Reduced-Complexity Soft-In-Soft-Out Decoding of Variable-Length Codes

Claudio Weidmann[1]
ftw. Telecommunications Research Center Vienna
Donau-City-Strasse 1, A-1220 Vienna, Austria
e-mail: claudio.weidmann@ieee.org

Abstract — **We present approaches to complexity reduction for a recently introduced soft-in-soft-out (SISO) decoder for variable-length codes (VLC). One method is based on a new trellis construction that reduces complexity by an order of magnitude, but incurs only a small performance loss in many settings.**

I. Introduction

Several methods to use soft information to decode variable-length codes have been proposed over the last few years. Our starting point is a symbol-by-symbol MAP method, which may be described in a BCJR-type framework [1] or as an instance of Bayesian networks [2]. It can exploit the residual redundancy left in the code sequence when symbols from a (first-order) Markov source are coded independently with a fixed VLC. Additionally, it provides bit-level soft outputs, which may be used to iterate between a SISO channel decoder and a SISO source decoder.

II. Soft-In-Soft-Out VLC Decoding

Consider a sequence of discrete random variables S_1^k emitted by a first-order Markov source with conditional probabilities $P_{S_t|S_{t-1}}(s|s')$ and marginal $P_S(s)$. The sequence is encoded symbol by symbol using a fixed binary VLC and transmitted over a binary-input memoryless channel. Let $\ell(s)$ be the length of the codeword for s. Define the segmentation points $N_t = \sum_{j=1}^{t} \ell(S_j)$, $t = 1, \ldots, k$, such that N_t points to the end of the t-th codeword in the encoded sequence. The key assumption made is that the decoder knows the number of symbols k and the number of bits N_k used to encode them. Then it can build a *segmentation trellis* with states (N_t, S_t) that represents all possible partitions of N_k into k codewords. The BCJR algorithm can then be applied on this trellis to obtain the a posteriori probabilities (APP) $\Pr\{N_t = n, S_t = s \,|\, y_1^{N_k}\}$, which in turn yield the MAP symbol decisions and the bit-level soft outputs. The main problem is that the size of this trellis grows as $O(k^2)$.

III. Complexity reduction

A straightforward approach to reduce the complexity is to prune the trellis, e.g. by limiting the bit counter N_t to a window $[n_{min}(t), n_{max}(t)]$. That method gives a fair performance-complexity tradeoff, but the window needs to be quite wide (multiples of maximum codeword length), since the typical VLC decoding error behavior involves desynchronizations of the symbol clock t.

On the other hand, after an error, VLCs tend to resynchronize quickly to the correct codeword boundaries, regardless of symbol clock desynchronizations. This suggests a radically simplified trellis that no longer keeps track of the symbol clock, but only of the codewords ending in bit n. Figure 1 shows an example for a VLC with $\ell(a) = 1$ and $\ell(b) = \ell(c) = 2$.

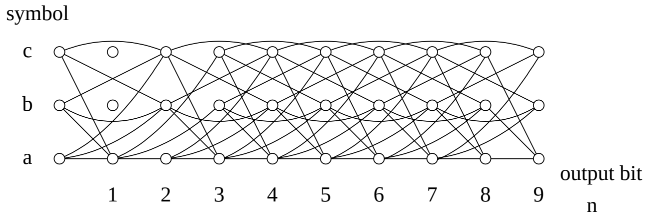

Figure 1: Reduced segmentation trellis

Let $\{S_n = s\}$ denote the event that the codeword for symbol s was sent in bit positions $n - \ell(s) + 1, \ldots, n$. The BCJR to obtain the APP $\Pr\{S_n = s \,|\, y_1^{N_k}\}$ uses the variables

$$\alpha_n(s) = \Pr\{S_n = s, y_1^n\},$$
$$\beta_n(s) = \Pr\{y_{n+1}^{N_k} \,|\, S_n = s\},$$
$$\gamma_n(s'; s) = \Pr\{S_n = s, y_{n-\ell(s)+1}^{n} \,|\, S_{n-\ell(s)} = s'\}$$

and the recursions

$$\alpha_n(s) = \textstyle\sum_{s'} \alpha_{n-\ell(s)}(s')\gamma_n(s'; s), \qquad n = 1, \ldots, N_k.$$
$$\beta_n(s) = \textstyle\sum_{s'} \beta_{n+\ell(s')}(s')\gamma_{n+\ell(s')}(s; s'), \quad n = N_k - 1, \ldots, 1,$$

initialized with $\alpha_0(s) = P(s)$ and $\beta_{N_k}(s) = 1$, respectively. After computing the αs and βs, the decoder proceeds as follows:

1. Compute the APP $\Pr\{S_n = s \,|\, y_1^{N_k}\} \propto \lambda_n(s) = \Pr\{S_n = s, y_1^{N_k}\} = \alpha_n(s)\beta_n(s)$ and the probability that a symbol ends at bit n (jointly with $y_1^{N_k}$), $\chi(n) = \sum_s \lambda_n(s)$.
2. Find the k largest $\chi(n)$, say $\chi_{\pi(1)}, \ldots, \chi_{\pi(k)}$ with $0 \le \pi(1) \le \ldots \le \pi(k) \le N_k$. Output $\hat{s}_t = \arg\max_s \lambda_{\pi(t)}(s)$ as the t-th decoded symbol.

This algorithm first finds the k most likely segmentation points and then outputs the most likely symbols at those points. If implemented via a sort, the computational complexity of step 2 is $O(k \log k)$, which is essentially linear for practical block sizes. It turns out that the symbol error performance on this linear-size reduced trellis is close to that of the quadratic-size trellis, especially for sources with some correlation.

References

[1] R. Bauer and J. Hagenauer. "Symbol-by-symbol MAP decoding of variable length codes," In *3rd ITG Conf. on Source and Channel Coding*, pp. 111–116, Munich, Germany, Jan. 2000.

[2] A. Guyader, E. Fabre, C. Guillemot, and M. Robert. "Joint source-channel turbo decoding of entropy-coded sources," *IEEE JSAC*, vol. 19, pp. 1680–1696, Sept. 2001.

[1] This work was supported by the French ministry of research under contract COSOCATI while the author was with INRIA Rennes.