EM Algorithm for DMC Channel Estimation in NAND Flash Memory

Warangrat Wiriya and Brian M. Kurkoski
Japan Advanced Institute of Science and Technology, Ishikawa, Japan
{warangrat.wiriya, kurkoski}@jaist.ac.jp

***Abstract*—The data read system for NAND flash memory can be modeled by a DMC channel with unknown channel transition probabilities. However, LDPC decoders need a channel estimate, and incorrect channel estimation degrades the performance of LDPC decoder. This abstract proposes using the EM algorithm to estimate channel transition probabilities, needed to compute LLRs for the LDPC decoders. At word-error rate 10^-5, the performance of the EM system was only 0.02 dB loss compared to the system that knows the channel exactly.**

***Index Terms*—expectation maximization algorithm, NAND flash memory, binary asymmetric channel, discrete memoryless channel.**

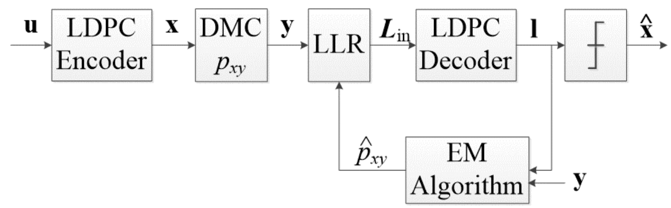

Fig. 1. Block diagram for an LDPC-coded NAND flash memory system. The decoder uses the EM algorithm in turbo equalization-style setting.

I. Introduction

NAND flash memory systems read data by setting a threshold to obtain binary channel outputs, but the allowed number of reads is limited. Due to manufacturing variations, device aging and high storage temperature, the true underlying channel parameters may be difficult to model accurately. Thus, the channel can be approximated by a discrete memoryless channel (DMC) with unknown channel transition probabilities and a small number of outputs. Low-density parity-check (LDPC) decoders require a channel estimate, and incorrect channel estimation degrades the word-error rate (WER) of the LDPC decoder.

In this abstract, the expectation maximization (EM) algorithm is used to estimate unknown DMC transition probabilities. In general, the EM algorithm estimates the parameters of a statistical model. The LDPC decoder plays a key role, providing a priori estimates of the code bits, required by the EM algorithm.

The only previous work we are aware of applying the EM algorithm to DMC estimation is by Boutros et al. [1]. This work also includes an LDPC code to aid estimation. However, only the symmetrical errors-and-erasures channel with three outputs was considered, whereas our aim is to consider general DMC channels. [2] proposed alternative methods using ML, MAP or MSE for estimating the flash memory error model. However, those metrics are calculated based on the number of bits that were read from the same voltage bin and participate in unsatisfied parity check equations, whilst the EM algorithm does not need above information. Recurrent neural networks (RNNs) can also be applied to this problem; the method proposed in [3] is targeted at NAND flash memories, but requires a large amount of training data, whereas the EM algorithm requires no training data.

This work has been partly executed with the support of Kioxia Corporation (formerly Toshiba Memory Corporation). This work was supported by a JAIST Off-Campus Research Grant.

II. EM algorithm

Consider a statistical model which produces an observed sequence $\mathbf{y}$, and has an unobserved sequence $\mathbf{x}$. These are generated by a set of unknown parameters, represented by the vector $\mathbf{p}$. Maximum likelihood estimation $\mathbf{p}^* = \arg\max_{\mathbf{p}} \log \Pr[\mathbf{y}|\mathbf{p}]$ is usually computationally intractable.

The EM algorithm is an iterative method to estimate the unknown channel parameters $\mathbf{p}$. The following is the general EM algorithm [4, p. 441]:

1) Choose initial $\mathbf{p}^{\text{old}}$.
2) **E Step** Evaluate $\Pr(\mathbf{x}|\mathbf{y}, \mathbf{p}^{\text{old}})$.
3) **M Step** Evaluate

$$\mathbf{p}^{\text{new}} = \arg\max_{\mathbf{p}} \sum_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{y}, \mathbf{p}^{\text{old}}) \log \Pr(\mathbf{x}, \mathbf{y}|\mathbf{p}) \quad (1)$$

4) Check for convergence of the log likelihood or the parameter values. If convergence is not achieved, then $\mathbf{p}^{\text{old}} \leftarrow \mathbf{p}^{\text{new}}$ and go to Step 2.

With each iteration, the log-likelihood will increase until it approaches a local maximum.

While the EM algorithm can be applied to general DMCs, for clarity the description is given for a general two-output channel; this may be termed the binary asymmetric channel (BAC). The BAC has error probability $p_{\mathsf{Y}|\mathsf{X}}(1|0) = p_0$ and $p_{\mathsf{Y}|\mathsf{X}}(0|1) = p_1$. The channel input sequence is $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and the channel output sequence is $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. The a priori information is $\mathbf{l} = (l_1, l_2, \ldots, l_n)$ where $l_i = \Pr[\mathsf{X}_i = 0]$. The goal is to estimate the BAC error probability $\widehat{p}_0, \widehat{p}_1$ from $\mathbf{y}_i$ and $l_1, \ldots, l_n$ using the EM algorithm.

For the E Step, use the initial estimates $\widehat{p}_0 \leftarrow p_0^{\text{old}}$ and $\widehat{p}_1 \leftarrow p_1^{\text{old}}$, and compute the function $q(x, y, l)$ given by:

$$\frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)}{p_{\mathsf{Y}}(y)} = \begin{cases} \frac{(1-\widehat{p}_0)l}{(1-\widehat{p}_0)l+\widehat{p}_1(1-l)} & \text{if } (x,y)=(0,0) \\ \frac{\widehat{p}_1(1-l)}{(1-\widehat{p}_0)l+\widehat{p}_1(1-l)} & \text{if } (x,y)=(1,0) \\ \frac{\widehat{p}_0 l}{\widehat{p}_0 l+(1-\widehat{p}_1)(1-l)} & \text{if } (x,y)=(0,1) \\ \frac{(1-\widehat{p}_1)(1-l)}{\widehat{p}_0 l+(1-\widehat{p}_1)(1-l)} & \text{if } (x,y)=(1,1) \end{cases} \tag{2}$$

For the M Step, perform the following optimization:

$$\widehat{p}_0, \widehat{p}_1 = \arg\max_{p_0,p_1} \sum_{i=1}^{n} \sum_{x} q(x_i, y_i, l_i) \log p_{\mathsf{XY}}(x_i, y_i). \tag{3}$$

Using Lagrange multipliers, the estimates $\widehat{p}_0$ and $\widehat{p}_1$ are:

$$\widehat{p}_0 = \frac{\sum_{i:y_i=1} q(0,1,l_i)}{\sum_{i:y_i=1} q(0,1,l_i) + \sum_{i:y_i=0} q(0,0,l_i)} \tag{4}$$

$$\widehat{p}_1 = \frac{\sum_{i:y_i=0} q(1,0,l_i)}{\sum_{i:y_i=0} q(1,0,l_i) + \sum_{i:y_i=1} q(1,1,l_i)} \tag{5}$$

III. LDPC-coded NAND flash memory system

Consider the use of the EM algorithm in the following LDPC-coded system in Fig. 1. Information $\mathbf{u}$ is encoded to an LDPC codeword $\mathbf{x}$ of an (n, k) LDPC code. Then, the codeword $\mathbf{c}$ is programmed to each cell of NAND flash memory. To model possible charge leakage, a constant offset $a \geq 0$ is subtracted from programmed levels. Additive white Gaussian noise z_i, with mean 0 and variance σ^2, is added to the signal $\widetilde{y}_i = x_i + z_i - a$, for $i = 1, \ldots, n$. The SNR definition is $1/\sigma^2$. To model the NAND flash read process, $\widetilde{y}$ is quantized to two levels by a threshold at 0, producing two channel output levels. This is a binary asymmetric channel (BAC) where the error probabilities p_0, p_1 depend on a, σ^2.

Numerical evaluations are performed using this channel. The (4608,4096) and (36864,32768) LDPC codes have rate 8/9, column weight of 3, and the row weight varies from 26–29 and 26–28, respectively. This LDPC code is constructed using the progressive-edge-growth (PEG) algorithm [5]. For the LDPC decoder, we use the belief propagation (BP) algorithm. Numerical results for three systems are shown in Fig. 2.

In the **nonblind system**, the detector knows p_{xy} exactly and these are used in channel estimation. Decoding is performed for $I_{\text{LDPC}} = 30$ and 90 iterations. This represents the ideal case, since the channel is known.

In the **blind system**, the detector does not know p_{xy}. Instead, it assumes a symmetric channel with low SNR, specifically $\widehat{a} = 0$ and $\widehat{SNR} = 5$ dB. Decoding is performed for $I_{\text{LDPC}} = 30$ iterations.

In the **EM system**, the decoder uses the EM algorithm in a turbo equalization-style setting, as shown in Fig. 1. Initially, the decoder has no knowledge of p_{xy} and makes an initial estimate $\widehat{p}_{xy}$. This is used to compute LLRs, $L_{\text{in}} = \log \frac{\Pr[x=0|y_i]}{\Pr[x=1|y_i]} = \log \frac{\widehat{p}_{0y_i}}{\widehat{p}_{1y_i}}$. Using these, the LDPC decoder operates for I_{LDPC} iterations, and produces soft outputs $\mathbf{l}$. Next the EM algorithm performs channel estimation. It has inputs $\mathbf{y}$, a priori information $\mathbf{l}$ and the old estimate $\widehat{p}_{xy}^{\text{old}}$. It operates for I_{EM} iterations. It outputs a new channel estimate $\widehat{p}_{xy}^{\text{new}}$. This is used to improve the computation of the LLRs, which is used for the next I_{LDPC} decoding iterations.

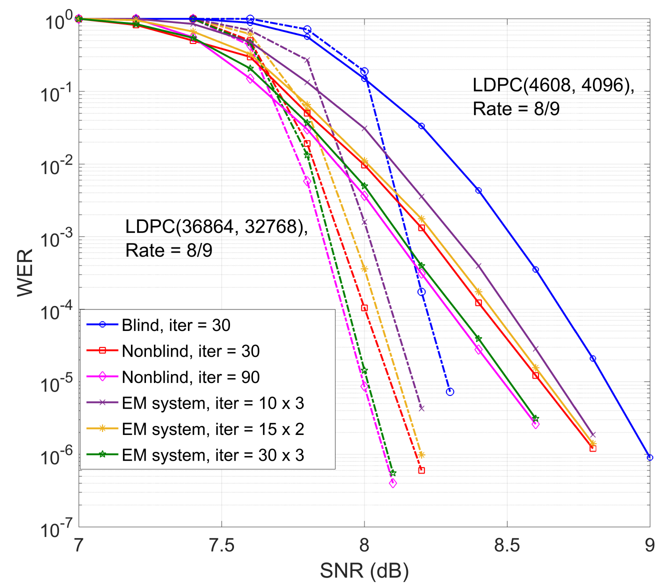

Fig. 2. WER results for channel with offset $a = 0.5$. Rate 8/9 LDPC code with $n = 4608$ (solid lines) and 36864 (dashed lines).

In the numerical results, the EM system initially assumes the same conditions as the blind system. Three iteration schedules are shown, $I_{\text{LDPC}} \times I_{\text{turbo}} = 10 \times 3, 15 \times 2$ and 30×3. The EM algorithm operates for $I_{\text{EM}} = 8$ iterations. The results show that the 15×2 and 30×3 EM system give the closet WER to the nonblind system with $I_{LDPC} = 30$ and 90 iterations, respectively. The 10×3 EM system loses the performance around 0.1 dB due to worse estimation of $\widehat{p}_{xy}$.

The main conclusion from the numerical results is that the EM algorithm can estimate unknown channel error probabilities well enough to provide the similar WER as the nonblind detector which knows the channel exactly. This is about 0.3 dB better than the blind system which has no channel knowledge.

References

[1] J. J. Boutros, A. Alloum, and G. I. Shamir, "Enhanced channel decoding via em source-channel estimation," in *Proc. 2nd International Symposium Commun., Control, Signal Processing*, 2006.

[2] E. Sharon and A. Bazarsky, "Dynamic memory error model estimation for read and ecc adaptations," in *Proc. Non-Volatile Memories Workshop*, 2017.

[3] Z. Mei, K. Cai, and X. He, "Deep learning-aided dynamic read thresholds design for multi-level-cell flash memories," *CoRR*, vol. abs/1907.03938, 2019. [Online]. Available: http://arxiv.org/abs/1907.03938

[4] C. M. Bishop, *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006.

[5] Xiao-Yu Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, Jan 2005.