Kikuchi Approximation Method for Joint Decoding of LDPC Codes and Partial-Response Channels

Payam Pakzad (payamp@eecs.berkeley.edu)
Venkat Anantharam (ananth@eecs.berkeley.edu)

4/1/2004
Berkeley, California

Abstract

In this paper we apply the Kikuchi approximation method to the problem of joint decoding of a low-density parity-check (LDPC) code and a partial response (PR) channel. The Kikuchi method is in general more powerful than the conventional loopy belief propagation (BP) algorithm, and can produce better approximations to an underlying inference problem. We will first review the Kikuchi approximation method and the generalized belief propagation (GBP) algorithm, which is an iterative message-passing algorithm based on the Kikuchi method. We will then show that for the specific code and channel considered, the Kikuchi method outperforms the best conventional iterative method.

1 Introduction

Iterative decoding techniques, such as the decoding algorithms for turbo codes (see [2]) and low-density parity-check (LDPC) codes (see [3, 5]), have enjoyed much attention in the recent years due to their apparent ability to efficiently produce good approximations to the optimal maximum *a posteriori* (MAP) estimates. In applications such as magnetic storage where a target bit-error-rate (BER) needs to be achieved, these codes can operate at much lower SNRs than any other practical error-correcting approach. Operation at lower SNRs, in turn, translates to achieving higher bit densities on the same storage device. At high densities, any efficient decoding technique must properly model and address the issue of intersymbol interference (ISI). The ISI in magnetic recording channels are conventionally described using partial response (PR) models, in which the channel is modelled as a finite impulse response (FIR) filter. "Turbo equalization" is an iterative technique in which information is passed back and forth between soft decoders for the ISI channel, and a preceding error-correcting code. As we will see later in Section 3, this corresponds to an implementation of the well-known belief propagation (BP) algorithm of [9] on a graph with cycles.

In this paper we propose to use a more powerful iterative algorithm, known as the generalized belief propagation (GBP) algorithm based on Kikuchi approximation method.

We will briefly introduce the Kikuchi approximation method and the GBP algorithm in Section 2. Readers are referred to [8] and [11] for more details. We will then apply this method to a particular "turbo equalization" problem, which was addressed using conventional methods in [4]. We will show that the Kikuchi based GBP algorithm outperforms the best iterative method based on the belief propagation algorithm.

2 Kikuchi Approximation Method

We will now give a short description of the Kikuchi approximation method for calculating the desired marginals of a product distribution. Let $\mathbf{x} := (x_0, \cdots, x_{N-1})$, where for each $i \in [N] := \{0, \cdots, N-1\}$, x_i is a variable taking value in $[q_i] := \{0, \cdots, q_i - 1\}$, with $q_i \geq 2$.

Let R be a collection of subsets of $[N]$; we call each $r \in R$ a *region*. We assume that each variable index $i \in [N]$ appears in at least one region $r \in R$.

Associated with each region $r \in R$ is a nonnegative *kernel function*, $\alpha_r(\mathbf{x}_r)$, depending only on the variables that appear in r. Then the corresponding *R-decomposable* (*Boltzmann*) *product distribution* is defined as

$$B(\mathbf{x}) := \frac{1}{Z} \prod_{r \in R} \alpha_r(\mathbf{x}_r) \tag{1}$$

Here Z is the normalizing constant and is called the *partition function*. For a subset $s \subset [N]$, we denote by $B_s(\mathbf{x}_s) := \sum_{\mathbf{x}_{[N] \setminus s}} B(\mathbf{x})$ the s-marginal of $B(\mathbf{x})$. We are interested in finding one or more of the $B_r(\mathbf{x}_r)$'s for $r \in R$, and/or the partition function Z.

Let $b(\mathbf{x})$ denote a probability distribution on $\mathbf{x}$. We define the *variational free energy* for the problem as

$$F(b(\mathbf{x})) := U(b(\mathbf{x})) - H(b(\mathbf{x})) \tag{2}$$

where $U := \sum_{\mathbf{x}} b(\mathbf{x}) \prod_{r \in R} \alpha_r(\mathbf{x}_r)$ is the *average energy* and $H := -\sum_{\mathbf{x}} b(\mathbf{x}) \log(b(\mathbf{x}))$ is the *entropy* of the system.

It can be shown that $F(b)$ is uniquely minimized when $b(\mathbf{x})$ equals the Boltzmann distribution $B(\mathbf{x})$ of (1), and we have

$$F_0 := \min_{b(\mathbf{x})} F(b(\mathbf{x})) = F(B(\mathbf{x})) = -\log(Z). \tag{3}$$

The Kikuchi approximation method proposes to solve a related constrained minimization problem of the following form (see [8] for details):

$$\{B_r(\mathbf{x}_r)\} \simeq \{b_r^*(\mathbf{x}_r)\} := \arg \min_{\{b_r(\mathbf{x}_r)\} \in \Delta_R^K} F_R^K(\{b_r(\mathbf{x}_r)\}) \tag{4}$$

Here $F_R^K(\{b_r\})$, known as the *Kikuchi free energy*, see e.g. [8], is defined as

$$F_R^K(\{b_r(\mathbf{x}_r)\}) := \sum_{r \in R} \sum_{\mathbf{x}_r} b_r(\mathbf{x}_r) E_r(\mathbf{x}_r) + \sum_{r \in R} \sum_{\mathbf{x}_r} c_r \, b_r(\mathbf{x}_r) \log(b_r(\mathbf{x}_r)) \tag{5}$$

where c_r's are constants known as the *overcounting factors*, and are uniquely defined given the collection R of regions. Also, Δ_R^K is a set of constraints to enforce consistency between the b_r's, and is defined as

$$\Delta_R^K := \Big\{\{b_r(\mathbf{x}_r),\, r \in R\} \;:\; \forall\, t, u \in R \;\text{ s.t. } t \subset u, \sum_{\mathbf{x}_{u\setminus t}} b_u(\mathbf{x}_u) = b_t(\mathbf{x}_t) \\ \text{and } \;\forall u \in R,\; \sum_{\mathbf{x}_u} b_u(\mathbf{x}_u) = 1\Big\} \quad (6)$$

We will refer to the constrained minimization problem of (4) as the *Kikuchi approximation problem*, where it is understood that the desired marginals $\{B_r(\mathbf{x}_r)\}$ are approximated by the minimizers $\{b_r^*(\mathbf{x}_r)\}$.

As discussed in [10], belief propagation is an algorithm that tries to solve a simple class of Kikuchi approximation problems known as the Bethe case. When using a suitable choice of the collection R of the regions, the *Kikuchi beliefs* $\{b_r^*(\mathbf{x}_r)\}$ of (4) can better approximate the true marginals $\{B_r(\mathbf{x}_r)\}$ of the product distribution than the beliefs obtained from the conventional belief propagation algorithm, see e.g. [8] for discussion and examples.

2.1 Graphical Representations of the Kikuchi Problem

The standard technique to solve a constrained optimization problem such as that of (4) is to form the Lagrangian, where for each constraint of (6) a multiplier will be defined. These multipliers (or a function of them) play the role of 'messages' in an iterative message-passing algorithm, see [8] for details. We will describe one such algorithm called GBP in Section 2.2 below.

As with the belief propagation algorithm, graphical models can be used to represent a Kikuchi problem and serve as the basis for such iterative message-passing algorithm to solve that problem. We therefore define a graphical representation of a Kikuchi problem as a graph, whose edges correspond to the consistency constraints that define Δ_R^K. Clearly such representations are not unique, since redundant constraints (and the corresponding edges) may be freely added or removed. It is therefore advantageous to find the *minimal* such graphical representation; algorithms on such minimal graphs have the fewest messages and are hence the least complex per each iteration. A detailed discussion of graphical representations of a Kikuchi problem can be found in [8].

2.2 Generalized Belief Propagation Algorithm

Generalized belief propagation (GBP) is a message-passing algorithm that tries to solve the Kikuchi constrained minimization problem (4).

Let G be a graphical representation of the Kikuchi problem (4). We associate with each (region) vertex r of G a *belief* function $b_r(\mathbf{x}_r)$. We also associate with each edge $(p \to r)$ of G a *message* function $m_{pr}(\mathbf{x}_r)$. For each $r \in R$ define $\mathcal{D}(r) := \{s \in R \,:\, s \subset r\}$, and $\mathcal{P}_G(r) := \{s \in R \,:\, (s \to r) \text{ an edge in } G\}$. At each iteration, belief $b_r(\mathbf{x}_r)$ is updated

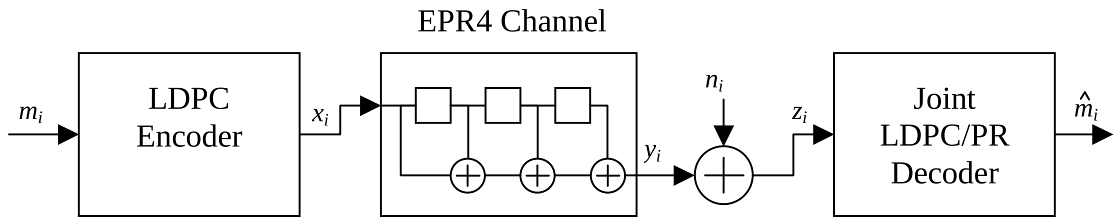

Figure 1: Block Diagram of the Serial Concatenation of an LDPC code and a PR channel.

as

$$b_r(\mathbf{x}_r) = k\,\beta_r(\mathbf{x}_r)\Big(\prod_{p\in\mathcal{P}_G(r)} m_{pr}(\mathbf{x}_r)\Big)\Big(\prod_{d\in\mathcal{D}(r)}\prod_{p'\in\mathcal{P}_G(d)\backslash(\{r\}\cup\mathcal{D}(r))} m_{p'd}(\mathbf{x}_d)\Big) \tag{7}$$

where constant k is chosen to normalize b_r so it will sum to 1, and a message $m_{pr}(\mathbf{x}_r)$ is updated to satisfy the edge-constraint $\sum_{\mathbf{x}_{p\backslash r}} b_p(\mathbf{x}_p) - b_r(\mathbf{x}_r) = 0$, i.e.

$$m_{pr}(\mathbf{x}_r) = k'\frac{\sum_{\mathbf{x}_{p\backslash r}} \beta_p(\mathbf{x}_p)\Big(\prod_{s\in\mathcal{P}_G(p)} m_{sp}(\mathbf{x}_p)\Big)\Big(\prod_{d\in\mathcal{D}(p)}\prod_{s'\in\mathcal{P}_G(d)\backslash(\{p\}\cup\mathcal{D}(p))} m_{s'd}(\mathbf{x}_d)\Big)}{\beta_r(\mathbf{x}_r)\Big(\prod_{s\in\mathcal{P}_G(r)\backslash\{p\}} m_{sr}(\mathbf{x}_r)\Big)\Big(\prod_{d\in\mathcal{D}(r)}\prod_{p'\in\mathcal{P}_G(d)\backslash(\{r\}\cup\mathcal{D}(r))} m_{p'd}(\mathbf{x}_d)\Big)} \tag{8}$$

where k' is any convenient constant. Note that the common terms from the numerator and denominator of (8) can be cancelled, but to avoid even longer expressions we will not write the explicit form here.

As shown in [8], the fixed points of (7) and (8) above are precisely the stationary points of the constrained minimization problem (4).

3 Joint Decoding of LDPC Code and PR Channel

Consider a partial response channel precoded by a low-density parity check code, as depicted in Figure 1.

We identify the LDPC code by its parity check matrix $H_{m\times n}$ where n is the blocklength of the code, and m is the number of parity checks. Therefore a codeword $\mathbf{x} := (x_1, \cdots, x_n)$ satisfies $H\cdot\mathbf{x} = \mathbf{0}$. The partial response channel is identified by a transfer polynomial $h(D) := \sum_{i=0}^{\nu} h_i D^i$, where ν is the degree of the channel. For example, the EPR4 channel depicted is identified by $h(D) = 1 + D - D^2 - D^3$. Therefore the output of the channel is related to its input by $y(D) = h(D)x(D)$, in the Z-transfer domain. We will assume an additive white Gaussian noise (AWGN) with variance $\sigma_{\rm n}^2$. The objective is to find the maximum likelihood estimates of the transmitted code symbols x_i's given the noisy observations $\mathbf{z} = (z_1, \cdots, z_n)$.

It is clear that this problem can be described as that of finding the marginals of a product function, as posed in Section 2. Let $P(\mathbf{x})$ denote the joint distribution of the

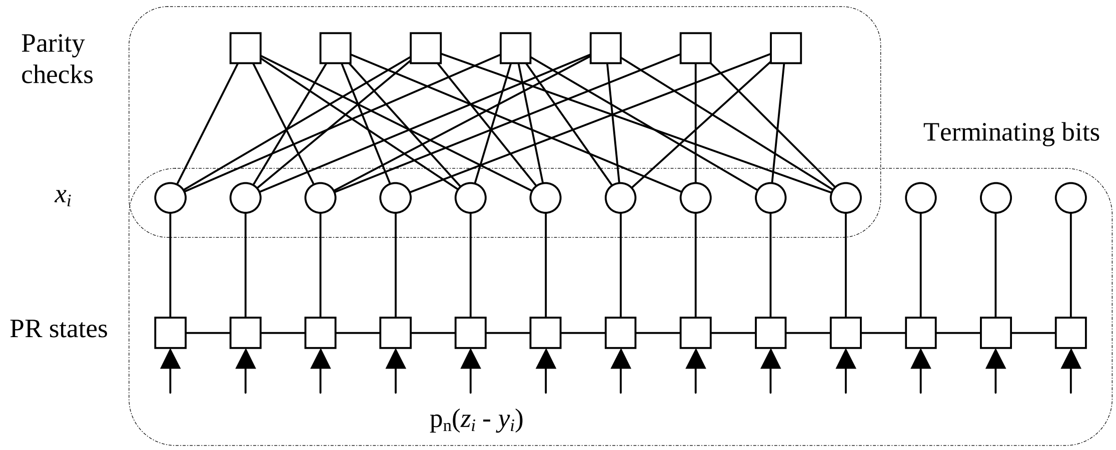

Figure 2: Graphical model for joint BP decoding of LDPC/PR problem

codeword $\mathbf{x}$ given the observations. Then

$$P(\mathbf{x}) = \frac{1}{Z}\prod_{j=1}^{c} 1(H_j \cdot \mathbf{x} = 0) \prod_{i=1}^{n} p(z_i|\mathbf{x}) \tag{9}$$

$$= \frac{1}{Z}\prod_{j=1}^{c} 1(H_j \cdot \mathbf{x} = 0) \prod_{i=1}^{n} p_{\mathrm{n}}(z_i - \sum_{j=0}^{\nu} h_j x_{i-j}) \tag{10}$$

where H_j denotes the jth row of the parity check matrix H, and p_{n} is the probability density of the noise. In particular, for the AWGN of variance σ_{n}^2, we have $p_{\mathrm{n}}(n) = ke^{-\frac{n^2}{2\sigma_{\mathrm{n}}^2}}$.

The best performing method discussed in [4] involves iteration between BCJR decoding of the PR channel (see [1]), and the Gallager-Tanner decoding of the LDPC code (see [3]). This corresponds to the standard BP algorithm performed on the graph of Figure 2, with three classes of nodes: the 'bit-nodes' corresponding to the bits x_i of the LDPC code; the 'check-nodes' corresponding to the parity checks of the LDPC code; and 'PR-state-nodes' corresponding to the states of the PR channel. The corresponding regions are, respectively, $\{i\}$; $\{j : \mathbf{H}_{i,j} = 1\}$; and $\{i - j : h_j \neq 0\}$, for all possible values of i.

To apply the Kikuchi approximation method for this problem, we used the poset obtained by the cluster variation method. Specifically, we appended all the intersection of the above regions to form the collection R of regions. Notice that good LDPC codes do not have small loops of size 4, so no two check-nodes can intersect in more than one index. However, the check-nodes can have nontrivial intersections with the PR-state-nodes. We considered a specific example from [4], with a rate 7/8 LDPC code with block-length $n = 495$, and with an EPR4 channel. The resulting poset had 495 bit-node regions (singletons), 62 check-node regions (each of size 24, since the LDPC code is regular with 24 bits per check), 495 PR-state-node regions (each of size 4, since the channel is EPR4), and a total of 659 nontrivial intersection regions, with nonzero overcounting factors. Of these 659 regions, 494 correspond to the intersections of neighboring PR-state-node regions (each of size 3). The remaining 165 regions (with sizes 2 or 3) are the new regions

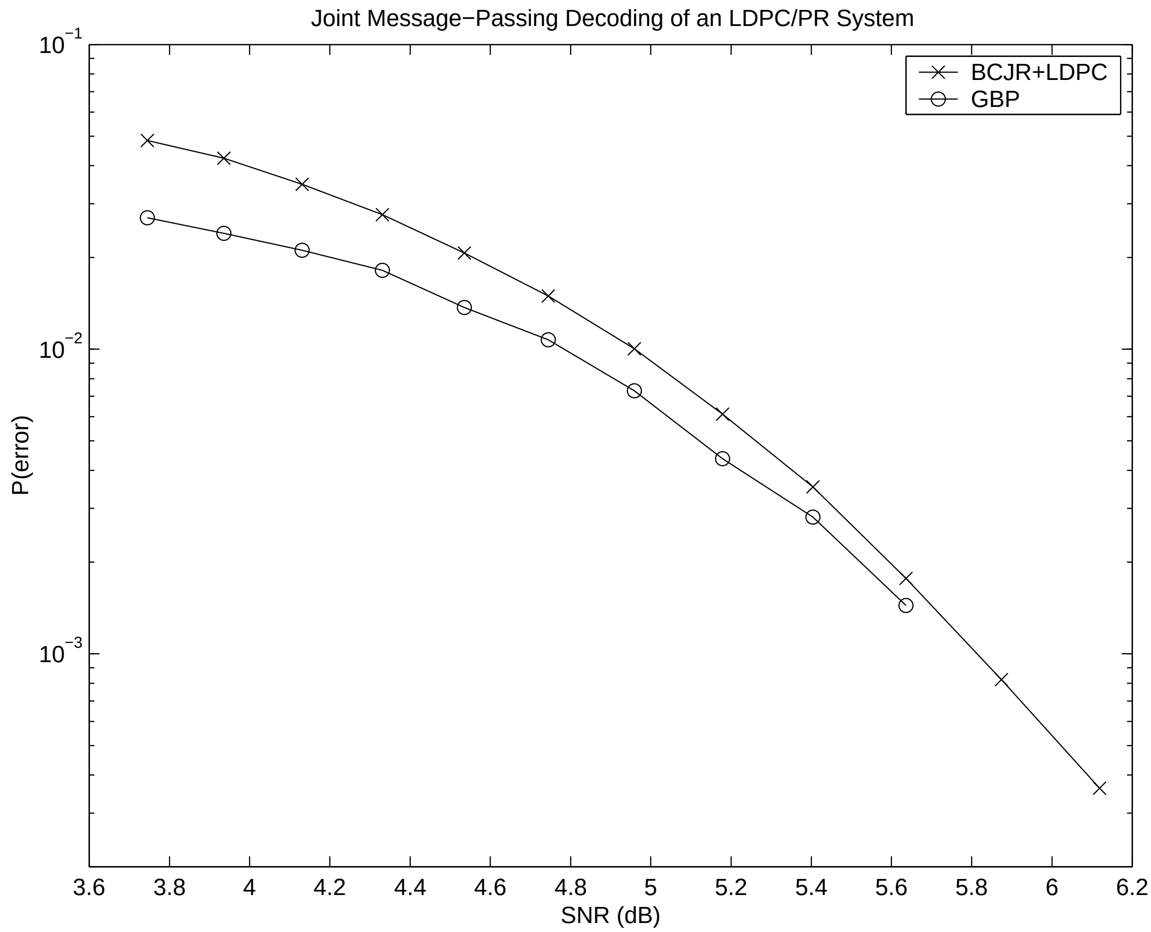

Figure 3: Simulation results for joint decoding of LDPC/PR

that make the difference from the belief propagation algorithm.

The full Hasse diagram on this collection of regions has 1711 vertices, and 3973 edges. The minimal graph for this collection has 1711 vertices and 2951 edges. For comparison, the corresponding graph for the original problem, before addition of the intersection regions, which corresponds to the loopy BP algorithm, has 2476 edges.

Simulation results are reported in Figure 3 below. 'BCJR+LDPC' data points are averaged over 500 simulation runs, with 8 iterations between the BCJR and LDPC algorithms, where the LDPC algorithm consisted of 20 iterations. 'GBP' data points are averaged over 50 simulation runs, with 8 full iterations of the GBP algorithm for each run.

These results suggest that, as expected, the GBP algorithm considered performs better than the BCJR+LDPC method. Our new technique appears to be particularly well suited to the low SNR regime, which is the one that is most important for current magnetic recording applications.

4 Acknowledgments

This work was supported by grants from (ONR/MURI) N00014-1-0637, (NSF) SBR-9873086, (DARPA) F30602-00-2-0538, California Micro Program, Texas Instruments, Marvell Technologies and ST MicroElectronics.

References

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J.Raviv. "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Transactions on Information Theory*, 20(2):284–287, March 1974.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near shannon limit error-correcting coding and decoding: Turbo-codes". In *Proceedings of the IEEE International Conference on Communications*, number 2, pages 1064–1070, Geneva, May 1993.

[3] R.G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.

[4] B. Kurkoski, P. Siegel, and J. Wolf. "Joint message-passing decoding of ldpc codes and partial-response channels", *IEEE Transactions on Information Theory*, 48(6):1410–1422, June 2002.

[5] D.J.C. MacKay and R.M. Neal. "Good codes based on very sparse matrices". In *Cryptography and Coding: 5th IMA Conference*, number 1025, pages 100–111. Springer-Verlag, Berlin, 1995.

[6] R.J. McEliece and M. Yildrim, *"Belief Propagation on Partially Ordered Sets,"* 15th MTNS International Symposium, 2002

[7] P. Pakzad and V. Anantharam, *"Belief Propagation and Statistical Physics,"*, Proc. CISS Mar. 2002.

[8] P. Pakzad and V. Anantharam, *"Estimation and Marginalization using Kikuchi Approximation Method,"*, Submitted for publication in *Neural Computation*.

[9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann, 1988.

[10] J.S. Yedidia, W.T. Freeman and Y. Weiss, *"Bethe Free Energy, Kikuchi Approximations, and Belief Propagation Algorithms,"*, MERL T.R., 2000.

[11] J.S. Yedidia, W.T. Freeman and Y. Weiss, *"Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms,"*, MERL TR2002-40, August 2002.