Bit Error Rate Estimation for Turbo Decoding

Nick Letzepis and Alex Grant

Abstract

We propose a method for on-line estimation of Bit Error Rate during turbo decoding. We model the log-likelihood ratios as a mixture of two Gaussian random variables and derive estimators for the mean and variance of these distributions, which can be used to estimate BER.

I. INTRODUCTION

Turbo codes [1] exhibit coding gains remarkably close to the Shannon Limit. We give a brief overview of the encoder and decoder, mainly to fix notation. As shown in Figure 1(a), a turbo code is the parallel concatenation of two Recursive Systematic Convolutional (RSC) codes via an interleaver Π. By \( b_k \) we denote the information bit at time \( k \). The sequences \( S, P \) and \( Q \) correspond to the systematic and two parity sequences. The iterative decoder, Figure 1(b), employs an A-Posteriori Probability (APP) decoder for each constituent code. Log-likelihood ratio (LLR) information for the systematic \( \Lambda_S \) and parity bits \( \Lambda_P, \Lambda_Q \) are inputs to the APP decoders along with a-priori information \( \Lambda_A \). Each APP decoder produces extrinsic information \( \Lambda_E \) which is essentially independent of the received systematic sequence [2].

Extrinsic information exchange is critical in understanding the convergence behavior. In particular, Gaussian LLR models have been used extensively. In [3], the Gaussian assumption is used to characterize convergence behavior in terms of the extrinsic Signal-to-Noise ratio (SNR). Using EXIT charts [4], one can predict. The BER at a given iteration can be predicted, by imposing a Gaussian assumption. Density Evolution [5] uses a symmetry assumption in addition to the Gaussian assumption to give more accurate results.

In Section II we model LLRs as a mixture of two Gaussian random variables with equal variances and equal means in magnitude but opposite in sign. In Section III we derive maximum likelihood (ML) based estimators for the parameters of the LLR distribution. Using these expressions in Section IV we explore methods for estimating BER without knowledge of the original transmitted data.

The authors are with the Institute for Telecommunications Research, University of South Australia. This work was supported by the Australian Government under ARC Grant LP0219304.
II. LLR DISTRIBUTION MODEL

Denote the LLR for bit \(k\) as \(\lambda(k) = \log(\Pr(b_k = 1) / \Pr(b_k = 0))\). Subscripts \(S\), \(A\) and \(E\) respectively denote systematic, prior and extrinsic LLRS. Let \(\Lambda = \{\lambda(0), \ldots, \lambda(N - 1)\}\) be a sequence of \(N\) LLRs. From Figure 1(b), the decoder output is \(\lambda_D(k) = \lambda_S(k) + \lambda_A(k) + \lambda_E(k)\) [1]. Assume that, conditioned on the systematic bit \(b_k\) at time \(k\), \(\lambda_S(k), \lambda_A(k)\) and \(\lambda_E(k)\) are independent Gaussian random variables. From this assumption we have \(f_{\lambda_S(k)}(x|b_k = 1) = \mathcal{N}(\mu_A, \sigma_A^2)\), \(f_{\lambda_A(k)}(x|b_k = 1) = \mathcal{N}(\mu_S, \sigma_S^2)\) and \(f_{\lambda_E(k)}(x|b_k = 1) = \mathcal{N}(\mu_E, \sigma_E^2)\), where \(\mathcal{N}(\mu, \sigma^2)\) is the Gaussian density with mean \(\mu\) and variance \(\sigma^2\). Hence \(f_{\lambda_D(k)}(x|b_k = 1) = \mathcal{N}(\mu_D, \sigma_D^2)\) with \(\mu_D = \mu_S + \mu_A + \mu_E\) and \(\sigma_D^2 = \sigma_S^2 + \sigma_A^2 + \sigma_E^2\).

We will focus on \(\lambda_D\), but it should be noted that the following expressions are valid for \(\lambda_A\), \(\lambda_E\) and \(\lambda_S\). For clarity we drop the subscript denoting the LLR type. By assumption, \(f_{\lambda(k)}(x|b_k = 1) = \mathcal{N}(\mu, \sigma^2)\) and \(f_{\lambda(k)}(x|b_k = 0) = \mathcal{N}(-\mu, \sigma^2)\). Further assuming equi-probable bits,

\[
f_{\lambda(k)}(x) = \frac{1}{2} \mathcal{N}(\mu, \sigma^2) + \frac{1}{2} \mathcal{N}(-\mu, \sigma^2) \tag{1}
\]

We must estimate \(\mu\) and \(\sigma\) without knowledge of \(b_k\). This is parameter estimation for a Gaussian mixture [6]. We are interested in sub-populations with equal variance and means differing only in sign.

III. MAXIMUM LIKELIHOOD BASED METHODS

We now find estimators for \(\mu\) and \(\sigma\) of (1) using an approximate maximum likelihood approach. We begin by assuming there is no relationship between the parameters. Then we impose the symmetry \(\sigma^2 = 2\mu\).

For large interleavers it is reasonable to assume that \(\lambda(k)\) are iid. From (1), the LLR of the sequence \(\Lambda\) is

\[
\ln f_{\lambda}(\lambda; \mu; \sigma^2) = -N \ln \left(2\sqrt{2\pi\sigma^2}\right) - \frac{N\mu^2}{2\sigma^2} - \sum_{k=0}^{N-1} \frac{\lambda^2(k)}{2\sigma^2} + \sum_{k=0}^{N-1} \ln \left(2 \cosh \left(\frac{\lambda(k)\mu}{\sigma^2}\right)\right) \tag{2}
\]

Taking the partial derivative of (2) with respect to \((\text{wrt})\) \(\mu\) we have

\[
\frac{\partial}{\partial \mu} \ln f_{\lambda}(\lambda; \mu; \sigma^2) = \frac{N\mu}{\sigma^2} + \sum_{k=0}^{N-1} \frac{\lambda(k)}{\sigma^2} \tanh \left(\frac{\lambda(k)\mu}{\sigma^2}\right) \tag{3}
\]

Now using the large \(x\) approximation \(x \tanh(x) \approx |x|\), setting (3) to zero and solving for \(\mu \geq 0\) results in

\[
\hat{\mu} = \hat{E}[|\Lambda|] = \frac{1}{N} \sum_{k=0}^{N-1} |\lambda(k)| \tag{4}
\]

Partial differentiating (2) wrt \(\nu = \sigma^2\) we have

\[
\frac{\partial}{\partial \nu} \ln f_{\lambda}(\lambda; \mu; \nu) = \frac{1}{2\nu^2} \sum_{k=0}^{N-1} \lambda^2(k) - \frac{N\mu^2}{2\nu} + \frac{N\mu^2}{2\nu^2} - \sum_{k=0}^{N-1} \frac{\lambda(k)\mu}{\nu^2} \tanh \left(\frac{\lambda(k)\mu}{\nu}\right) \tag{5}
\]

Using the same approximations as before, \(\hat{\sigma}^2 = \hat{E}[|\Lambda|^2] - 2\hat{\mu}\hat{E}[|\Lambda|] + \hat{\mu}^2\) and after substitution of (4),

\[
\hat{\sigma}^2 = \hat{E}[|\Lambda|^2] - \hat{\mu}^2, \text{ where } \hat{E}[|\Lambda|^2] = \sum_{k=0}^{N-1} |\lambda(k)|^2/N \tag{6}
\]

Note that (4) has already been (somewhat arbitrarily) used for convergence analysis (without the above motivation or derivations). In [7] a stopping criterion is given, comparing (4) to a threshold (determined
through trial and error). In [8], they call (4) the mean reliability and use it for a stopping, when the change in (4) is less than some threshold (again found by trial and error). In Section IV we propose using estimates of \( \mu \) to estimate the BER in an on-line fashion. This allows us to set meaningful stopping thresholds.

We now derive a true ML estimate (MLE) for \( \mu \) assuming \( \sigma^2 = 2\mu \) an assumption also used in [4, 5].

**Theorem 1 (MLE For \( \mu \) under a symmetry assumption).** Suppose that the LLRs are iid according to (1) and \( \sigma^2 = 2\mu \). Then the MLE of \( \mu \) is

\[
\hat{\mu} = -1 + \sqrt{1 + \hat{E} [\Lambda^2]}, \text{where } \hat{E} [\Lambda^2] = \frac{1}{N} \sum_{k=0}^{N-1} \lambda(k)^2
\]

Proof. Substituting \( \sigma^2 = 2\mu \) into (2), differentiating wrt \( \mu \) and equating to zero yields \( \mu^2 + 2\mu - \hat{E} [\Lambda^2] = 0 \). Since \( \mu \geq 0 \), the required solution is \( \hat{\mu} = -1 + \sqrt{1 + \hat{E} [\Lambda^2]} \).

By taking the second partial derivative of the log-likelihood function wrt \( \mu \) we can determine the following Cramer-Rao Bound (CRB), which is an upper bound for the mean-squared error performance of any estimator for \( \mu \) (under the symmetric-Gaussian assumption).

\[
\text{var}[\hat{\mu}] \geq \frac{2\mu^2}{N(1 + \mu)}.
\]

The effectiveness of the two approaches has been investigated using Monte-Carlo simulations to measure the actual histogram of the extrinsic LLRs and then comparing this with the Gaussian PDF generated from the parameter estimates for \( \mu \) and \( \sigma \). The simulation model involved turbo encoding a 216 bit block of randomly generated binary data using a rate 1/2 code with constituent RSC codes \((G_r, G) = (7, 5)\), \((G_r \text{ is the feedback polynomial})\), transmitting using BPSK modulation at \( E_b/N_0 = 1.5 \text{ dB} \) and turbo decoding the received noisy encoded symbols. The histogram and parameter estimates were averaged over 1000 trials.

Figure 2 compares the histogram of the extrinsic LLRs output from the first constituent decoder (solid) with the Gaussian PDF when estimating \( \mu \) and \( \sigma \) independently (dashed) and when enforcing symmetry (dot-dashed). Estimating \( \mu \) and \( \sigma \) independently appears to give a closer approximation, however we found that enforcing symmetry gives a better approximation for the tails which is more important for BER estimation.

We suspect that an \( \alpha \)-stable distribution [9] may be a more appropriate model, reflecting the heavy tail and skewness. The stable law contains the Gaussian distribution as a limiting case. The central limit theorem states that if the sum of iid random variables converges in distribution, the limit distribution must belong to the family of stable laws. For large block lengths and after a large number of iterations, turbo decoding can be thought of as the accumulation of a large number of iid effects giving some justification to modelling the LLRs as an \( \alpha \)-stable. Preliminary experiments indicate that the \( \alpha \)-stable distribution indeed provides a
better fit. However, parameter estimation is difficult, since the distribution can in general only be described by its characteristic equation [10]. Online BER estimation based on an $\alpha$-stable model would most likely be impractical. It would none the less be interesting to develop such a model, which we leave for future work.

IV. BER ESTIMATION

A method of predicting the bit error probability $P_b$ from an EXIT chart after an arbitrary number of iterations is given in [4]. This involved performing the inverse of the transfer characteristic of the EXIT chart to determine $\sigma_D$ (assuming $\Lambda_D$ is Gaussian). Thus $P_b$ can be predicted using

$$\hat{P}_b = \frac{1}{2} \text{erfc} \left( \frac{\sigma_D}{2\sqrt{2}} \right)$$

where $\text{erfc}$ is the complimentary error function. This provides reliable BER predictions down to $10^{-3}$, but is not suitable for determining BER floors. This technique is not suitable for on-line BER estimation since it requires knowledge of the original transmitted data bits.

Three BER estimation methods have been presented in [11]. The second of these methods, namely

$$\hat{P}_b = \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{1 + e^{[\Lambda_D[i]]}}$$

was the best and we shall label this as HLS [11].

We propose using (7) to estimate $\mu_D$, and substituting (9) to find the following “Gaussian Assumption Model Maximum Likelihood” (GAMML) estimate,

$$\hat{P}_b = \frac{1}{2} \text{erfc} \left( \sqrt{|\mu_D|}/2 \right)$$

Since $\hat{P}_b$ is a function of $\hat{\mu}$ it’s CRB is

$$\text{var}[\hat{P}_b] \geq \frac{\mu}{1 + \mu} \frac{\exp(-\mu/2)}{8\pi N}. \quad (11)$$

To compare these BER estimators we considered three cases. First we actually generated LLRs iid according to a Gaussian mixture described by (1) with $\sigma^2 = 2\mu$. Next, we considered the LLRs resulting from using a convolutional code. Finally, we considered the LLRs measured from a turbo decoder at each iteration.

Figure 3(a) shows the performance of the BER estimators given that the LLRs are iid according to a Gaussian mixture. The normalized standard deviation of the estimates was calculated by dividing the standard deviation of the estimator outputs by $P_b = \frac{1}{2} \text{erfc} \left( \sqrt{\mu}/2 \right)$ where $\mu$ is determined from $E_b/N_0$. It can be seen that HLS is quite some distance away from the CRB, which is achieved by GAMML. Both estimators and the CRB degrade linearly with SNR, due to the assumption $\sigma^2 = 2\mu$.

Figure 3(b) shows the normalized standard deviation of the estimators after APP decoding of a RSC code with $(G_r, G) = (7, 5)$ and $N = 2^{16}$ bits. The LLRs is not entirely Gaussian, causing two affects. Firstly, GAMML no longer achieves the CRB. Secondly the estimator is no longer unbiased. GAMML tends to underestimate the BER at low $E_b/N_0$ and overestimate it for high $E_b/N_0$. This observation is consistent with convergence analysis results in [5]. HLS always underestimates the BER. To calculate the CRB for BER estimation from the output of the APP decoder we work backwards to determine $\mu$ assuming that the LLRs are iid Gaussian, ie: $\mu = \left(2\text{erf}^{-1}(1 - 2P_b)\right)^2$. Then (11) is used to determine the CRB.

Figure 4 compares the average BER estimate compared to the measure BER for a turbo code at each decoder iteration for different constituent codes and block lengths. The estimates were averaged over 60000 trials for $N = 2^{10}$ bits and 1000 trials for $N = 2^{16}$ bits. For low levels of $E_b/N_0$ despite a small bias,
the mean of the BER estimate tends to follow the actual BER. For large block lengths and high $E_b/N_0$, the average BER from the GAMML estimator tends not to follow the steepness the actual BER curve. Since the GAMML estimator is based on a Gaussian assumption, its performance depends on how well the LLRs fit this model. This can be seen in Figure 4, which shows that the average bias of the two estimators varies between different codes. It can also be seen that for small block lengths the GAMML estimator always under estimates BER and for large block lengths the bias swaps from an under estimate to an over estimate and then back to an under estimate.

Figure 5 shows a histogram of the BER estimates for each estimator at an $E_b/N_0 = 2$ dB with a packet length of $2^{10}$ bits at iterations 1, 2, 3, 5, 18. We note that the distribution of HLS estimates tends to follow the BER for individual packets. For packets with no errors the HLS estimator will output a very low BER and vice-versa for packets with many errors. The GAMML estimator gives values distributed nearer the average
BER plus some bias due to the skewness and heavy tail of the LLRs actual distribution.

![Histogram of BER estimates from the turbo decoder at iterations 1, 2, 3, 5 and 18.](image)

Fig. 5. Histogram of BER estimates from the turbo decoder at iterations 1, 2, 3, 5 and 18.

V. CONCLUSION

We have derived estimators for the parameters of the LLR distribution by modelling LLRs as a Gaussian mixture. By estimating the mean and variance of this mixture the corresponding BER can be estimated without knowledge of the original transmitted data (the GAMML estimate).

We compared the GAMML estimator with the HLS estimator from [11]. The performance of GAMML depends on how well the LLRs fit the Gaussian model. For LLRs that perfectly fit the Gaussian mixture model (with symmetry) the GAMML estimator achieves the CRB (it is the ML estimator in this case). When APP decoding of a convolutional code, the LLRs become less Gaussian and the GAMML estimator becomes biased and strays away from the CRB but still out-performs HLS. For turbo decoding, the Gaussianity of the LLRs varies between iterations and it becomes difficult to compare the performance of the two estimators. The GAMML estimator tends to have estimates closer to the average BER but suffers from a larger bias at the error floor region. The HLS estimator tends to have a higher variance but less average bias at the error floor region.

In terms of complexity, one could argue that the GAMML estimator would require less operations. It requires the calculation of the mean square of the LLR values, then a single square root function and a single erfc, for which (reduced complexity) approximations are available. The HLS estimator requires exponent and division operations for each LLR value which is very computationally expensive.

REFERENCES