Non-Statistical Soft-in, Soft-out Decoding with the Euclidean Metric

Prof Patrick Guy Farrell, FIMA

Universities of Lancaster and Kent, UK
Paddy.Farrell@virgin.net

Summary

Soft-decision decoding of linear error correcting codes, using a quantised or real number soft metric such as Euclidean distance, has been well understood for over 40 years. The output of a minimum soft distance decoder is the codeword (in the case of a block code) or code sequence (for a convolutional code) closest in distance to the soft word or sequence received from the channel and input to the decoder. This is soft-in, hard-out (SIHO) decoding, and it can be implemented optimally by applying the Viterbi algorithm (VA) to the code trellis [1], for example. SIHO decoding minimises the average output block or sequence error rate. Conceptually, but not in practice except in the case of a rather simple code, the codeword or code sequence closest in soft distance to the received word is found by determining the soft distances between the received word and all the words or sequences in the code, and then selecting the closest.

In many coding applications, however, it is desirable to have soft estimates of each symbol output from the decoder. This is soft-in, soft-out (SISO) decoding, which minimises the average decoded symbol error rate. Therefore the output of a SISO decoder will not necessarily be a word or sequence in the code. The soft values provide estimates of the confidence of each individual symbol output from the decoder, which can then be used as the soft inputs to another decoder, as is required when decoding concatenated, product or array codes, or for iterative decoding of turbo and low-density parity-check (LDPC) codes. There are several well known ways of obtaining these soft decoder output values, starting with Gallager's 1962 method for LDPC codes [2], which was later re-discovered by Tanner in 1981 [3], and including the BCJR two-way [4], the soft-output VA (SOVA) [5], the turbo iterative [6] and the one-sweep [7] trellis-based algorithms, and the sum-product graph-based algorithms [8, 9, 10, 11], among others. All of these *a posteriori* probability (APP) and maximum *a posteriori* (MAP) algorithms are statistically based, using probabilities or log-likelihood ratios as the decoding metric, and requiring the actual value or a close estimate of the signal-to-noise ratio (SNR) at the decoder input.

In contrast, the new SISO algorithm briefly presented here is non-statistical in nature, and does not require knowledge of the SNR at the decoder input [12-15]. The soft decoded symbols take values computed from the soft values of the received word or sequence (the decoder input) and the parity-check constraints of the code, using (squared) Euclidean distance as the soft decoding metric.

Conceptually, and assuming a binary linear block code, the basic idea of the algorithm is as follows. For a given position in the block (the i^{th}, say) we first compute the soft distances between the soft received (input) word and those codewords in the code which have a zero in the i^{th} position. Then these distance values are combined by taking minus the log of the sum of the anti-logs of the negative values of the squared distances. For example, the combination of squared soft distances A and B is given by:

$$C_0 = -\log_2 (2^{-A} + 2^{-B}).$$

Any number of distances can be combined in this way, using a suitable base for the logarithms and antilogarithms. The above process is then repeated for those codewords which have a one in the i^{th} position, to obtain the combined value C_1. If C_0 is less than C_1 then the symbol in the i^{th} position is decoded as a zero, and vice-versa. The modulus of the difference between C_0 and C_1 indicates the confidence of the decoded (hard) value: high if relatively large and low if not. With suitable scaling to lie within an appropriate range (±1, for example) a single soft value of distance can be output from the decoder. Of course in practice this process is too lengthy except for rather simple codes, so it is much more effective to use one of the trellis or graphical decoding structures and processes referenced above, appropriately amended to fit the new algorithm. In the case of a trellis structure and the two-way BCJR process [4], for example, the antilog-sum combination as above takes place between the soft distance values on the edges entering each node in the trellis.

It is also possible to calculate the values of C as follows:

$$C = \min\{ A, B \} - \log_2 (1 + 2^{-|A - B|}).$$

This expression consists of an approximation and a correction factor. The approximation is easily determined, and the correction factor can be stored in a pre-calculated table of suitable size, thus much simplifying practical implementations of the antilog-sum algorithm [16, 17]. This feature also makes it easy to vary the base of the logarithm, since we have found that base 2 is appropriate for codes with rate ≤ ½ and base 4 for codes with rate > ½.

It is perhaps not surprising that soft distances can be used for SISO decoding of error-correcting codes, since they can serve as representatives or proxies for likelihood ratios [18]. By analogy with the basic *a posteriori* probability (APP) decoding process, the antilogs of A and B can be thought of as pseudo-probabilities and the result of the combination as a pseudo log-likelihood ratio (LLR). Also, the soft distance combination process can be seen to be a generalisation of Hagenauer’s “box-plus” operation [19].

What did surprise us, however, was the excellent performance of the new algorithm. So far, the results of iterative decoding simulations of binary block codes with a range of rates and lengths up to 2560 bits show that the antilog-sum decoding algorithm has a performance in additive white Gaussian noise (AWGN) which is almost identical to that of the most powerful algorithms in current use, such as the sum-product and log-sum-product algorithms. In addition, it seems that the complexity of the antilog-sum algorithm is significantly lower than that of the sum-product and BCJR algorithms. We also have preliminary results for the performances of some of these codes on the Rayleigh fading channel which are similarly close to that of the statistical algorithms.

The discovery of this non-statistical antilog-sum algorithm for SISO decoding of error-correcting codes has resolved a problem which I have contemplated for many years, and it has led to a number of new insights and ideas, which with my colleagues Jorge Castiñeira and Leonardo Arnone we are continuing to explore.

References

[1] A.J. Viterbi: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm; IEEE Trans. Info. Theory, Vol IT-13, No 2, pp 260-9, April 1967.

[2] R.G. Gallager: Low-density parity-check codes; IRE Trans. Info. Theory, Vol IT-8, No 1, pp 21-8, Jan. 1962.

[3] L.M. Tanner: A recursive approach to low complexity codes; IEEE Trans. Info. Theory, Vol IT-27, No 5, pp 533-47, Sept. 1981.

[4] L. Bahl, J. Cocke, F. Jelinek and J. Raviv: Optimal decoding of linear codes for minimising symbol error rate; IEEE Trans. Info. Theory, Vol 42, No 2, pp 284-7, March 1974.

[5] J. Hagenauer and P. Hocher: A Viterbi algorithm with soft-decision outputs and its applications; Proc. Globecom, Dallas, Texas, USA, pp 1680-86, Nov. 1989.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima: Near Shannon limit error-correcting coding and decoding: turbo codes; Proc. IEEE Int. Conf. on Communications, Geneva, Switzerland, pp 1064-70, May 1993.

[7] T. Johansson and K. Zigangirov: A simple one-sweep algorithm for optimum APP symbol decoding of linear block codes; IEEE Trans Info. Theory, Vol IT-44, No 7, pp 3124-9, Nov. 1998.

[8] N. Wiberg, H.-A. Loeliger and R. Kötter: Codes and iterative decoding on general graphs; Euro. Trans. Telecommun, Vol 6, pp 513-26, 1955.

[9] D.J.C. MacKay: Good error-correcting codes based on very sparse matrices; IEEE Trans. Info. Theory, Vol IT-45, No 2, pp 399-431, March 1999.

[10] L.J. Arnone, A. Gayoso, C.M. Gonzalez and J. Castiñeira Moreira: Sum-subtract fixed-point LDPC logarithmic decoder; Proc XI RPIC, Rio Cuarto, Argentina, (1), pp 8-9, September 2005.

[11] J. Castiñeira Moreira and P.G. Farrell: Essentials of Error-Control Coding; Wiley, 2006.

[12] P.G. Farrell: Decoding of error-control codes using soft distance as the metric; Invited Seminar, EE Dept., California Institute of Technology, USA, 5th Dec. 2007.

[13] P.G. Farrell: Decoding of error-control codes with soft distance as the metric; Workshop on Maths Techniques in Coding Theory, Edinburgh, UK, 24th April, 2008.

[14] P.G. Farrell and J. Castiñeira Moreira: Soft-input soft-output Euclidean distance metric iterative decoder for LDPC codes; Argentine Symposium on Computing Technology (AST 2008), Santa Fe, Argentina, 4-12 September, 2008.

[15] J. Castiñeira and P.G. Farrell: Soft-decision SISO decoding of error-control codes; 13th Int. Conf. on Telecommunications (SENACITEL'08), Valdivia, Chile, 12-15 Nov. 2008.

[16] P.G. Farrell, L. Arnone and J. Castiñeira Moreira: Implementación en FPGA de un decodificador LDPC de distancia métrica Euclidiana; XV IBERCHIP Workshop, Buenos Aires, Argentina, 25-27 March 2009.

[17]. L. Arnone, J. Castiñeira Moreira and P.G. Farrell: FPGA implemen-tation of a Euclidean distance metric SISO decoder; submitted to ISCTA'09, Ambleside, 13-17 July, 2009.

[18] T.K. Moon: Error-Correction Coding – Mathematical Methods and Algorithms; Wiley, 2005.

[19] J. Hagenauer, E. Offer and L Papke: Iterative decoding of binary block and convolutional codes; IEEE Trans. Info. Theory, Vol IT-42, No 2, pp 429-45, March 1996.