LSF QUANTIZATION WITH MULTIPLE SCALE LATTICE VQ FOR TRANSMISSION OVER NOISY CHANNELS *

Adriana Vasilache and Ioan Tăbuş
Institute of Signal Processing, DMI,
Tampere University of Technology,
PO Box 553 FIN-33101, Tampere, Finland
Tel: +358 (0) 3 3115 3976; fax: +358 (0) 3 3115 3857
e-mail: {adrianav, tabus}@cs.tut.fi

ABSTRACT

In this paper we discuss the performance of several enumeration techniques for lattice quantizers when the transmission channel is noisy. We consider the multiple scale leader lattice quantization structure and we test three variants of lattice vectors indexing for LSF parameters quantization. We compare the results obtained with the proposed methods and also compare with the results obtained with the codec G.729, in the presence of noise.

1 INTRODUCTION

The multiple scale lattice vector quantization (MSLVQ) has been introduced in [10] and extended to multiple scale leader lattice vector quantization (MSLLVQ) in [8]. These structures performed very well for the LSF quantization, but the problem of error resilience has not been taken into account. The large number of codevectors prohibits the use of an index assignment as, in the some practical applications, this will mean the storage of the permutations of the index assignment as a table of 2^{18} or 2^{19} integers. The channel distortion optimization of the codebook, besides the fact that it should be made over far too many codevectors, would also destroy the structure of the codebook. Therefore, the only possibility to deal with a noisy channel is an adequate choice of the enumeration algorithm of the lattice codevectors.

Several lattice enumeration techniques have been proposed over the years for different truncations and lattice types. For instance, Fischer introduced the first enumeration technique on pyramid truncated lattice in 1986 [1], further modified in [2] in order to improve the channel robustness. These algorithms apply mainly to Z_n lattices. An indexing technique based on Schalkwijk formula and on the notion of leader vector of a lattice is developed in [6] for Z_n and D_n lattices. We have also proposed a method based on leader vectors for Z_n and D_n lattices and used it in conjunction with lattice entropy coding in [9]. Recently Rault and Guillemot [7] have presented an enumeration based on signed leaders or generated signed leaders valid for a large class of lattices (Z_n, A_n, D_n and D_n^{++}). However, with the exception of [2], the error resilience over the channel has not been taken into account.

In this paper we present the MSLLVQ structure and introduce several enumeration algorithms for it. The enumerations are based on the notion of leader class and they apply to lattices which are invariant to signed permutations with or without constraints. The enumeration methods are tested in an application concerning LSF parameters quantization.

2 MSLVQ STRUCTURE

2.1 General notions

An n-dimensional lattice Λ is a set of real vectors whose coordinates are integers in a given basis $\{b_i \in \mathbb{R}^n\}_{\overline{1,n}}$:

$$\Lambda = \{v \in \mathbb{R}^n \mid v = \sum_{i=1}^{n} \alpha_i b_i,\ \alpha_i \in \mathbb{Z}\}. \quad (1)$$

In this paper we use the D_n^+ lattice which can be obtained from the D_n lattice as follows

$$D_n = \{x \in \mathbb{Z}^n | \sum_{i=0}^{n} x_i \text{ is an even integer}\}$$

$$D_n^+ = D_n \cup (w_1 + D_n),\ \text{where } w_1 = (1/2, 1/2, \ldots, 1/2).$$

When used as a VQ, the lattice has to be truncated to the finite number of vectors allowed by the finite rate R. Our goal is to deal with sources whose distribution possesses a certain symmetry property (e.g. spherical, pyramidal) and, therefore, we consider truncations reflecting those symmetries. A *truncation* of a lattice Λ is defined as the set of lattice points having the norm less or equal to a given value K, $\overline{\Lambda} = \{x \in \Lambda | N(x) \leq K\}$, where $N(x)$ is the selected norm of x. If $N(x)$ is the Euclidean norm the truncation is spherical while in the case of the l_1 norm the truncation is pyramidal.

A *leader* vector v of a truncated lattice $\overline{\Lambda}$ is a vector which has positive elements ordered decreasingly. Every such a vector defines a subset of vectors in $\overline{\Lambda}$, dubbed as the *leader class*, containing all vectors obtained by a signed permutation of v, described in more detail by the following two properties:

(1) Together with v, all the vectors obtained by permutations of entries in v belong to $\overline{\Lambda}$, if Λ is invariant under permutation, which is the case for the lattices we consider in this paper.

(2) The sign combinations —with some possible constraints— of the elements of these vectors result in vectors from Λ. The vectors obtained through permutations and (constrained) sign switching form the class of the leader vector v.

With these definitions for the leader vector and the leader class, a truncation of a lattice can be represented as the union of several leader classes.

*This work has been supported by Academy of Finland, project No. 44876 (Finnish Centre of Excellence Program (2000-2005))

A truncated lattice used as a vector quantizer means that the codebook is a scaled truncated lattice: $\mathcal{C} = \{s \cdot c_j | c_j \in \overline{\Lambda}, s \in \mathbb{R}\}$, where s is the scaling factor. The nearest neighbor (NN) search in the codebook can be performed by using, for instance, the search on leaders [8].

2.2 Multiple scale leader-lattice VQ (MSLLVQ)

The multiple-scale lattice VQ was introduced in [10] as a lattice VQ which is composed of the union of copies of the same truncation of a lattice scaled with different scales.

This structure can be extended to two more flexible variants, dubbed as MSLLVQ-A and MSLLVQ-B [8].

The codebook (2) of a MSLLVQ-A structure consists of several *different* truncations (at different radii) of the lattice scaled with *different* scales.

$$\mathcal{C}_1 = \{s_i \cdot c_j^i | s_i \in \Sigma, c_j^i \in \overline{\Lambda}_i^*\}, \tag{2}$$

The sets $\overline{\Lambda}_i^*$, $i = \overline{1,T}$, are the truncations of the lattice Λ. The asterisk $(*)$ marks the fact that not necessarily all the leader classes of $\overline{\Lambda}_i$ are considered, some of those belonging to the last shell of the truncation being possibly ignored. All vectors in $\overline{\Lambda}_i^*$ are scaled with the same scale s_i.

The MSLLVQ-B structure considers the lattice as a union of leader classes and the codebook (3) is obtained as a superposition of different lattice truncations but each leader class inside the truncations has its own scale,

$$\mathcal{C}_2 = \{s_{ik} \cdot c_j^{ik} | s_{ik} \in \Sigma_i, c_j^{ik} \in \mathcal{V}^{ik}, \mathcal{V}^{ik} \subset \overline{\Lambda}_i^*\}, \tag{3}$$

$\mathcal{V}^{ik}$ is the k-th leader class of the lattice truncation $\overline{\Lambda}_i^*$ and Σ_i is the set of scales that multiply the vectors in leader class $\overline{\Lambda}_i^*$. Note that for codebooks of same size the total number of scales in MSLLVQ-B is larger than in the case of MSLLVQ-A. Note also that the leader classes $\mathcal{V}^{ik} \subset \Lambda_i$ can also be chosen independently of any truncation $\overline{\Lambda}_i$ (i.e. not all the leader classes in a truncation are considered) but the choice of the leader classes becomes a combinatorial optimization task. We do not use this flexibility, and note that the definition (3) implicitly assumes that the truncation $\overline{\Lambda}_i$ is well suited to the input vector distribution.

3 ENUMERATION OF MSLLVQ

In order to proceed with the enumeration of a codevector from an MSLLVQ structure we need an enumeration technique for a single scale lattice.

We present here an enumeration method derived from those used in [6] and [9] and propose different variations.

The existing enumerations for Z_n lattices presented in [1], [2] are not easily generalized to other lattices. The enumeration methods considered in [6] and [9] apply to Z_n as well as D_n lattices and are based on the leader classes of a lattice. We will extend them to D_n^+ and D_n^* lattices.

3.1 Enumeration on leaders

The cardinality of a leader class can be determined from the components of its leader vector. Consider the n-dimensional leader vector

$$v = (v_m \dots v_m \dots v_i \dots v_i \dots v_1 \dots v_1),$$

where m is the number of different values (zero included) taken by v_i, in the vector. There are n_i components equal to v_i in the leader vector. The leader class has the parity α. The number of non-zero components of the leader vector is k. The cardinality of the leader class $\mathcal{V}$ is thus given by the polynomial coefficient

$$|\mathcal{V}| = 2^{k-|\alpha|} \binom{n}{n_1 \dots n_m} = 2^{k-|\alpha|} \frac{n!}{n_1! n_2! \dots n_m!}. \tag{4}$$

As remarked in [2], the information concerning the distribution of signs is very sensitive to channel errors. Therefore it should be extracted and isolated, for instance, in the least significant bits of the codevector index. Also it should be noticed that in a leader class the index for the signs (see subsection 3.1.3) has the same number of bits and is placed on the same positions in the resulting index. Therefore the indexing of signs and the indexing of the component distributions of a vector from a leader class can be done independently and, further, independently optimized.

3.1.1 Leader classes order

The order in which the leader classes are enumerated affects the error resilience of the lattice enumeration. However, as in the practical situations we consider, the number of leader classes is approximately 10, it is almost impossible to consider all the permutations of the leader classes. Therefore, only several permutations can be considered, taking into account, for instance, the norms or the number of non zero components of the vectors in the leader class. Usually the leader classes are ordered by their norms.

From this point on, we face two problems: how we enumerate the distributions of signs (sign enumeration) in the vectors and how we enumerate the distribution/position (position enumeration) of the components in the vectors.

3.1.2 Position enumeration

In this subsection, given a leader vector, we present different possibilities to enumerate the distinct permutations of its elements.

Lexicographical order (P1)

One of the first choice on how to enumerate the vectors of a leader class is to order them lexicographically. We say that the vector $(x_1^{(1)}, \dots, x_n^{(1)})$ precedes lexicographically $(x_1^{(2)}, \dots, x_n^{(2)})$ if $\exists j$ such that $x_j^{(1)} < x_j^{(2)}$ and $\forall i < j$, $x_j^{(1)} = x_j^{(2)}$. From the definition of the leader class we have that $v_1 < \dots < v_m$. The lexicographical enumeration can be explained as being based on the following identity

$$\binom{n}{n_1 \dots n_m} = \binom{n-1}{n_1 - 1 \dots n_m} + \dots + \binom{n-1}{n_1 \dots n_m - 1} \tag{5}$$

the terms on the right side of the equality representing the number of vectors that start with v_1, with v_2 and so on. An example of an algorithm realizing this enumeration can be found in [5] or [7].

Counting binomial coefficients (P2)

The part, corresponding to the positions of the components from the cardinality of a leader class (4), can also be written as

$$\binom{n}{n_1 \dots n_m} = \binom{n}{n_1} \binom{n - n_1}{n_2} \dots \binom{n - \sum_{i=1}^{m-1} n_i}{n_m}. \tag{6}$$

Therefore, an algorithm counting in how many ways n_1 values v_1 can be put on n positions, then in how many ways n_2

Index	P2|S				P1|S			
000000	3	1	0	0	0	0	1	3
000001	3	-1	0	0	0	0	1	-3
000010	-3	1	0	0	0	0	-1	3
000011	-3	-1	0	0	0	0	-1	-3
000100	3	0	1	0	0	0	3	1
001000	3	0	0	1	0	1	0	3
001100	1	3	0	0	0	1	3	0
010000	0	3	1	0	0	3	0	1
010100	0	3	0	1	0	3	1	0
011000	1	0	3	0	1	0	0	3
011100	0	1	3	0	1	0	3	0
100000	0	0	3	1	1	3	0	0
100100	1	0	0	3	3	0	0	1
101000	0	1	0	3	3	0	1	0
100100	0	0	1	3	3	1	0	0

Table 1: Illustration of P1 and P2 position enumerations for the leader class (3 1 0 0).

values v_2 can be put on $n - n_1$ positions and so on, is obtained. The recursive function used to enumerate binomial coefficients is given by the following equation

$$c2i(n,p,q) = \begin{cases} p[0], & q=1 \\ \sum_{i=1}^{p[0]} \binom{n-i}{q-1} + & \\ +c2i(n-p[0]-1, p+1, q-1), & q>1 \end{cases} \tag{7}$$

where n is the dimension of the vector (available positions), q the number of values that are placed on the positions $p[0], p[1], \ldots, p[q-1]$ (C language notation is used for the pointer p). In Table 1 the two position enumerations are illustrated.

3.1.3 Sign enumeration

Leader class with null parity

A zero value parity for a leader class means that there is no constraint on the number of either positive or negative components in a vector from that class. A very simple indexing method for the vector of signs can be realized by assigning to each strictly negative component the value 1 and to each strictly positive component the value 0 the resulting binary string being the index I_S.

Leader class with non-null parity

If a leader class has a non-null parity, α, 1 or -1, then its vectors should have only an even or odd number of negative components, which makes unusable the previous method of indexing the signs.

The principle of the sign enumeration in this case is to order the vectors after the number of negative components. Thus, if the parity is -1, we will have first the vectors with 1 negative component, then $3, 5, \ldots$ negative components and so on, but the vectors are from the same leader class. The corresponding index is:

$$I_S = \begin{cases} \sum_{i=0}^{2i<n_-} \binom{n}{2i} + c2i(k,p,n_-) & \alpha = 1 \\ \sum_{i=1}^{2i+1<n_-} \binom{n}{2i+1} + C2I(k,p,n_-) & \alpha = -1 \end{cases} \tag{8}$$

where k is the number of non zero components of the vector, n_- is the number of negative components of the vector and p is an array specifying the position of the negative components relative to n_s.

3.2 Multiple scale lattice enumeration

If there are several leader classes or several lattice truncations the indexing of their vectors is realized taking into account the order of the leaders discussed in subsection 3.1.1. Thus the index of a codevector is represented as:

Scale / L / P	Sign

where L enumerates the leader class and P enumerates the components positions in the vector. The bits for the sign are separated by a vertical line because the same number of bits is used for the sign distribution in a given leader class (the total number of sign distributions in a leader class is a power of 2).

4 RESULTS

4.1 Application to LSF quantization

Linear predictive coding (LPC) method is one of the most popular approaches used for describing the short-term spectrum of speech signal. Due to the quasistationary nature of speech a new prediction filter is computed for each frame of 10 ms or 20 ms. The process of quantizing the filter coefficients to a finite number of bits/frame is known as LPC quantization.

The spectral distortion is often used as an objective measure of the quantization performance:

$$SD = \left\{ \frac{100}{\pi} \int_0^{\pi} \left[\log_{10} |A_n(e^{j\omega})|^2 - \log_{10} |\hat{A}_n(e^{j\omega})|^2 \right]^2 d\omega \right\}^{1/2}, \tag{9}$$

where SD is given in dB, $A_n(e^{j\omega})$ and $\hat{A}_n(e^{j\omega})$ are the spectra of the n-th speech frame, without and with quantization, respectively.

4.1.1 LPC Analysis

A speech coder based on LPC extracts the information concerning the short-term spectral envelope by using an all-pole filter $H(z) = 1/A(z)$, where $A(z)$ is the prediction filter, given by $A(z) = 1 + a_1 z^{-1} + \ldots + a_p z^{-r}$. The order r of the filter is usually 10. A description of the filter $A(z)$ must be communicated to the receiver for each frame. To ensure the stability of the all-pole filter $H(z)$, it is necessary to transform the LPC coefficients in other representations, usually the LSF representation [4]. Since we wanted to compare the quantization results with those obtained by the G.729 speech codec, the LPC analysis part is the same as for the codec, described thoroughly in [3].

4.1.2 LSF quantization with MSLLVQ

We experimented the MA predictive multiple-scale lattice VQ for the quantization of LSF parameters. That is, seeking to eliminate inter-frame correlations, the prediction errors of the LSF parameters are quantized, instead of the actual LSFs. The MSLLVQ structures have given very good results for LSF quantization for an error-free channel, reducing by 14% the spectral distortion relatively to the codec G.729 and by more than 10 times the computational and storage complexity [8].

4.1.3 LSF quantization with MSLLVQ over noisy channels

We are interested in the influence of channel errors on the overall coding process. We have tested the indexing on leaders with the different enumerations of components positions presented in subsection 3.1.2 assuming a binary symmetric

BER	Sc/L/P3/Sg			Sc/L/P1/Sg		
	$\overline{SD}$	$[2,4]$	>4	$\overline{SD}$	$[2,4]$	>4
[%]	[dB]	[%]	[%]	[dB]	[%]	[%]
0.0	1.29	10.21	0.84	1.29	10.21	0.84
0.1	1.52	13.27	3.93	2.06	29.25	10.06
0.2	1.65	15.83	5.70	2.18	30.61	11.84
1.0	2.60	29.71	19.50	2.98	37.07	25.59

Table 2: Average SD and outliers in LSF quantization for MSLLVQ-A at 18 bits using P1 (lexicographical) and P3 (leaders ordered by the norm and the number of non-zero components) enumerations.

BER	Sc/L/Sg/P2			Sc/L/P2/Sg		
	$\overline{SD}$	$[2,4]$	>4	$\overline{SD}$	$[2,4]$	>4
[%]	[dB]	[%]	[%]	[dB]	[%]	[%]
0.0	1.29	10.21	0.84	1.29	10.21	0.84
0.1	1.46	13.47	2.92	1.44	13.41	2.67
0.2	1.60	16.08	4.90	1.60	16.01	4.38
1.0	2.63	29.74	20.36	2.52	30.04	18.25

Table 3: Average SD and outliers in LSF quantization for MSLLVQ-A at 18 bits using P2 (counting binomial coefficients) with the bits of sigs distributions on the least significant positions (Sc/L/P2/Sg) or inside the index (Sc/L/Sg/P2) enumerations.

channel. w-MSLLVQ means that in the NN search the selection between different leader classes is made according to a weighted distance [3]. The multiple scale structure is specified as $\sum_i n_i \times l_i$, where l_i is the last leader class that is contained in the truncation and n_i indicates how many times the corresponding truncation is considered in the codebook. The lattice used for quantization is the D_{10}^{+} lattice. The scales are trained on 115006 speech frames from the TIMIT speech database. The results are reported over 500000 speech frames from the TIMIT test set. From Tables 2 and 3 it can be observed that our enumeration based on the counting of polynomial coefficients leads to the best results, the lexicographical order leading to the poorest ones, the difference between the two being significant. We have also used an idea from the enumeration of [2] by changing the leader classes order such that for the same norm, the leader vectors should be decreasingly ordered by the number of non-zero components (P3). However this change does not bring any improvement as the number of leader classes with the same norm is quite small. From Table 3 we can observe the beneficent effect of the delimitation of the signs distribution bits as the least significant bits.

Finally we have compared the behavior over noisy channels for the MSLLVQ scheme and for the quantization scheme in the codec G.729 (Table 4). Although for small bit error rates the MSLLVQ scheme maintains its superiority, for larger values of bit error rates the codec shows a better error-resilience. This comparison strengthen the fact that the lattice enumeration is a very delicate matter in the presence of channel errors. At a closer look to the enumerations of positions it can be observed that there are in fact a multitude of ways to enumerate them and we are currently studying new procedures of enumeration optimization aimed at the MSLLVQ structure.

BER	Codec G.729			w-MSLLVQ-B		
	$\overline{SD}$	$[2,4]$	>4	$\overline{SD}$	$[2,4]$	>4
[%]	[dB]	[%]	[%]	[dB]	[%]	[%]
0.0	1.35	12.0	0.5	1.19	6.8	0.3
0.1	1.40	13.78	0.74	1.36	11.71	2.84
0.2	1.42	14.37	0.79	1.43	14.77	4.25
1.0	1.82	28.40	3.93	2.42	31.77	16.01
2.0	2.26	39.82	8.79	3.24	39.74	29.60

Table 4: LPC quantization with the codec G.729 over a binary symmetric noisy channel.

5 CONCLUSION

In this paper we have presented a quantization structure consisting of a union of lattice truncations or of lattice leader vectors differently scaled. We have introduced several enumeration techniques and we have studied their error-resilience in the presence of a noisy channel. The new quantization structure outperforms the quantization scheme of the codec G.729 for zero and small bit error rates, but still needs some further refinement of the enumeration methods at large bit error rates where the performance degrades significantly. We have pointed out that the enumeration method is very important in the presence of noisy channel, the error resilience being very sensitive to the choice of codevectors enumeration.

References

[1] T. R. Fischer. A pyramid vector quantizer. *IEEE Trans. on Information Theory*, IT-32(4):568–583, July 1986.

[2] A. C. Hung, E. K. Tsern, and T. H. Meng. Error-resilient pyramid vector quantization for image compression. *IEEE Trans. on Image Processing*, 7(10):1373–1386, October 1998.

[3] Coding of speech at 8kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP). ITU Telecommunications Standardization Sector, December 1995. Draft Recommandation.

[4] W. B. Kleijn and K. K. Paliwal, editors. *Speech coding and synthesis*. Elsevier Science, Amsterdam, 1995.

[5] O. Milenkovic and B. Vasić. Permutation (d, k) codes: efficient enumerative coding and phrase length distribution shaping. *IEEE Trans. on Information Theory*, 46(7):2671–2675, 2000.

[6] J.-M. Moureaux, P. Loyer, and M. Antonini. Low complexity indexing method for Z_n and D_n lattice quantizers. *IEEE Trans. on Communications*, 46(12):1602–1609, December 1998.

[7] P. Rault and C. Guillemot. Indexing algorithms for Z_n, A_n, D_n and D_n^{++} lattice vector quantizers. *IEEE Trans. on Multimedia*, 3(4):395–404, December 2001.

[8] A. Vasilache, B. Dumitrescu, and I. Tăbuş. Multiple-scale leader-lattice VQ with application to LSF quantization. to appear in Signal Processing, 2002.

[9] A. Vasilache and I. Tăbuş. Indexing and entropy coding of lattice codevectors. In *ICASSP'2001*, Salt Lake City, USA, May 2001.

[10] A. Vasilache, M. Vasilache, and I. Tăbuş. Predictive multiple-scale lattice VQ for LSF quantization. In *Proceedings of Int. Conf. Acoustics, Speech, and Signal Processing*, pages 657–660, Phoenix, Arizona, March, 15-19 1999.