References

1 LUCYSZYN, S., and ROBERTSON, I.D.: 'Synthesis techniques for high performance octave bandwidth 180° analogue phase shifters', *IEEE Trans. Microw. Theory Tech.*, 1992, **40**, (4)

2 AUST, M., WANG, H., CARANDANG, R., TAN, K., CHEN, C.H., TRINH, T., ESFANDIARI, R., and YEN, H.C.: 'GaAs monolithic components development for Q-band phased array application'. IEEE MTT-S Int. Microwave Symp., Albuquerque, New Mexico, USA, June 1992, pp. 703–706

3 MARUHASHI, K., MIZUTANI, H., and OHATA, K.: 'Design and performance of a Ka-band monolithic phase shifter utilizing nonresonant FET switches', *IEEE Trans. Microw. Theory Tech.*, 2000, **48**, (8)

Density evolution for low-density parity-check codes under Max-Log-MAP decoding

X. Wei and A.N. Akansu

A density evolution procedure for low-density parity-check (LDPC) codes under Max-Log-MAP decoding is presented. Using this technique, the precise convergence threshold for LDPC code could be easily derived.

Introduction: Much progress has been made recently on estimating the convergence threshold of turbo codes and low-density parity-check (LDPC) codes [1], assuming the underlying graphical model to be cycle-free. However, these works have been in the context of Log-MAP decoding. A Max-Log-MAP algorithm, with reduced complexity and no requirement on channel SNR estimation, has not been considered. In this Letter, a density evolution procedure under Max-Log-MAP decoding is developed. Using this procedure, convergence thresholds at a rate of half regular LDPC codes for binary-input AWGN channels are derived.

Max-Log-MAP decoding for LDPC codes: Gallager invented the LDPC code as well as an efficient probability decoding scheme [2], which was later elaborated in [3] and termed a sum-product algorithm. It is equivalent to Log-MAP decoding except that the former works on probabilities while the latter works on log probability ratios. Max-Log-MAP decoding approximates the optimal Log-MAP algorithm by substituting each 'log-exponential'-operation with 'max'-operation,

$$\ln(e^x + e^y) = \max(x, y) + \ln\left(1 + e^{-|x-y|}\right) \simeq \max(x, y) \tag{1}$$

Let $\mathbf{x} = [x_n]$ and $\mathbf{H} = [H_{mn}]$ be the codeword and the parity-check matrix, respectively, of an LDPC code, so that $\mathbf{Hx} = 0$. Denote the set of bits n that participate in check m by $N(m) = \{n: H_{mn} = 1\}$. Similarly, the set of checks in which bit n participates is denoted as $M(n) = \{m: H_{mn} = 1\}$. We denote a set $N(m)$ with bit n excluded by $N(m)\backslash n$, and a set $M(n)$ with parity check m excluded by $M(n)\backslash m$. ξ_{mn} is defined as extrinsic information extracted from check node m and to be passed to variable node n. ζ_{mn} is the message from variable node m to check node n. The Max-Log-MAP decoding algorithm is described as follows:

Initialisation: $\zeta_{mn}^{(0)} = L_n$.

Horizontal pass:

$$\xi_{mn}^{(i)} = \min_{n' \in N(m)\backslash n} \left|\zeta_{mn'}^{(i-1)}\right| \prod_{n' \in N(m)\backslash n} \operatorname{sgn}\left(\zeta_{mn'}^{(i-1)}\right) \tag{2}$$

Vertical pass:

$$\zeta_{mn}^{(i)} = L_n + \sum_{m' \in M(n)\backslash m} \xi_{m'n}^{(i)} \tag{3}$$

Make decisions:

$$D^{(i)}(\hat{x}_n) = L_n + \sum_{m \in M(n)} \xi_{mn}^{(i)} \tag{4}$$

The decision so far is given by $\hat{\mathbf{x}} = [\hat{x}_n]$ such that $\hat{x}_n = 0$ if $D^{(i)}(\hat{x}_n) > 0$; otherwise $\hat{x}_n = 0$. If $\hat{\mathbf{x}}$ is a valid codeword so that $\mathbf{H}\hat{\mathbf{x}} = 0$, then the algorithm halts; otherwise, repeat the horizontal and vertical pass until some maximal number of iterations is reached without a valid decoding.

In Log-MAP decoding, the horizontal pass is set as

$$\xi_{mn}^{(i)} = 2\tanh^{-1}\left(\prod_{n' \in N(m)\backslash n} \tanh\left(\zeta_{mn'}^{(i-1)}/2\right)\right) \tag{5}$$

Max-Log-MAP decoding not only reduces the computation burden in the horizontal pass, but also eliminates the requirement of SNR estimation. In Log-MAP decoding, L_n is set to $L_{ch}y_n$, which means both channel value and channel parameter are required. In Max-Log-MAP decoding, L_n is simply set to channel value y_n.

Density evolution procedure: Given the PDF of initial values $\zeta^{(0)}$ as $P^{(0)}$, and the specific decoding algorithm, theoretically we should be able to compute the PDF of $\xi^{(i)}$ and $\zeta^{(i)}$ for any $i \geq 1$, which are denoted as $Q^{(i)}$ and $P^{(i)}$, respectively. Through observing the evolution of $P^{(i)}$ and $Q^{(i)}$, we would be able to ascertain whether the fraction of an incorrect message approaches zero or not as the number of iterations increases. Assuming the underlying graphical model to be cycle-free, $Q^{(i)}$ and $P^{(i)}$ are numerically computable. For vertical pass, the PDF of ζ is obtained by convolutions, for the random variables on the right side of eqn. 3 are pair-wise independent,

$$P^{(i)} = f_y \otimes (Q^{(i)})^{\otimes d_v - 1} \tag{6}$$

where f_y is the PDF of channel values and d_v is the degree of variable node. For horizontal pass, we take the conventional approach. View the random variable ξ as a function of random variable vector $\mathbf{X}$: $\xi = \min_{k=1}^{d_c-1}|X_k| \prod_{k=1}^{d_c-1} \operatorname{sgn}(X_k)$, the cumulative distribution function (CDF) of ξ is found by integrating the joint PDF of vector $\mathbf{X}$ over the region R corresponding to the event $\{\min_{k=1}^{d_c-1}|x_k| \prod_{k=1}^{d_c-1} \operatorname{sgn}(x_k) \leq \xi\}$,

$$\begin{aligned} F_\xi(\xi) &= \Pr\left[\min_{x=1}^{d_c-1}|x_k| \prod_{k=1}^{d_c-1} \operatorname{sgn}(x_k) \leq \xi\right] \\ &= \int \cdots \int_R P(x_1) \cdots P(x_{d_c-1}) dx_1 \cdots dx_{d_c-1} \end{aligned} \tag{7}$$

where d_c is the degree of check node. The PDF of ξ is found by taking the derivative of $F_\xi(\xi)$. It is convenient to define $\phi_+^{(i)}(x_1) = \int_{|x_1|}^{+\infty} P^{(i)}(x)dx$ and $\phi_-^{(i)}(x_1) = \int_{-\infty}^{-|x_1|} P^{(i)}(x)dx$. Then, for $\xi \leq 0$, the CDF is

$$\begin{aligned} &F_\xi^{(i+1)}(\xi) = \\ &(d_c - 1)\left[\int_{-\infty}^{\xi} P^{(i)}(x) \sum_{\substack{k=0 \\ \text{even}}}^{d_c-2} \left(\phi_+^{(i)}(x)\right)^{d_c-2-k} \left(\phi_-^{(i)}(x)\right)^k dx \right. \\ &\left. + \int_{-\xi}^{+\infty} P^{(i)}(x) \sum_{\substack{k=0 \\ \text{odd}}}^{d_c-2} \left(\phi_+^{(i)}(x)\right)^{d_c-2-k} \left(\phi_-^{(i)}(x)\right)^k dx\right] \end{aligned} \tag{8}$$

For $\xi > 0$, the CDF is

$$\begin{aligned} &F_\xi^{(i+1)}(\xi) = \\ &1 - (d_c - 1)\left[\int_{-\infty}^{-\xi} P^{(i)}(x) \sum_{\substack{k=0 \\ \text{odd}}}^{d_c-2} \left(\phi_+^{(i)}(x)\right)^{d_c-2-k} \left(\phi_-^{(i)}(x)\right)^k dx \right. \\ &\left. + \int_{\xi}^{+\infty} P^{(i)}(x) \sum_{\substack{k=0 \\ \text{even}}}^{d_c-2} \left(\phi_+^{(i)}(x)\right)^{d_c-2-k} \left(\phi_-^{(i)}(x)\right)^k dx\right] \end{aligned} \tag{9}$$

Take the derivative of $F_\xi(\xi)$ with respect to ξ and the PDF $Q^{(i+1)}$ is finally found as

$$\begin{aligned} &Q^{(i+1)}(\xi) = \\ &\frac{d_c - 1}{2}\left[\left(P^{(i)}(\xi) + P^{(i)}(-\xi)\right)\left(\phi_+^{(i)}(\xi) + \phi_-^{(i)}(\xi)\right)^{d_c-2} \right. \\ &\left. + \left(P^{(i)}(\xi) - P^{(i)}(-\xi)\right)\left(\phi_+^{(i)}(\xi) - \phi_-^{(i)}(\xi)\right)^{d_c-2}\right] \end{aligned} \tag{10}$$

Table 1: Thresholds for binary-input AWGN channel under Max-Log-MAP and Log-MAP decoding

j	k	Rate	Thresholds, dB	
			Max-Log-MAP	Log-MAP
3	6	0.5	1.7	1.11
4	8	0.5	2.5	1.62
5	10	0.5	3.1	2.04

Numerical results and conclusions: Convergence thresholds at a rate of one half regular LDPC codes under Max-Log-MAP decoding are given in Table 1, and contrasted with those under Log-MAP decoding derived in [1]. For (3, 6) regular LDPC code, the Max-Log-MAP algorithm yields a threshold 0.6 dB higher than the Log-MAP decoding. This means a performance loss of 0.6 dB due to the lack of channel parameters. With some minor modifications, the above density evolution procedure could easily be applied to irregular LDPC codes.

© IEE 2001
8 June 2001
Electronics Letters Online No: 20010755
DOI: 10.1049/el:20010755

X. Wei and A.N. Akansu (*ECE Department, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA*)
E-mail: xxw3114@oak.njit.edu

References

1 RICHARDSON, T., and URBANKE, R.: 'The capacity of low-density parity-check codes under message-passing decoding', *IEEE Trans. Inf. Theory*, 2001, **47**, (2), pp. 599–618
2 GALLAGER, R.G.: 'Low-density parity-check codes' (MIT Press, 1963)
3 MACKAY, D.J.C.: 'Good error-correcting codes based on very sparse matrices', *IEEE Trans. Inf. Theory*, 1999, **45**, (2), pp. 399–431

One-pass training of optimal architecture auto-associative neural network for detecting ectopic beats

G. Clifford, L. Tarassenko and N. Townsend

The authors have previously described a method for ectopic beat detection in the electrocardiogram using an auto-associative neural network. Here they present a method that utilises principal component analysis to optimise the complexity of the neural network and uses singular value decomposition to determine the initial values for the weights.

Introduction: Effective cardiology requires the identification of significant morphology variations in a patient's electrocardiogram (ECG), a record of the mV fluctuations of the heart's electropotential. Abnormal beat identification is required for heart rate variability calculations and warnings of the onset of potentially fatal arrhythmias. In an earlier paper [1] we described a method for training an auto-associative multilayer perceptron (AAMLP) to perform QRS detection of normal beats and rejection of ectopic beats [1]. It is trained to reconstruct a *normal* waveform in the ECG (with the main features labelled P-QRS-T as in Fig. 2), centred on the R-peak as accurately as possible for that particular subject.

Once the AAMLP has been trained, new ECG data is passed, sample by sample, across the input nodes. The trained weights of the AAMLP then perform the auto-associative mapping of this data and produce an output pattern. When the input data is centred on the R-peak, the AAMLP will reconstruct the QRS complex accurately. Otherwise the reconstruction will be poor since the mapping was learned for R-peak centred waveforms only.

Since the QRS complexes are easily located by their R-peaks [2], which occur approximately half way between the start of the P-wave and the end of the T-wave, a 0.5 s window centred on the R-peak effectively segments each heart beat. To reduce the input/output dimensionality for the auto-associative network and avoid learning fine (and potentially irrelevant) detail while still preserving the main features, the P-QRS-T waveform is down-sampled from 256 to 64 Hz. A 0.5 s window then corresponds to 32 input and output nodes.

However, neural network training can be time consuming and the association for the advancement of medical instrumentation (AAMI) requires that any on-line ECG algorithm must be trained within the first five minutes [3]. There is therefore a trade-off between the number of training patterns that can be collected and the time required for training the AAMLP: the smaller the number of training vectors, the greater the number of iterations through the training set required. Even using a powerful modern PC it is difficult to collect data and train within five minutes. We therefore use singular value decomposition (SVD) to calculate the weights in one pass.

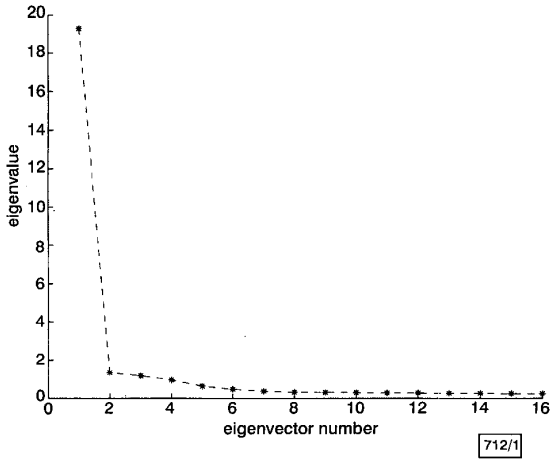

Fig. 1 *Size-ranked set of eigenvectors from SVD decomposition of training set composed of approximately 200 P-QRS-T waveforms*

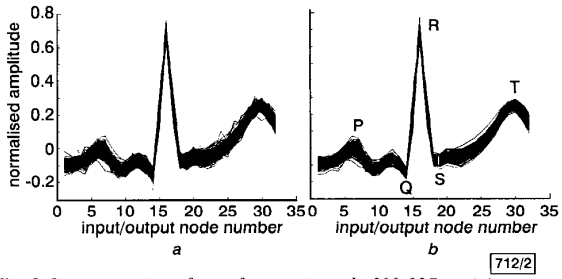

Fig. 2 *Reconstruction of set of approximately 200 32D training vectors using the eigenvectors with eight largest eigenvalues derived from SVD*

P, *Q*, *R*, *S*, and *T* refer to clinical labels [1]
a Training vectors
b Reconstruction

Method: A typical P-QRS-T segment is composed of approximately 10 major turning points. Individual nodes in an AAMLP do not encode individual features, but this number can be used for order of magnitude calculations [4]. The AAMLP encodes the variance in the training set in the same manner as principal component analysis (PCA), projecting the variance onto the same number of orthogonal axes as there are hidden units [4]. A more rigorous method for determining the number of hidden units is to perform PCA on the training set, rank the components in order of magnitude and identify the knee of the curve [5]. Fig. 1 shows a knee at two components, with significant eigenvectors up to eight components. Beyond this any further components contribute very little further information, as they encode noise. In tests on the MIT-BIH database [6] and data collected from a local hospital we found that there was a high degree of inter-patient similarity between singular spectra of training sets, with eight eigen-components being the upper limit of the boundary between signal and noise. The number of hidden units was therefore set to a value of eight. Fig. 2 shows how a typical training set (Fig. 2*a*) is reconstructed (Fig. 2*b*) by the first eight principal components.

For a two layer AAMLP with linear hidden units and a sum-of-squares error function, the weights can be calculated from the