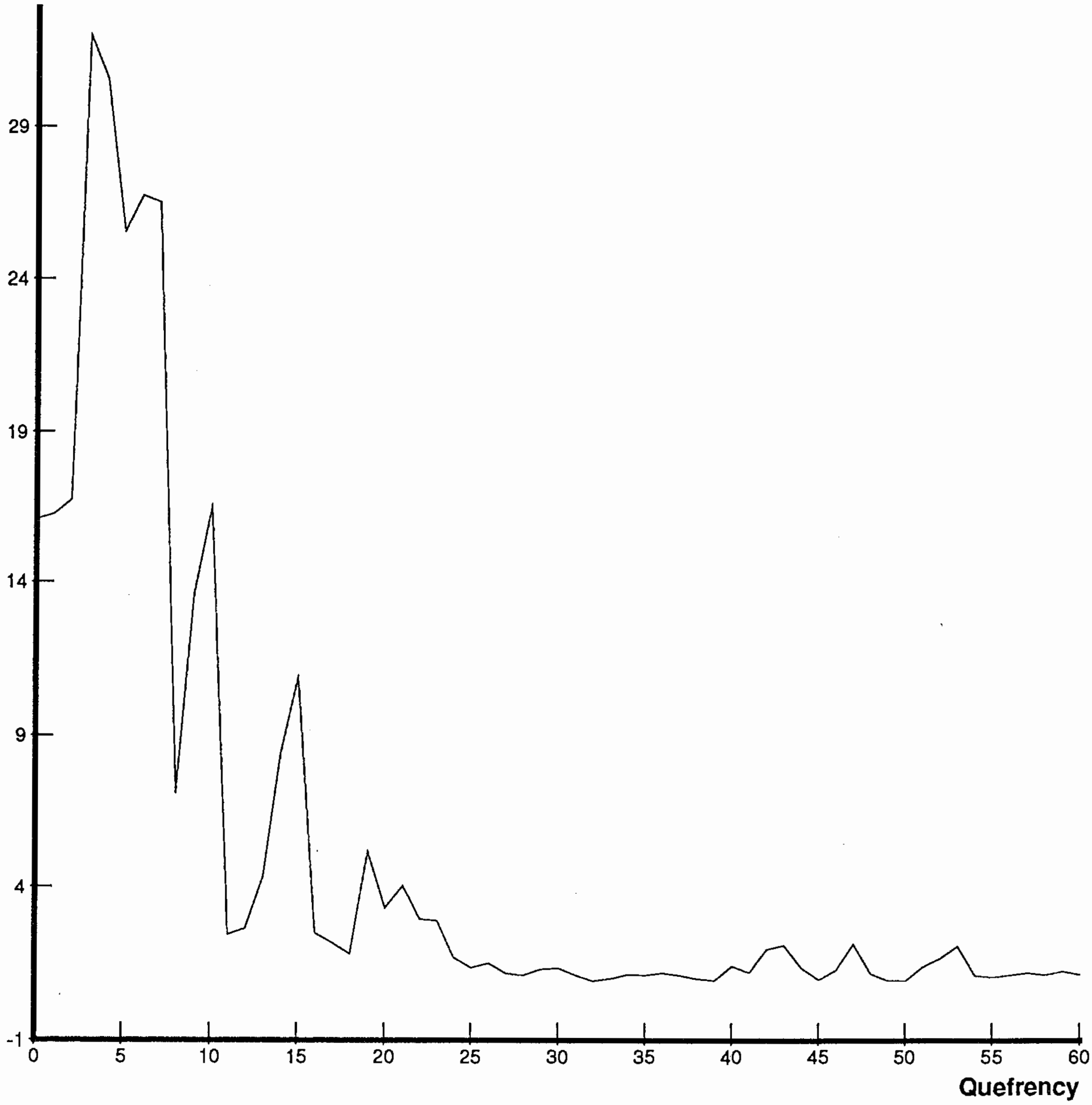

Figure 3. A lifter example of the Discriminative Feature Extraction Design.

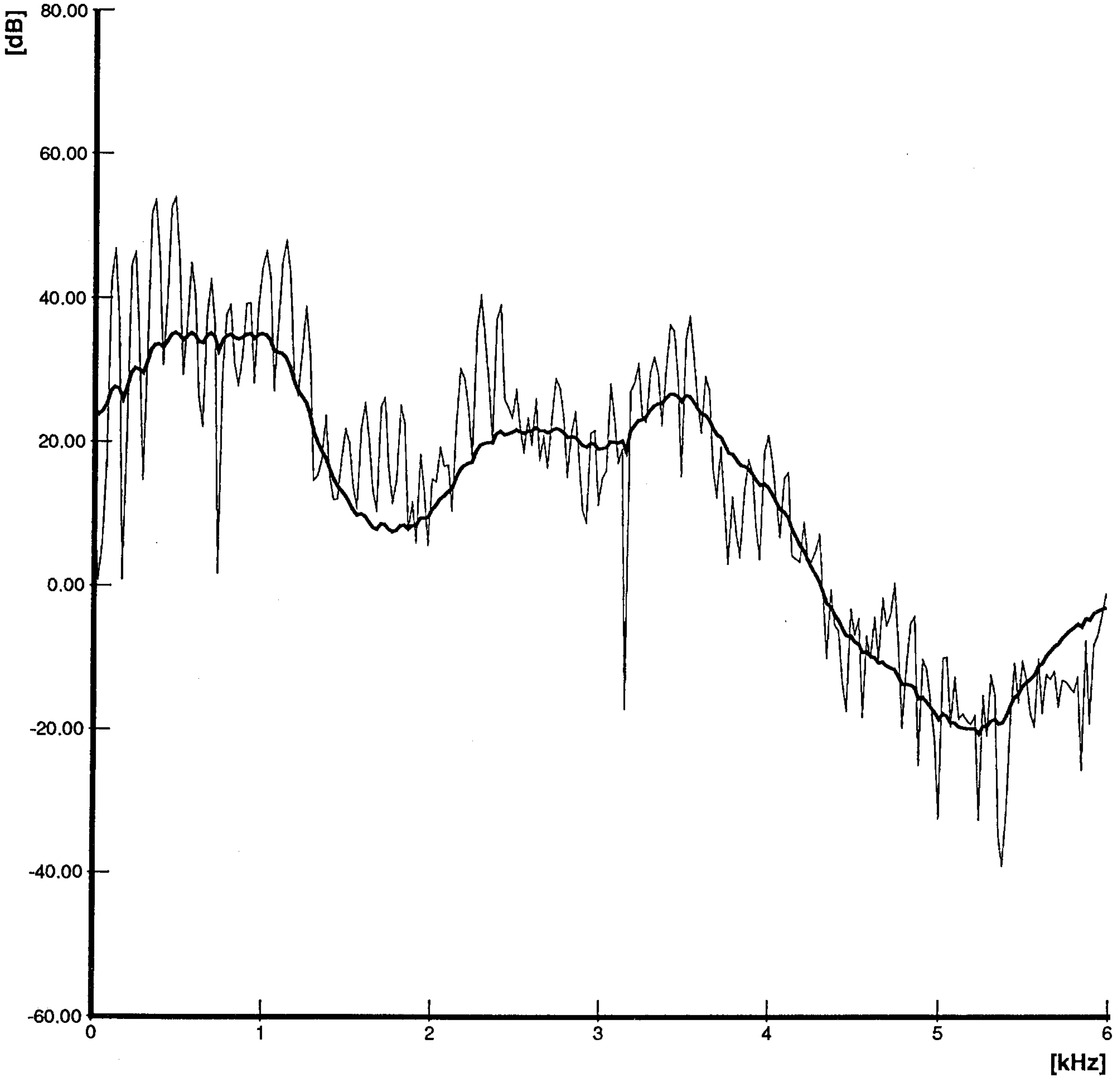

Figure 4. Logarithmic power spectra of a single input cepstrum example: one calculated without liftering (thin curve) and one calculated by using the lifter in Fig. 3 (thick curve).