No. 41 2001 年 12 月 26 日発行

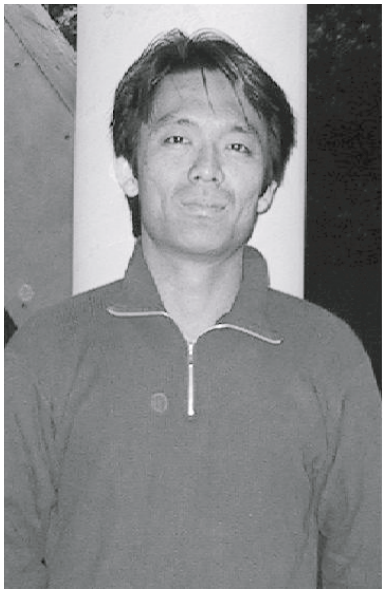
SITA 情報理論とその応用学会ニューズレター

「確率伝搬に基づく復号法と符号ワークショップ」技術特集

巻頭言

ワークショップオーガナイザ　井坂 元彦（東京大学）

情報理論とその応用学会主催の本ワークショップは，2001 年 3 月 13 日に東京大学駒場キャンパスにて 6 件の講演，130 名強の参加者を集めて開催されました．基礎的事項とアドバンスコース，理論と応用に関する講演をバランスよく配置するのが理想的ではありましたが，当初想定していた話題に対する講演者の選定が難航した結果，断念せざるをえなかったことをはじめとして，オーガナイザとしては多くの課題が残りました．それでも，ある程度の好評は頂けたようで，盛況なワークショップの原動力となった参加者の皆様，講演者の先生方，運営に協力頂いた各位に深く感謝を申し上げる次第です．



最近の（1999 年までの）研究動向については，筆者が駄文を記すよりも本年刊行の特集号 [1,2,3] のまえがき及び論文群が遥かに参考になりますので，そちらを参照下さい．ターボライク符号や低密度パリティ検査符号及びその反復復号法の研究が”ホットトピック”であった時代はこれらの成果が投稿された時期をもって収束したように認識していますが，これは符号研究の一分野として十分に定着したことの裏返しとも捉えられます．一方他分野と同様に困難な未解決問題は数多く残存しており，今後はこれらの基礎的な検討が徐々に進展していく段階に入るのではないかと思われます．特にグラフによる状態実現においてサイクルを許容することが問題を難解にしており，テイルバイティングトレリスですらサイクルフリーの状態実現 [4] のような見通しのよい議論ができず，また漸近的な反復特性限界とは裏腹に，有限長の符号に対する反復復号については，極めて限られた知見しか見られていません．符号長に対して線形時間で復号可能な通信路容量を与える符号の構成及び解析などと併せて，実用的に役立つ話題では必ずしもないものの，理論的興味は尽きないものがあります．筆者は SITA2001 のため，残念ながら参加を見送りましたが，DIMACS ワークショップ [5] においても最新成果に関して興味深い議論が種々なされたようです．一方で，これらと並行して情報・通信理論の諸問題に対する反復復号手法の応用や，実用化に直結した話題は今後も継続的に検討されていくことが予想されます．

残念なことに，本分野を総括する際に不可欠となるような”Key Paper”は，国内からは出ていないのが現状のように思われます．自身に対する反省と叱咤も込めて，日本発の重要な成果が今後生まれることに期待したいと考えています．

[1] C.E. Shannon, "A mathematical theory of communications," Bell System Technical Journal, vol.27, pp.379-423, July, pp.623-656, Oct., 1948.

[2] Special issue on "Codes on graphs and iterative decoding", IEEE Trans. Inform. Theory, Feb. 2001.

[3] Special issue on "The turbo principle: theory and practice, part-I", IEEE J. Sel. Areas in Commun., May 2001.

[4] Special issue on "The turbo principle: theory and practice, part-II", IEEE J. Sel. Areas in Commun., Aug. 2001.

[5] S. Lin, T. Kasami, T. Fujiwara and M. Fossorier, Trellises and trellis-based decoding algorithms for linear block codes, Kluwer Academic Publishers, 1998.

[6] DIMACS workshop on codes and complexity, Piscataway, NJ, Dec. 2001. See http://dimacs.rutgers.edu/Workshops/Codescomplexity/

２１世紀の符号理論の中のターボ符号の誕生

山口 和彦（電気通信大学）

1　はじめに

この原稿は、２００１年3月１３日に行われた情報理論とその応用学会ワークショップ「確率伝搬に基づく復号法と符号ワークショップ」 —ターボ符号，低密度パリティ検査符号の基礎，理論と応用においての中で、ターボ符号の誕生 - ターボ符号以前の誤り訂正技術と比較して　-としての発表に元にしている。同発表は、ワークショップの第１番目の話題として、ターボ符号やターボ復号，低密度パリティ検査符号等の様々話題の導入を目的としたものであった。

本原稿は短い文章ではあるが SITA のニューズレターを意識し、現代そして将来の符号理論とターボ符号に焦点を当てることとした。以下，基本的な話題のみを取り上げていおり、釈迦に説法となる面の多いことにはご容赦いただきたい。

2　歴史：ターボ符号の誕生

フランスの Berrou らが ICC ９３（通信分野の国際会議）で行った発表に始まり、第３世代の移動通信や、様々な実用化が進められるターボ符号，ターボ復号。実用を踏まえた研究や、信号処理等の関連分野への研究も進む中これまで以上にターボ符号・復号に関わる技術者も多いだろう。

ターボ符号とターボ復号、どちらも高い信頼度の誤り訂正方式を実現するための重要な両輪だが、方式が生まれるまでの過程を考えれば，復号に鍵があると考えられる。衛星通信での軟判定復号、深宇宙通信での直列連接符号の１段目の復号における軟入力軟出力復号という流れに密接に関係している。

ターボ復号の注目点はこの直列連接の確率伝播の仕組みを繰り返しフィードバックに変えた点にある。その構造を実現する符号化法がターボ符号なのだ。

真の発明・発見者尊重するという向きからは Gallager による低密度パリティ検査符号の復号に用いられた復号や、Massy の最大事後確率復号に確率伝搬手法を用いる復号の原典があるといえる。

ただ、再帰的な畳込み符号を並列に連接させるターボ符号と、線形ブロック符号の一種である低密度パリティ検査符号は非常によく似ている（技術的な論文でないのを生かし、曖昧な表現を使う。）言い方を変えれば、良い符号を設計する場合の設計規範の多くは通じる所があるようだ。

ターボ符号の出現により、低密度パリティ検査符号の研究も盛んになっている。今後総合的な理解が深まっていくだろう。符号理論の歴史という面からは、代数的な符号理論とのターボ符号、低密度パリティ検査符号は関係が希薄な点も今後の成果に期待したい。

ただ、Lin によれば、低密度パリティ検査符号を代数的の構成しようと試みたところ、差集合巡回符号こそが、その一つの例となるという。日本のＴＶ、ＦＭ多重の文字放送等に利用された符号が、一種の低密度パリティ検査符号であることは興味深い。文字放送では、硬い判定ではあるが復号結果をフィードバックする繰り返しアルゴリズム「可変閾値復号法」が当初から採用されている。

3　符号理論とターボ符号

ターボ符号のテキストは既に Heggard や、Bucettic により執筆出版されている。そのほかの教育的な資料も多く実用化されているターボ符号を理解するには十分詳細な内容を提供している。しかし、２章に述べたようにターボ符号は複数の技術の組み合わせであるため、その複雑な内容をきちんと理解するにはまだ敷居が高いようだ。

尤度情報の正確な扱いの理解等は一番大きな問題といえる。よく理解に用いられる縦横２次元のパリティ検査符号による繰り返し復号の例題と実際のターボ符号は基本的には同じ動作なのではあるが、その対応が結びつかないようだ。もちろん、符号理論をターボ符号とそれ以外の誤り訂正符号を関係付けて学ぶことは、現時点では難しく、時間がかかることに思われる。

２０世紀中には和文でも優れた符号理論の本が多数出版されているように感じている。ターボ符号の内容を的確に扱った良い符号理論のテキストが、今、もっとも期待されている。ただ、それがどういう扱い方であるかはまだ山口には回答はない。

SOVA と Max-Log-Map

田島 正登（富山大学）

1 まえがき

ビタビアルゴリズム (VA : Viterbi algorithm) [1] は大局的な最適化アルゴリズムであり，ビット（シンボル）単位の局所的な最適化を行うには，事後確率 (APP : a posteriori probability) を最大化する復号方式である，MAP(maximum a posteriori) 復号法を必要とする．MAP 復号法を実現するには，事後確率の対数比である対数尤度比 (LLR : log-likelihood ratio) を計算すればよいが，これを忠実に実行するアルゴリズムが BCJR アルゴリズム [2] である．BCJR アルゴリズムを近似する準最適なアルゴリズムとして既にいくつかの方式が提案されている [3]-[10]．この内，この報告では，SOVA(soft-output Viterbi algorithm) [9]，および，Max-Log-MAP [10]-[12] を取り上げる．SOVA は Berrou 等がターボ符号の考案に至る過程でその LSI 化に注力したアルゴリズムであり [13], [14]，基本的な軟出力 (soft-output) 復号方式と見なされる．また，Max-Log-MAP は，MAP の対数 (ln) 領域での近似方式であるが，結果的には，双方向（前向きおよび後ろ向き）の VA の実行と基本的に等価であることが示される [10]．以下では，SOVA と Max-Log-MAP についてアルゴリズムの導出を行った後，両者の関係について述べる．これらの軟出力復号アルゴリズムは，ターボ符号 [15] の復号など，繰り返し復号法 [3] において本質的な役割を果たす．

2 準備

以下では，原則として，符号化率 1/2，メモリ長 ν の再帰的組織畳込み (RSC : recursive systematic convolutional) 符号を扱う．情報系列はフレーム化されており，有効情報系列 $\{u_j\}_{j=1}^{N}$ に続いて（$t=N$ の状態 s_N に依存した）ν 個のテイルビットが入力されて，トレリスの状態 $s_{N+\nu}=(0)$(全零状態) で終端されると仮定する．対応する受信系列を $z=\{z_j\}$ とする．受信系列 z に基づく最尤 (ML : maximum-likelihood) パスを $\hat{x}=\{\hat{u}_j\}$ と表す．以下，最尤パスに関係する量 A は $\hat{A}$ と書くことにする．また，トレリス上のパス x^a の時刻 $t=j$ に対応する情報シンボルを u_j^a と表現する．この報告では，対数尤度比

$$L(\hat{u}_k) \equiv \ln \frac{P(u_k=0|z)}{P(u_k=1|z)} \tag{1}$$

の計算方式が議論の対象である．

3 SOVA

3.1 SOVA の導出-その 1

シンボル u_k の復号結果 $\hat{u}_k$ が正しくない確率 $\hat{p}_k$ の更新式に注目する．ML パス上の $t=j$ の状態を $\hat{s}_j$ とする．ここで，$\hat{s}_j$ に対する競合パスに着目し（ML パスとのメトリック差を Δ_j とする），競合パス上の $\hat{u}_k$ に対応する値を $u_k^{(2)}$ とする．また，$u_k^{(2)} \neq \hat{u}_k$ を仮定する．$\hat{s}_j$ における ACS(add-compare-select) 演算の結果，復号結果が正しくない場合として次の二つが考えられる：

(i) ML パスが正しく選択される（確率は $1-p_{\hat{s}_j}$）$\rightarrow$ $\hat{u}_k$ が復号される $\rightarrow$ これが正しくない確率 $=\hat{u}_k$ が正しくない確率 $=\hat{p}_k$

(ii) 競合パスが誤って選択される（確率は $p_{\hat{s}_j}$）$\rightarrow$ $u_k^{(2)}$ が復号される $\rightarrow$ これが正しくない確率 $=u_k^{(2)}(=1-\hat{u}_k)$ が正しくない確率 $=\hat{u}_k$ が正しい確率 $=1-\hat{p}_k$

この結果，更新後の復号結果が正しくない確率は $\hat{p}_k(1-p_{\hat{s}_j})+(1-\hat{p}_k)p_{\hat{s}_j}$ と表現され，$\hat{p}_k$ の更新式は

$$\hat{p}_k \leftarrow \hat{p}_k(1-p_{\hat{s}_j})+(1-\hat{p}_k)p_{\hat{s}_j} \tag{2}$$

と書ける．ここで，$\hat{u}_k$ の信頼度は

$$\hat{L}_k = \ln \frac{P(u_k = \hat{u}_k)}{P(u_k = 1 - \hat{u}_k)} = \ln \frac{1 - \hat{p}_k}{\hat{p}_k} \tag{3}$$

で定義されるから，$\hat{p}_k$ へ式 (2) の右辺を代入し，関係式

$$\Delta_j = \ln \frac{1 - p_{\hat{s}_j}}{p_{\hat{s}_j}} \tag{4}$$

を用いると，$\hat{L}_k$ の更新式

$$\begin{aligned}
\hat{L}_k &\leftarrow \ln \frac{1 - \hat{p}_k(1 - p_{\hat{s}_j}) - (1 - \hat{p}_k)p_{\hat{s}_j}}{\hat{p}_k(1 - p_{\hat{s}_j}) + (1 - \hat{p}_k)p_{\hat{s}_j}} \\
&= \ln \frac{\frac{1-\hat{p}_k}{\hat{p}_k}\frac{1-p_{\hat{s}_j}}{p_{\hat{s}_j}} + 1}{\frac{1-p_{\hat{s}_j}}{p_{\hat{s}_j}} + \frac{1-\hat{p}_k}{\hat{p}_k}} \\
&= \ln \frac{e^{\hat{L}_k + \Delta_j} + e^0}{e^{\Delta_j} + e^{\hat{L}_k}} \\
&\approx \max(\hat{L}_k + \Delta_j, 0) - \max(\Delta_j, \hat{L}_k) \\
&= \min(\hat{L}_k, \Delta_j)
\end{aligned} \tag{5}$$

が得られる [9]．なお，途中，近似式

$$\ln \sum_j e^{\alpha_j} \approx \max_j \alpha_j \tag{6}$$

を用いた．式 (5) に基づく信頼度の入れ換えは，$u_k^{(2)} \neq \hat{u}_k$ と $\Delta_j < \hat{L}_k$ が成り立つ全ての k で実行されることに注意する．この方式は，SOVA の register exchange mode [9] と呼ばれる．

3.2 SOVA の導出-その 2

$t = j$ での，ML パスと競合パスとのメトリック差を Δ_j と書く．SOVA では，ML パス上のシンボル $\hat{u}_k$ の信頼度は，δ を生き残りパス収束長として，$\Delta_{k+j}(0 \leq j \leq \delta)$ に依存して決定される．$t = k + j$ における競合パス（x^j と書く）上の $\hat{u}_k$ に対応する値を $u_k^j = \hat{u}_k \oplus e_k^j$ とする．e_k^j はパス x^j に関係したシンボル $\hat{u}_k$ に対するエラーであり，関係式

$$L(e_k^j) = \ln \frac{P(e_k^j = 0)}{P(e_k^j = 1)} = \begin{cases} \infty, & u_k^j = \hat{u}_k \\ \Delta_{k+j}, & u_k^j \neq \hat{u}_k \end{cases} \tag{7}$$

が成り立つ．全ての競合パスに対応したトータルエラーは

$$e_k = \sum_{\substack{\oplus \\ 0 \leq j \leq \delta}} e_k^j \tag{8}$$

と表現される（$u_k = \hat{u}_k \oplus e_k$ が成り立つ）．これより，対数尤度比 $L(\hat{u}_k)$ に関して

$$\begin{aligned}
\ln \frac{P(u_k = 0|z)}{P(u_k = 1|z)} &\approx (1 - 2\hat{u}_k) \ln \frac{P(u_k = \hat{u}_k)}{P(u_k = 1 - \hat{u}_k)} \\
&= (1 - 2\hat{u}_k) \ln \frac{P(e_k = 0)}{P(e_k = 1)} \\
&= (1 - 2\hat{u}_k) \ln \frac{P(\sum_{\substack{\oplus \\ 0 \leq j \leq \delta}} e_k^j = 0)}{P(\sum_{\substack{\oplus \\ 0 \leq j \leq \delta}} e_k^j = 1)} \\
&= (1 - 2\hat{u}_k) L(\sum_{\substack{\oplus \\ 0 \leq j \leq \delta}} e_k^j) \\
&= (1 - 2\hat{u}_k) \sum_{\substack{[+] \\ 0 \leq j \leq \delta}} L(e_k^j) \\
&\approx (1 - 2\hat{u}_k) \cdot \min_{0 \leq j \leq \delta} \Delta_{k+j}
\end{aligned} \tag{9}$$

が得られる [16], [17]．ただし，$u_k^j = \hat{u}_k$ のとき，$\Delta_{k+j} = \infty$ と約束する．なお，途中，L 関数の性質 [3]

$$L(\sum_{\substack{\oplus \\ j}} v_j) = \sum_{\substack{[+] \\ j}} L(v_j) \approx \left(\prod_j sign\left(L(v_j)\right) \right) \cdot \min_j |L(v_j)| \tag{10}$$

を用いた．これより，

$$\Delta_k^H \equiv \min_{0 \le j \le \delta} \Delta_{k+j} \tag{11}$$

は $\hat{u}_k$ の信頼度と見なせる．式 (11) に基づく実現は SOVA の trace back mode [16], [17] と呼ばれる．

3.3 forward SOVA と backward SOVA

この節では，nonrecursive な符号（符号化率 $1/n$，メモリ長 ν）を対象として，forward SOVA と backward SOVA の関係を考察する．$\hat{u}_k$ の forward SOVA による信頼度を Δ_k^H，また，backward SOVA による信頼度を Δ_k^{HB} と書く．今，Δ_k^H は $\Delta_{k+d'}$ により達成される（i.e., $\Delta_k^H = \Delta_{k+d'}$）と仮定し，前向きの復号で $t = k + d'$ で ML パスと競合したパスを x' とする（$u'_k \neq \hat{u}_k$）．x' は $t = k - d''$ で ML パスから分岐を開始していると仮定する．ここで，後ろ向きの復号で，$t = k - d''$ で ML パスと競合したパス x'' に注目する（一般に，x'' は x' に一致しない）．x'' と ML パスとのメトリック差を Δ'' とすると，関係

$$\Delta'' \le \Delta_{k+d'} \tag{12}$$

が成り立つ．さて，x'' については，一般に $u_k^{''} \neq \hat{u}_k$ は保証されないが，常に $u_{k-d''+1}^{''} \neq \hat{u}_{k-d''+1}$ は成り立つので，

$$\Delta_{k-d''+1}^{HB} \le \Delta'' \tag{13}$$

が従う．この結果，

$$\Delta_{k-d''+1}^{HB} \le \Delta'' \le \Delta_{k+d'} = \Delta_k^H \tag{14}$$

が導かれる．なお，もし，x'' について $u_k^{''} \neq \hat{u}_k$ が成り立てば，SOVA の定義に従って

$$\Delta_k^{HB} \le \Delta'' \le \Delta_{k+d'} = \Delta_k^H \tag{15}$$

が得られる．まとめると，

1) 一般に $\Delta_{k-d''+1}^{HB} \le \Delta_k^H$ が成り立つ

2) $u_k^{''} \neq \hat{u}_k$ のとき（e.g., $d'' = 1$），$\Delta_k^{HB} \le \Delta_k^H$ が成り立つ

同様の結果は後ろ向きの復号についても成り立つ．上記の議論より，無条件では $\Delta_k^{HB} = \Delta_k^H$ が成り立たないことに注意する．実際，$\Delta_k^{HB} \neq \Delta_k^H$ となる具体例を構成することができる．

4 Max-Log-MAP

4.1 Max-Log-MAP の導出

対数尤度比 $L(\hat{u}_k)$ の定義にもどって考える．今，$s_{k-1} = s'$，$s_k = s$ とすれば，

$$L(\hat{u}_k) = \ln \frac{\sum_{\substack{(s',s) \\ u_k=0}} P(s', s, z)}{\sum_{\substack{(s',s) \\ u_k=1}} P(s', s, z)} \tag{16}$$

と表現される．右辺は ln(分子) – ln(分母) と変形され，ln(分子) と ln(分母) は同じ形をしているので，以下では，$\sum_{\substack{(s',s) \\ u_k=0}} P(s', s, z)$ を評価する．最初に，結合確率 $P(s', s, z)$ を評価する．この値は

$$\begin{aligned} P(s', s, z) &= P(s', z_{j<k}) P(s|s') P(z_k|s', s) P(z_{j>k}|s) \\ &= P(s', z_{j<k}) \gamma_k(s', s) P(s, z_{j>k}) / P(s_k = s) \end{aligned} \tag{17}$$

と分解される [3]．なお，

$$\begin{aligned} \gamma_k(s', s) &\equiv P(s|s') P(z_k|s', s) \\ &= P(u_k) P(z_k|s', s) \end{aligned} \tag{18}$$

は状態遷移 $s' \to s$ に対応する枝遷移確率を表す．ここで，近似式 (6) を用いると，

$$\ln \sum_{\substack{(s',s)\\ u_k=0}} P(s',s,z) \approx \max_{\substack{(s',s)\\ u_k=0}} [\ln P(s', z_{j<k}) + \ln \gamma_k(s',s) + \ln P(s, z_{j>k}) - \ln P(s_k = s)] \tag{19}$$

が得られる．なお，$-\ln P(s_k = s)$ は補正項を表す [18]．ここで，$\ln P(s', z_{j<k})$ と $\ln P(s, z_{j>k})$ に対し，同じく近似式 (6) を適用すると，

$$\ln P(s', z_{j<k}) \approx \max_{u_1,u_2,\cdots} \ln P(u_1, u_2, \cdots, s', z_{j<k}) \tag{20}$$

$$\ln P(s, z_{j>k}) \approx \max_{\cdots,u_{N+\nu-1},u_{N+\nu}} \ln P(s, \cdots, u_{N+\nu-1}, u_{N+\nu}, z_{j>k}) \tag{21}$$

が得られる．右辺はそれぞれ，前向きの復号での状態 s' に対する best パスのメトリック（$m_f(s')$ と書く），および，後ろ向きの復号での状態 s に対する best パスのメトリック（$m_b(s)$ と書く）に相当している．以上により，

$$\ln \sum_{\substack{(s',s)\\ u_k=0}} P(s',s,z) \approx \max_{\substack{(s',s)\\ u_k=0}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s) - \ln P(s_k = s)] \tag{22}$$

が得られる．ここで，$-\ln P(s_k = s)$ に注目する．$s_k = s$ に関しては，前向きおよび後ろ向きの復号で $\ln P(s_k = s)$ が二重に計算されていることが分かる。従って，これを補正するため，$-\ln P(s_k = s)$ が必要とされる．ただし，recursive 符号では，一般に，$\ln P(s_k = s)$ を評価することが困難であり，

$$\ln P(s_k = s) = \ln \frac{1}{N_s} = const. \tag{23}$$

と置くのが妥当である．ここで，N_s はトレリスの状態数を表す．この結果，$L(\hat{u}_k)$ を近似する際，分子と分母の差により，ちょうどキャンセルする．従って，最終的な $L(\hat{u}_k)$ の近似値は

$$L(\hat{u}_k) \approx \max_{\substack{(s',s)\\ u_k=0}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s)] - \max_{\substack{(s',s)\\ u_k=1}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s)] \tag{24}$$

$$= (1 - 2\hat{u}_k)([m_f(\hat{s}') + \ln \gamma_k(\hat{s}', \hat{s}) + m_b(\hat{s})] - \max_{\substack{(s',s)\\ u_k=1-\hat{u}_k}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s)]) \tag{25}$$

となる．これより，

$$\Delta_k^* \equiv [m_f(\hat{s}') + \ln \gamma_k(\hat{s}', \hat{s}) + m_b(\hat{s})] - \max_{\substack{(s',s)\\ u_k=1-\hat{u}_k}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s)] \tag{26}$$

は，$\hat{u}_k$ の信頼度と見なされる．ただし，$s_{k-1} = \hat{s}'$ および $s_k = \hat{s}$ は ML パス上の状態を表す．右辺第 1 項は ML パスのメトリック，また，第 2 項は条件 $u_k \neq \hat{u}_k$ を満たす best パスのメトリックであり，両者のメトリック差により信頼度が近似される [10]．この方式が Max-Log-MAP である．

4.2 nonrecursive 符号に対する Max-Log-MAP

nonrecursive な符号（符号化率 $1/n$，メモリ長 ν）に対しては，Max-Log-MAP は更に簡単化される．トレリスの状態 $s_k = s$ が $s_k = (u_{k-\nu+1}, \cdots, u_{k-1}, u_k)$ と表現されることより，

$$L(\hat{u}_k) \approx \max_{\substack{(s',s)\\ u_k=0}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s) - \epsilon_s] - \max_{\substack{(s',s)\\ u_k=1}} [m_f(s') + \ln \gamma_k(s',s) + m_b(s) - \epsilon_s]$$

$$= \max_{\substack{s\\ u_k=0}} [m_f(s) + m_b(s) - \epsilon_s] - \max_{\substack{s\\ u_k=1}} [m_f(s) + m_b(s) - \epsilon_s] \tag{27}$$

$$= (1 - 2\hat{u}_k)([m_f(\hat{s}) + m_b(\hat{s}) - \epsilon_{\hat{s}}] - \max_{\substack{s\\ u_k=1-\hat{u}_k}} [m_f(s) + m_b(s) - \epsilon_s]) \tag{28}$$

$$\equiv (1 - 2\hat{u}_k) \cdot \Delta_k^* \tag{29}$$

となる．ただし，

$$\epsilon_s \equiv \ln P(s_k = s)$$

$$= \ln P(u_{k-\nu+1}) + \cdots + \ln P(u_{k-1}) + \ln P(u_k) \tag{30}$$

とする．また，$m_f(s)$ は，前向きの復号で得られる状態 $s_k = s$ に対する best パスのメトリックを表す．すなわち，前向きおよび後ろ向きの復号器が，$t = k$ で接合すれば，信頼度の近似値 Δ_k^* が得られる．

5 SOVAとMax-Log-MAPの関係

5.1 Δ_k^H と Δ_k^* の関係

$$\Delta_k^H \equiv \min_{0\le j\le\delta} \Delta_{k+j} = \Delta_{k+j'} \tag{31}$$

とし，$t=k+j'$ で ML パスと競合したパスを $x^{j'}$ と書く．SOVA の条件より，$u_k^{j'} \neq \hat{u}_k$ と仮定できる．ここで，Δ_k^* に現れる

$$\max_{\substack{(s',s)\\ u_k=1-\hat{u}_k}} [m_f(s') + \ln\gamma_k(s',s) + m_b(s) - \ln P(s_k=s)] \tag{32}$$

は，条件 $u_k \neq \hat{u}_k$ を満たすパスに対するメトリックの最大値を表すから，$x^{j'}$ に $t=k+j'$ 以降の ML パスを接合したパスのメトリックを $m(x^{j'})$ と書いたとき，不等式

$$m(x^{j'}) \le \max_{\substack{(s',s)\\ u_k=1-\hat{u}_k}} [m_f(s') + \ln\gamma_k(s',s) + m_b(s) - \ln P(s_k=s)] \tag{33}$$

が成り立つ．これより，一般に，関係

$$\Delta_k^* \le \Delta_k^H \tag{34}$$

が得られる．

5.2 修正 SOVA（Battail-SOVA）

Δ_k^* を達成する，ML パスと対をなすパスを x^* とする（$u_k^* \neq \hat{u}_k$ が成り立つ）．また，x^* は $t=k+l$ で ML パスに合流しているとする．問題は，$t=k+l$ で ML パスと競合したパス x^l が x^* に一致しない場合である．そこで，競合パス x^l に注目する（メトリック差を Δ とする）．VA の条件より，$u_k^l = \hat{u}_k$ と仮定できる．さて，x^* は VA の途中で（状態 $\tilde{s}$ とする）捨て去られている．ここで，$\tilde{s}$ に対する生き残りパス（x^l の一部）に注目しよう．この生き残りパスに基づく $\hat{u}_k$ の信頼度を $L_k^{(2)}$ とすれば，

$$L_k^{(2)} \le \Delta_k^* - \Delta \tag{35}$$

が成り立つ．すなわち，$L_k^{(2)} + \Delta \le \Delta_k^*$ が従う．また，ML パスに基づく $\hat{u}_k$ の信頼度を $\hat{L}_k$ とすれば，Δ_k^* の定義より，$\Delta_k^* \le \hat{L}_k$ が成り立つから，

$$\min(L_k^{(2)} + \Delta, \hat{L}_k) = L_k^{(2)} + \Delta \le \Delta_k^* \tag{36}$$

が得られる．一方，$L_k^{(2)}$ を達成する，$\tilde{s}$ に対する生き残りパスに合流しているパス $\tilde{x}$ に注目すると，ML パスとのメトリック差は $L_k^{(2)} + \Delta$ であり，$\tilde{u}_k \neq \hat{u}_k$ が成り立つ．従って，Δ_k^* の定義より，

$$L_k^{(2)} + \Delta \ge \Delta_k^* \tag{37}$$

が成り立ち，$\hat{L}_k \ge \Delta_k^*$ と合わせると，

$$\min(L_k^{(2)} + \Delta, \hat{L}_k) \ge \Delta_k^* \tag{38}$$

が得られる．この結果，(36) および (38) より，等式

$$\min(L_k^{(2)} + \Delta, \hat{L}_k) = \Delta_k^* \tag{39}$$

が導かれる．修正 SOVA(Battail-SOVA) [19] では，$t=k+j(0\le j\le\delta)$ における ML パスとの競合パス x^j に対し（メトリック差を Δ_{k+j} とする），条件 $u_k^j = \hat{u}_k$ を満たす場合でも，

$$\hat{L}_k \leftarrow \min(L_k^{(2)} + \Delta_{k+j}, \hat{L}_k) \tag{40}$$

により，信頼度の更新を行う．ここで，$L_k^{(2)}$ は，競合パスに基づく $\hat{u}_k$ の信頼度を表す．従って，Battail-SOVA は Max-Log-MAP に等価なアルゴリズムであることが分かる [20]．

6 むすび

BCJR アルゴリズムを近似する方式の内，SOVA と Max-Log-MAP を取り上げ，それらの導出を行った．これらの方式は，順方向の VA および双方向の VA と非常に整合しているという特徴を持つ．特に，Max-Log-MAP について言えば，単に復号だけの目的ならば，順方向に VA を一回実行すればよいが，更に，復号シンボルの信頼度を求めたい場合は，逆方向にもう一回 VA を実行すればよいことを示している．このような理解の仕方は非常に分かりやすい．また，SOVA は，ML パスと競合パスとのメトリック差を保存していくことから，リストビタビ復号 [24] やラベル付きビタビ復号 [25] とも密接に関連していることが示される [26], [27].

参考文献

[1] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv,"Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 2, pp. 284–287, March 1974.

[3] J. Hagenauer, E. Offer, and L. Papke,"Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, March 1996.

[4] X. Wang and S. B. Wicker,"A soft-output decoding algorithm for concatenated systems," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 543–553, March 1996.

[5] P. Jung,"Comparison of turbo-code decoders applied to short frame transmission systems," *IEEE J. Selected Areas in Commun.*, vol. 14, no. 3, pp. 530–537, April 1996.

[6] M. B. Shoemake and C. Heegard,"Computationally efficient turbo decoding with the bi-directional Viterbi algorithm (BIVA)," *in Proc. ISIT'97*, p. 228, June/July 1997.

[7] V. Franz and J. B. Anderson,"Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Selected Areas in Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[8] A. J. Viterbi,"An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Selected Areas in Commun.*, vol. 16, no. 2, pp. 260–264, Feb. 1998.

[9] J. Hagenauer and P. Hoeher,"A Viterbi algorithm with soft-decision outputs and its applications," *in Proc. GLOBECOM'89*, pp.1680–1686, Nov. 1989.

[10] P. Robertson, E. Villebrun, and P. Hoeher,"A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," *in Proc. ICC'95*, pp. 1009–1013, June 1995.

[11] W. Koch and A. Baier,"Optimum and sub-optimum detection of coded data disturbed by time-varying inter-symbol interference," *in Proc. GLOBECOM'90*, pp. 1679–1684, Dec. 1990.

[12] J. Erfanian, S. Pasupathy, and G. Gulak,"Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1661–1671, Feb./March/April 1994.

[13] C. Berrou and A. Glavieux,"Reflections on the prize paper : "Near optimum error-correcting coding and decoding : turbo codes"," *IEEE IT Newsletter*, vol. 48, no. 2, June 1998.

[14] C. Berrou, P. Adde, E. Angui, and S. Faudeil,"A low complexity soft-output Viterbi decoder architecture," *in Proc. ICC'93*, pp. 737–740, May 1993.

[15] C. Berrou, A. Glavieux, and P. Thitimajshima,"Near Shannon limit error-correcting coding and decoding : Turbo-codes," *in Proc. ICC'93*, pp. 1064–1070, May 1993.

[16] J. Hagenauer, E. Offer, and L. Papke,"Matching Viterbi decoders and Reed-Solomon decoders in a concatenated system," *in Reed-Solomon Codes and Their Applications*, S. B. Wicker and V. K. Bhargava, Eds., IEEE Press, pp. 242–271, 1994.

[17] J. Hagenauer,"Source-controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, no. 9, pp. 2449–2457, Sept. 1995.

[18] M. Tajima, K. Takida, and Z. Kawasaki,"Comments on simplification of the BCJR algorithm using the bidirectional Viterbi algorithm," *IEICE Trans. Fundamentals*, vol. E82-A, no. 10, pp. 2306–2310, Oct. 1999.

[19] L. Lin and R. S. Cheng,"Improvements in SOVA-based decoding for turbo codes," *in Proc. ICC'97*, pp. 1473–1478, June 1997.

[20] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer,"On the equivalence between SOVA and Max-Log-MAP decodings," *IEEE Commun. Letters*, vol. 2, no. 5, pp. 137–139, May 1998.

[21] G. D. Forney, Jr., "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Trans. Inform. Theory*, vol. IT-14, no. 2, pp. 206–220, March 1968.

[22] T. Hashimoto,"A list-type reduced-constraint generalization of the Viterbi algorithm," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 6, pp. 866–876, Nov. 1987.

[23] T. Hashimoto,"On the error exponent of convolutionally coded ARQ," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 567–575, March 1994.

[24] N. Seshadri and C-E. W. Sundberg,"List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 313–323, Feb./March/April 1994.

[25] H. Yamamoto and K. Itoh,"Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 5, pp. 540–547, Sept. 1980.

[26] C. Nill and C-E. W. Sundberg,"List and soft symbol output Viterbi algorithms : Extensions and comparisons," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 277–287, Feb./March/April 1995.

[27] M. Tajima, K. Takida, and Z. Kawasaki,"On the relation between Viterbi decoding with labels and the SOVA," *IEICE Trans. Fundamentals*, vol. E83-A, no. 10, pp. 1966–1970, Oct. 2000.

[28] B. Sklar,"A primer on turbo code concepts," *IEEE Commun. Magazine*, vol. 35, no. 12, pp. 94–101, Dec. 1997.

[29] M. Isaka and H. Imai,"A tutorial on "parallel concatenated (Turbo) coding", "Turbo (iterative) decoding" and related topics," *Technical Report of IEICE*, IT98-51, pp. 1–18, Dec. 1998.

[30] C. Heegard and S. B. Wicker,"Turbo coding," Kluwer Academic Publishers, 1999.

ターボ符号化変調

荻原 春生（長岡技術科学大学）

1　まえがき

2 値信号を通信路に送出する Berrou らのターボ符号（以下，2 値ターボ符号と呼ぶ）の発表 [1] の後，それを多値変調へ拡張し，周波数帯域あたりの伝送情報量を増大しようとするターボ符号化変調の試みが各種なされている．本稿では，それらの符号器・復号器の基本構成と復号特性について概説する．

2　2 値ターボ符号化+多値変調 [2]

ターボ符号化変調の最初の提案は Goff らによるもである [2]．彼らは，2 値ターボ符号の出力を，インタリーバで並べ替えた後，数ビット毎にブロック化し，ブロック毎に，多値シンボルに写像する符号器を提案した．受信器では，時点 k の受信シンボル X_k から，その時点の各ビット $u_{k\,i}$ の対数ゆう度比で定義される通信路値 L_{ch} を

$$\begin{aligned} L_{ch}(u_{k\,i}) &= \log \frac{Pr(X_k/u_{k\,i}=1)}{Pr(X_k/u_{k\,i}=0)} \\ &= \log \frac{\sum_{\boldsymbol{u}_{k\,j\neq i}} Pr(X_k, Y_k/u_{k\,i}=1, \boldsymbol{u}_{k\,j})}{\sum_{\boldsymbol{u}_{k\,j\neq i}} Pr(X_k, Y_k/u_{k\,i}=0, \boldsymbol{u}_{k\,j})} \end{aligned} \qquad (1)$$

で求める，ただし，$\boldsymbol{u}_{k\,j\,j\neq i}$ は，u_{ki} を除く時刻 k における全てのビットを意味する．ビット毎の通信路値を求めた後，それらをディインタリーブし，通常の 2 値ターボ符号の復号処理を実行する．ブロック内のビットは，同じ雑音を受けるので，通信路値は統計的に独立ではない．2 値ターボ符号の復号では，これらを独立として扱うので，この近似が成立するように，ディインタリーブの処理が設けられている．

上記計算をそのまま実行するには，雑音の確率密度関数の形と，信号対雑音比（SNR) の情報が必要である．加法的白色ガウス雑音通信路では，総和の演算を最大値演算で近似すると，SNR の情報あるいは対数演算が不要となり簡易化される．これらは，2 値ターボ符号の復号で，BCJR アルゴリズムの代わりに MAX log MAP アルゴリズムを用いることによる処理の簡易化と同様の関係である．

復号特性は表 1 のように報告されている [2]．通常の誤り訂正符号の世界では，2 値符号化後，それをブロック化して，単純に多値シンボルに写像しても良い特性が得られないとうことで，多値シンボルの信号点間のユークリッド距離を考慮した符号化である符号化変調が生まれたが，Goff らの方法は，これらの考慮なしでも，ターボ符号では優れた特性を実現できることを示している．

表 1:　2 値ターボ符号化+多値変調の符号化利得 at BER=10^{-6} over AWGN.

turbo code rate	1/2	2/3	3/4
modulation	16-QAM	8-PSK	16-QAM
spectral efficiency (bit/Hz)	2	2	3
coding gain over uncoded	6.0dB	5.5dB	7.8dB
coding gain over 64 sate TCM	2.4dB	1.9dB	2.6dB

状態符号，64x64 の疑似ランダムインタリーバ，繰り返し 3 回 [2]

3　ビットインタリーブを用いた並列連接 TCM[3]

図 1 は，[3] で提案された並列連接 TCM の符号器である．第 1 の要素符号器に $2m$ ビットが同時に入力されパリティビットが 1 ビットつくられる．同時に入力された情報ビットの半分の m ビットとパリティビットの組が多値変調シンボルに写像され送信される．また，$2m$ ビットの情報ビットは，それぞれ独立なビットインタリーバを通過して，第 2 の要素符号器に入力され，つくられたパリティビットと，第 1 の要素符号器では，多値シンボルへの写像に使われなかった残りの m ビットと組み合わされ，多値変調シンボルに写像され送信される．

復号器では，一つの多値シンボルに写像された時点 k のビット u_{ki} の復号器 1 による外部値 $L_1(u_{kj})$ と通信路値 $L_{ch}(u_{ki})$(外部値は復号器 2 への事前値 $L_1(u_{kj})$ として使われる）の和を，時点 k の受信信号 X_k から

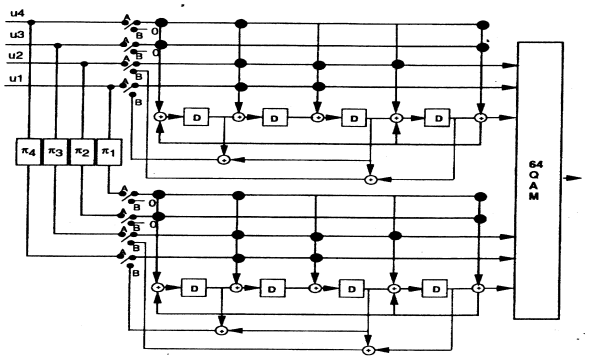


図 1: ビットインタリーブを用いた並列連接 TCM 符号器. 符号化率 $R = 2m/(2m+2)$[3]

$$
\begin{aligned}
&L_1(u_{kj}) + L_{ch}(u_{ki}) \\
&= \log\{\frac{\sum_{\boldsymbol{u}:u_{ki}=1} Pr(\boldsymbol{X}_1/\boldsymbol{u})}{\sum_{\boldsymbol{u}:u_{ki}=0} Pr(\boldsymbol{X}_1/\boldsymbol{u})} \\
&\quad \cdot \frac{\prod_n \prod_{j,(n,j)\neq(k,i)} \exp\{u_{nj}L_2(u_{nj})/2\}}{\prod_n \prod_{j,(n,j)\neq(k,i)} \exp\{u_{nj}L_2(u_{nj})/2\}}\} \quad (2)
\end{aligned}
$$

で計算する (原論分では，上式を外部値とし，分母・分子の最後の因子が $\exp\{u_{nj}L_2(u_{nj})\}$ となっているが，誤記と思われ修正して引用した)．ここで，$L_2(u_{nj})$ は，もう一方の要素復号器から送られてきた，u_{nj} についての事前値である．事前値は 1 で有る確率と 0 で有る確率の比の対数で定義される．

これにより，図 2, 図 3 に示す特性が報告されている [3].

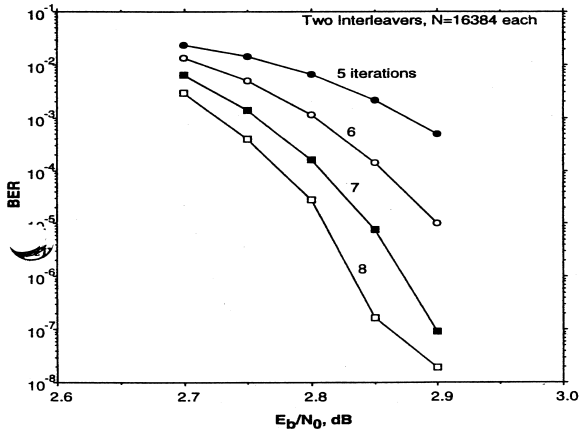


図 2: ビットインタリーブを用いた並列連接 TCM の特性. 16QAM. 2bit/sec/Hz.[3]

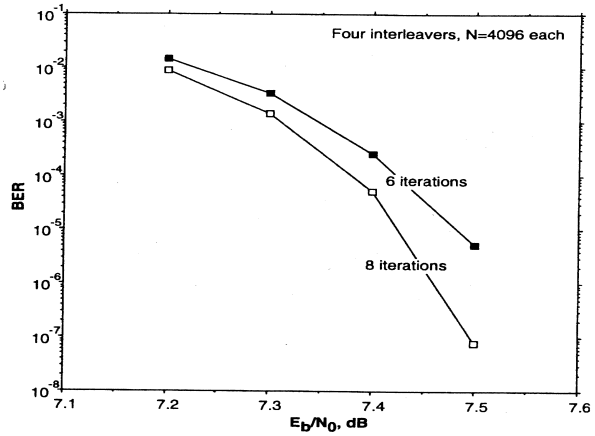


図 3: ビットインタリーブを用いた並列連接 TCM の特性. 64QAM　4bit/sec/Hz.[3]

4 シンボルインタリーブを用いた並列連接 TCM[4][5]

図 4 に [4][5] で提案されたシンボルインタリーブを用いた並列連接 TCM の符号器を示す．ここで，シンボルインタリーブとは，要素符号器に同時に入力されるビットの組を保存して並べ変えるインタリーバを意味する．それぞれの要素符号器は，通常の Ungerboeck 形の TCM 符号器である．第 2 の要素符号器からの出力はディインタリーバで元の順にもどされ，第 1 の要素符号器出力と交互に送出される．ここで，インタリーバによる並べ変えは，偶数番目のシンボルは偶数番目に，奇数番目のシンボルは奇数番目という制限をつけた odd-even インタリーバによる．この条件は，どの情報シンボルも，必ず 1 度だけ多値シンボルに写像されて送出されるために設けられている．

この構成では，インタリーバとディインタリーバが縦続接続されるので，符号化遅延が大きくなる．これを避ける図 5 の構成が提案されている [6]. また，この構成では，インタリーバから odd-even の制限をはずせる．

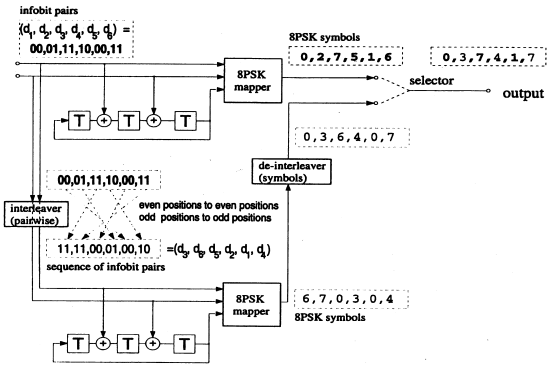


図 4: Ugerboeck 符号器の並列連接.[5]

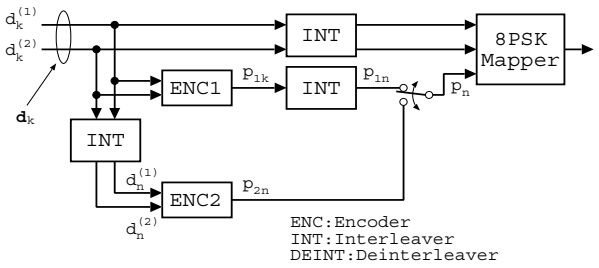


図 5: 符号化遅延の低減，odd-even 制限の除去.[6]

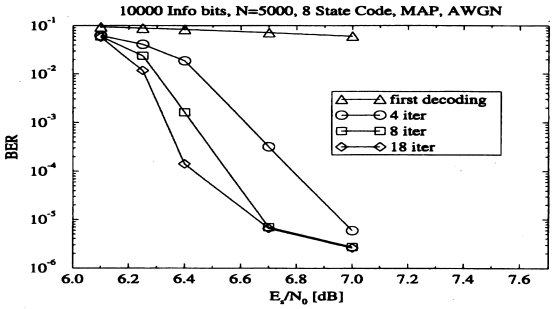


図 6: シンボルインタリーブを用いた並列連接 TCM の特性．8PSK, 2bit/sec/Hz.[5]

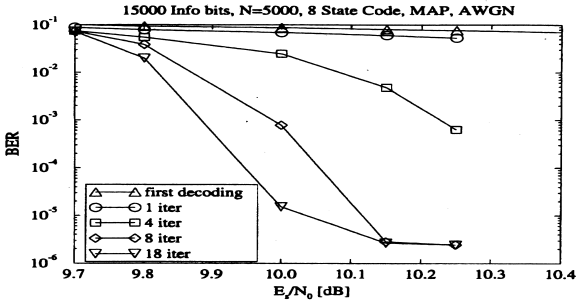


図 7: シンボルインタリーブを用いた並列連接 TCM の特性．16QAM, 3bit/sec/Hz.[5]

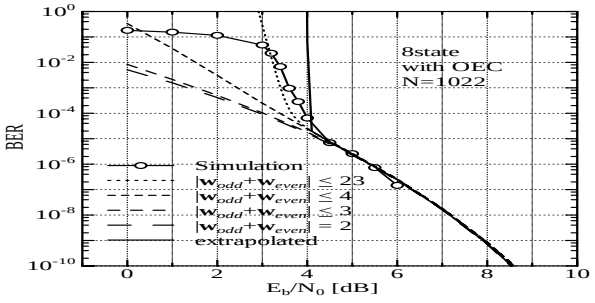


図 8: シンボルインタリーブを用いた並列連接 TCM の上界値とシミュレーション結果 [7]

復号は，基準シンボル（例えば，全零のビットの組に対応したシンボル）の事後確率に対する注目シンボルの事後確率の比の対数で定義でされる事後値 $\Lambda(d_k = i)$ を

$$\Lambda(d_k = i) = \ln \frac{\sum_m \sum_{m'} \gamma_i(R_k, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_{00}(R_k, m', m)\alpha_{k-1}(m')\beta_k(m)} \tag{3}$$

により計算することにより行われる．ここで，$\alpha_{k-1}(m')$ は，始点から時点 $k-1$ で状態 m' に至るパスの確率の総和，$\beta_k(m)$ は，時点 k で状態 m から出発し，終点に至るパスの確率の総和であり，再帰的に計算される．$\gamma_i(R_k, m', m)$ は，時点 k の状態 m' から m への遷移が情報シンボル $d_k = i$ の入力で起こり，受信シンボル R_k が得られる確率である．

図 6, 図 7 に復号特性を示す [5].

このシステムの特性解析が，２値ターボ符号の場合を，TCM の場合に拡張して発表されている [7]．解析結果と，シミュレーション結果を図 8 に示す．E_b/N_0 が大きい領域では，両者が一致していることがわかる．ここで，解析は全てのインタリーバでの平均特性の上界値を，シミュレーションは疑似ランダムインタリーバでの特性である．さらに，この領域では，シンボル重み２の情報誤り系列がビット誤り率に支配的影響を与えることが示されている [7].

5 ビットインタリーブを用いた直列連接 TCM[9]

m ビットを同時に入力し $m+1$ ビットを同時に出力する最初の符号器（外符号器）の出力の各ビットを独立なインタリーバに入力し，その出力を同時に，１ビットの遅延器よりなる第２の畳み込み符号器（内符号器）に入力し，$m+2$ ビットを生成し，それを多値シンボルに写像して送信する直列連接 TCM が発表されている [9]．内符号の復号では，式 (2) でビット毎の外部値＋通信路値を計算し，それを外符号の復号器に送り，ビットごとの復号を行って，事後値と外部値を求めている．外部値は内符号の復号器に送られ，繰り返し復号が行われる．

外符号は最小距離が最大となるよう，内符号は重み２の入力と重み０の入力に対する出力のユークリッド距離が最大になるよう構成されている．特性を図 9，図 10 に示す.

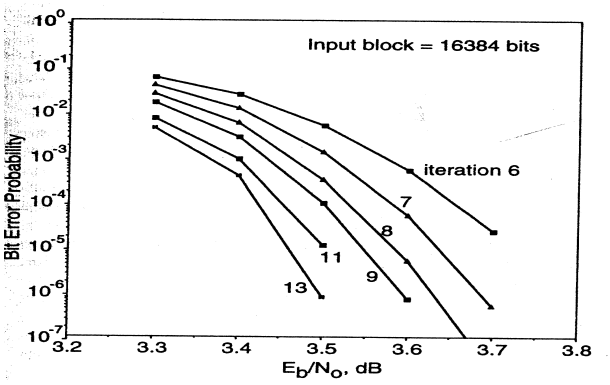


図 9: ビットインタリーブを用いた直列連接 TCM の特性. 8-state outer code, 2x8PSK, 2bit/sec/Hz.[9]

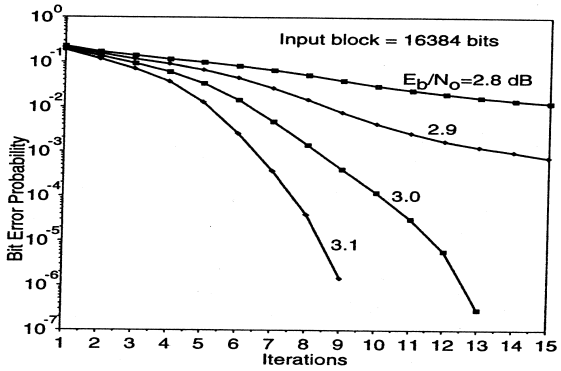


図 10: ビットインタリーブを用いた直列連接 TCM の特性. 16-state outer code, 16QAM, 2bit/sec/Hz.[9]

6 シンボルインタリーブを用いた直列連接 TCM[10]

図 11 に [11] で提案されたシンボルインタリーブを用いた直列連接 TCM の符号器を示す. 外符号器は, m ビットの情報を受け $m+1$ ビットを出力する. 内符号器は, それを組として並べ替えた系列を受け取り, $m+1$ の符号ベクトルを出力し, 2^{m+1} 個の信号点のひとつに写像する. 式 (3) と同様な計算により, 事後値・外部値を求め繰り返し復号を行う.

特性を図 12, 図 13 に示す. 比較のため, 同じ畳み込み符号を要素とした並列連接 TCM の特性も併せて示す. E_b/N_0 を増大したとき, 並列連接の方がすぐに誤り率が低下するが, ある程度まで下がると, それ以上低下しにくい傾向がみえる. 一方, 直列連接では, 誤り率がさがりはじめるのは遅いが, それ以降は急速に低下する. また, 並列連接では, ビット誤り率はおおよそインタリーバサイズに逆比例するが, 直列連接では, インタリーバサイズが 2 倍になる毎に, およそ 1 桁減少している.

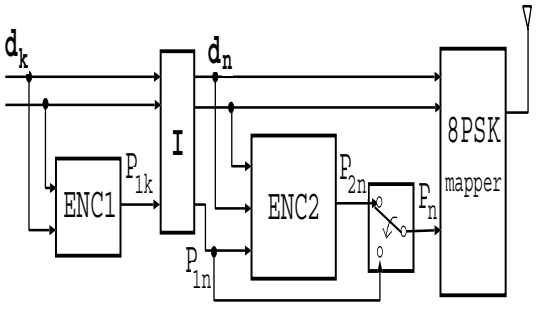


図 11: シンボルインタリーブを用いた符号化率 m/(m+1) の直列連接 TCM 符号器.[11]

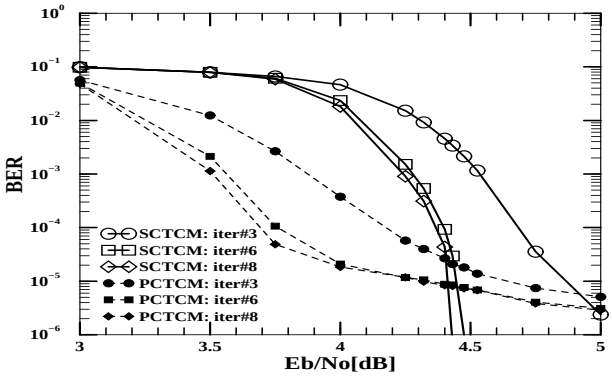


図 12: シンボルインタリーブを用いた直列連接 TCM と並列連接 TCM の特性比較.[11]

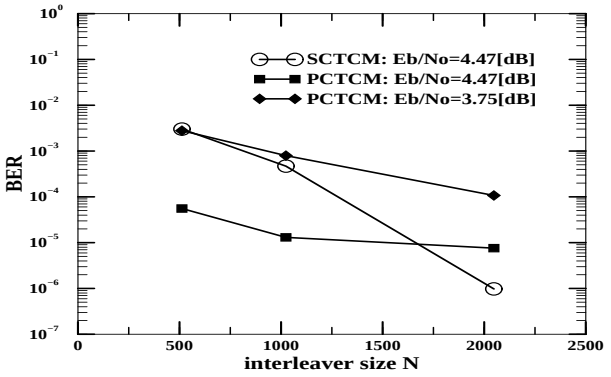


図 13: シンボルインタリーブを用いた直列連接 TCM の特性. インタリーバサイズ依存性.[11]

7 積符号と多値変調の組み合わせ [12]

文献 [12] は, 2 値の積符号の幾つかのビットごとに, 多値のシンボルに写像する符号を提案している. 受信器では, 式 (1) により, 通信路値を計算し, それを積符号の繰り返し復号器に入力している. その結果, 図 14 に示す結果が得られたと報告されている.

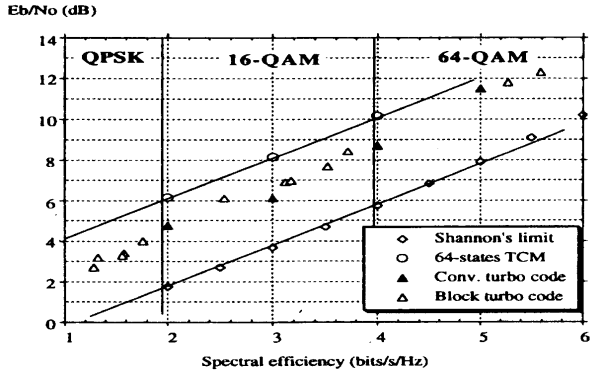


図 14: 長さ 64 から 128 の BCH 符号による積符号の繰り返しチェイス復号.[12]. 2 値化通信路値は式 (1)

表 2: マルチレベル符号化の性能.[14]

Modulation	Coding Rate [bits/Symbol]	Block length bits	E_b/N_0[dB] at $BER = 10^{-5}$	Distance to Shannon limit [dB]
8PSK	2	2000	4.0	1.3
	2	20000	3.4	0.7
16QAM	3	2000	5.8	1.3
	3	20000	5.3	0.8
32QAM	4	2000	7.8	1.2
	4	20000	7.2	0.6
64QAM	5	2000	10.5	1.3

8 マルチレベル符号化と繰り返し復号の組み合わせ

リードソロモン符号とマルチレベル符号の直列連接系 [13] および，ターボ符号を各レベルに使用したマルチレベル符号 [14] も提案されている．後者では 表 2 に示す特性が報告されている．ここでは，通信路値を，下位レベルの復号結果で条件付けた確率を用いて，式 (1) と類似の式で求めている．

参考文献

[1] C. Berrou, A. Glavieux and P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: turbo-codes, Proc. ICC'93, pp.1064-1070 (1993).

[2] Stephane Le Goff, Alain Glavieux and Claude Berrou, Turbo-code and high specral efficiency modulation, Proc. ICC94, pp.645–649 (1994).

[3] S. Benedetto, D. Divsalar, G. Montorsi , F. Pollara, Parallel Concatenated Trellis Coded Modulation, Proc. ICC'96, pp.974–978 (1996).

[4] P. Robertson and T. Woerz, Coded modulation scheme employing turbo codes, Electron. Letter, 31, pp. 1546–1547 (1995).

[5] P. Robertson and T. Woerz, Bandwidth-efficient turbo trellis-coded modulation using puncured component codes, IEEE J. Sel. Comm., 16, pp.206–218 (1998).

[6] H. Ogiwara and M. Yano, Improvement of turbo trellis-coded modulation, IEICE Trans. Fundamentals, E81-A, pp.2040–2046 (1998).

[7] H. Ogiwara, A. Mizutome and K. Koike, Performance evaluation of turbo TCM, Proc. 2nd Int. Symp. on Turbo codes and Related Topics, pp.197–200 (2000).

[8] D. Divsalar and F. Pollara, turbo codes for PCS applications, Proc. ICC'95, pp.54–59 (1995).

[9] S Benedetto, D Drvsalar, G. Montorsi and F. Pollara, Serial Concatenated Trellis Coded Modulatron wrth Iterative Decoding: Design and Performance, Proc. Gobecom'97, pp.38–43 (1997).

[10] H. Ogiwara and B. Bajo, Iterative decoding of serially concatenated punctured trellis-coded modulation, Proc. ICC'99, pp.1167–1171 (1999).

[11] H. Ogiwara and B. Bajo, Iterative decoding of serially concatenated punctured trellis-coded modulation, IEICE Trans. Fundamentals, E82-A, pp.2199–2203 (1999).

[12] R. Pyndian, A. Picart and A.n Glavieux, Performance of block turbo coded 16-QAM and 64-QAM modulations, Proc. Gobecom'95, pp.1039–1043 (1995).

[13] K. Fazel, L. Papke, Combined multilevel Turbo-code with 8PSK modulation Proc. Gobecom'95, pp.649–653 (1995).

[14] U. Wachsmann and J. Huber, Power and bandwidth efficient digital communication using turbo code in multilevel codes, European Trans. Telecommnincation, 6, pp.557–567 (1995).

低密度パリティ検査符号とその復号法

和田山 正（岡山県立大学）

1 はじめに

多項式時間の計算量で実行可能なシャノン限界に漸近する誤り訂正符号化／復号アルゴリズムの開発は符号理論の中心的課題であり、この 50 年間、その目標に向かった研究が数多くなされてきた。近年のターボ符号・復号法の発見、ならびに低密度パリティ検査符号（以下、LDPC 符号と略する）、sum-product アルゴリズムの再評価は上記の目標に迫るための重要なブレークスルーをもたらした。

ターボ符号、LDPC 符号を特徴づける最も顕著な点として、これらの符号が（擬似）ランダムに構成される点が挙げられる。通信路符号化定理により保証されているように、符号長が十分に長い場合、ランダム符号アンサンブルから選び出された符号は、ほぼ確実に優れた最尤復号性能を持つ。しかし、ランダム符号は効率の良い復号を助ける構造を持たないため、それを復号するには符号長に対して指数時間の計算時間が必要とされる。ターボ符号、LDPC 符号などの検査行列が低密度である符号は、優れた最尤復号特性を保証するランダム性を保ちつつ、復号に適した構造、すなわち ”疎なグラフ構造 ”を兼ね備えている。

一方、ターボ復号法、sum-product アルゴリズムは、大規模なグラフィカル確率モデル上における確率伝播アルゴリズムであり、隠れ確率変数（送信情報シンボル）の事後確率分布を観測値（受信信号系列）から計算する手法である。グラフィカルモデル（例えばベイジアンネットワーク、マルコフランダム場など）上での確率伝播アルゴリズムは、ベイズ的手法が利用される分野（例えば人工知能、パターン認識、画像処理、音声認識、DNA 解析など）で広く利用されている。

このようにターボ符号、LDPC 符号などの疎なグラフにより定義される符号に関する符号化技術は、大数の法則に基づく技術（ランダム符号）とベイズ的手法（グライフィカルモデル上での確率伝播アルゴリズム）との融合により生まれた新しい技術であり、従来の符号化技術とは一線を画す部分を持つ。

本稿では、LDPC 符号とその復号法に焦点を当てて、疎なグラフにより定義される符号に関する符号化・復号技術の原理について述べる[1]。

2 LDPC 符号

LDPC 符号は Gallager により、1960 年代初頭に提案された [1]。彼は、さらに符号アンサンブルに基づく最尤復号性能の解析、sum-product アルゴリズムの提案も行っている。しかし、この先駆的な業績は、90 年代の LDPC 符号の再評価に至る 30 数年間に渡り、いくつかの論文（例えば Tanner[2]）で関連する議論が展開された以外はほぼ忘れられた存在となっていた。その理由として、当時はコンピュータ資源が高価であり、LDPC 符号の性能が特に顕著に発揮される符号長が十分に長い場合に関するシミュレーションが困難であったことが考えられる[2]。また、その当時（60 年代から 70 年代にかけて）は、新しい符号と復号法、例えば、連接符号、代数的符号 (BCH 符号、Reed-Solomon 符号) とその復号法（Berlekamp-Massy, Euclid 復号法）、畳み込み符号、Viterbi アルゴリズムなどの登場ラッシュが続いていたため、その影に隠れてしまったというのも一因だろう。90 年代中頃の MacKay の仕事 [3] は、LDPC 符号の再評価の契機となった。彼は、最尤復号器を利用するという仮定のもとで、広いクラスの通信路モデルにおいて、LDPC 符号の符号系列の符号化率が通信路容量を達成することを示した。また、LDPC 符号と sum-product アルゴリズムの組み合わせに関する幅広いシミュレーション結果を示し、その優れた性能を明らかにした。

LDPC 符号と sum-product アルゴリズムの組み合わせにより得られる復号特性は、ターボ符号、RA 符号 (Repeat and Accumulate 符号) の性能に匹敵、もしくは（符号長が非常に長い場合には）それらを上回る。例えば、Richard-

[1]紙幅の都合上、本稿では最近の研究動向には触れていない。IEEE Trans.Inform.Theory, 2001 年 3 月号の特集 Graph and iterative algorithms が最近の研究動向を知るのに適している。

[2]それでも、Gallager は当時、符号長 1000 程度までのシュミレーションは実際に行っている。

[3]実験では、イレギュラー LDPC 符号 [4] が用いられた。この結果は、符号長 100 万の符号を現実的な時間で復号できるということを示した点でも重要である。

son らの実験 [4] によると符号長 100 万の LDPC 符号により、白色ガウス通信路のシャノン限界から 0.13dB(ビット誤り率 10^{-6} において) という復号特性が得られている[3]。

LDPC 符号は、非常に疎な検査行列により定義される線形符号である（本稿では、2 元の LDPC 符号のみについて議論する）。疎な行列とは、行列内の非零要素の数が非常に少ない行列を指す。疎な検査行列を構成する手法はいくつか知られているが、ここでは Gallager による検査行列構成法について説明する [1]。まず、検査行列を図 1 に示すようにいくつかの部分行列に分ける。ここで、部分行列数を j とする。図 1 の場合、検査行列は 3 つの部分行列 ($j = 3$) に分割されている。

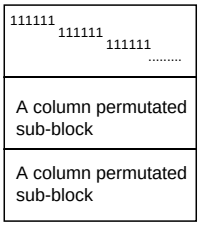


図 1: Gallager による疎な検査行列の構成法（行重み 6, 列重み 3）

第 1 部分行列の第 1 行目は、連続する k 個の 1（図 1 の場合、$k = 6$）とそれに続く $N - k$ 個の 0 から成る。ここで N は符号長である。第 2 行目は、第 1 行目を k シンボル右に巡回置換したものとなる。同様に第 3 行目以降が与えられ、第 1 部分行列は図 1 に示される形となる。そして、第 $2, \ldots, j$ 部分行列は第 1 部分行列に列置換を施した部分行列となる。ただし、第 2 部分行列から第 j 部分行列を決める列置換はそれぞれ独立にランダムに選択されると仮定する。この構成法により、図 1 の場合、検査行列の各行の重みは 6、各列の重みは 3 になることが容易にわかる。このような検査行列の各行、各列の重みが一定の LDPC 符号をレギュラー LDPC 符号と呼ぶ。

雑音のある通信路において、そのシャノン限界に接近するためには、良い符号クラスが必要である。良い符号クラスの条件として、(1) 符号長の長い符号を容易に構成できること、(2) 優れた最尤復号特性を持つこと、が挙げられる。例えば、2 元対称通信路において、ランダムに生成された線形符号（例えば、コイン投げにより検査行列を定めて作る）はこの良い符号の条件を満足する。LDPC 符号の場合も、上記の構成法により任意の符号長において符号を容易に構成できる。また、サブブロックの置換を乱数に基づいて選択することにより、ランダム符号に近い最尤復号性能が期待できることが知られている [1][3]。

3 Sum-product アルゴリズム

LDPC 符号の復号法として利用される sum-product アルゴリズムは符号の復号アルゴリズムとしてだけではなく、デジタル通信、信号処理、人工知能などの分野においてさまざまな形で利用されている。例えば、隠れマルコフモデルに対する前向き後ろ向きアルゴリズム（BCJR アルゴリズム）、Viterbi アルゴリズム、ターボ復号法、Belief propagation、カルマンフィルタなどが挙げられる [5][6]。このように様々な一見異なるアルゴリズムが同じアルゴリズム原理に基づいているということが明らかになってきたのは比較的最近のことである。

Gallager は彼の博士論文 [1] の中で, sum-product アルゴリズムを LDPC 符号の復号法として完全な形で述べている。Tanner は線形符号を定義するグラフとその復号法に関する議論を展開し、今日、タナーグラフと呼ばれる符号のグラフに関する基礎を築いた [2]。Wiberg は、タナーグラフに状態ノードを追加することを考案した [7]。このアイデアにより、線形符号の最小トレリス、畳み込み符号のトレリスに基づく BCJR アルゴリズムと Viterbi アルゴリズムが sum-product アルゴリズムの一例であることが明確になった。また、同時に彼は、sum-product アルゴリズムの抽象化を行った。Aji と McEliece は、Wiberg のアイデアを発展させるとともに、sum-product アルゴリズムが、因子分解される目的関数の計算を分配則（一般化分配則と彼らは呼んでいる）に基づいて効率的に行うアルゴリズムであること明らかにした [5]。Kschishang と Frey は因子グラフ [6] を導入し、目的関数の因子分解をグラフ化することにより sum-product アルゴリズムが見通しよく理解できることを示した。彼らは、さらに、基礎となるグラフィカルモデルを変えることにより、”sum-product アルゴリズム原理” から FFT など数多くの既知のアルゴリズムが自然に導かれることを明らかにした。

以下では、条件付確率分布 $P(Y|X)$ により定義された通信路を考える。ここで、X は符号 C の符号語に相当する確率変数、Y は受信語に相当する確率変数とする。受信者が通信路からの出力 Y を観測したとき、受信者は送信符号語に関する事後分布 $P(X|Y)$ をベイズルール $P(X|Y) = P(Y|X)P(X)/P(Y)$ により計算できる。ここで $P(X)$ は X の事前確率分布である。この分布に従って送信符号語を推定することにより、推定誤り率が最小となる送信符号語の推定が可能となる。

シンボルごとの最大事後確率 (MAP) 推定の場合は、第 i シンボルの対数事後確率比

$$L_i = \log \frac{\sum_{X \in C, X_i = 0} P(X|Y)}{\sum_{X \in C, X_i = 1} P(X|Y)} \tag{1}$$

を計算し、$L_i \geq 0$ ならば第 i シンボルは 0 と推定し、$L_i < 0$ ならば第 i シンボルは 1 と推定する。ここで、$X = (X_1, \ldots, X_N)$ と仮定している。式 (1) は符号語全てに関する和を含むため、全ての符号語を生成する単純なアルゴリズムを利用すると符号語数に比例（すなわち、符号長について指数時間）する計算時間が式 (1) を評価するために必要となる。このため、符号の構造を利用した高速アルゴリズムが必要となる。

やや人工的な例ではあるが、sum-product アルゴリズムの仕組みを述べる前に簡単な例を示す。今、8 つの変数

a, b, c, d, e, f, g, hの値が与えられており、積和 $acg+adh+beg+bfh$ の値を計算したいものとしよう。単純にこれを計算するならば、乗算 8 回、加算 3 回の計 11 回の算術演算が必要となる。図 2 に、この積和計算の構造をグラフ化したものを示す。始点 A から始まり、終点 F で終わる全てのパス系列 (そのパスに含まれる枝ラベル変数の積) が上の積和計算の各項として現れていることがわかる。次に、この図の左端から $B = a, C = b, D = Bc + Ce, E = Bd + Cf, F = Dg + Eh$ と計算を進めたとしよう。この場合、乗算 6 回、加算 3 回の計 9 回の算術演算により目的とする積和計算ができる。すなわち、同じ積和がより少ない算術計算のもとに得られたことになる。この計算量の削減は、分配則 $acg+adh+beg+bfh = g(ac+be)+h(ad+bf)$ の利用により得られたものである。すなわち、図 2 のグラフは目的とする積和の因子分解の構造を表しており、このグラフに基づき積和演算を再帰的に計算していくことにより分配則を生かした効率の良い計算が可能となる。

式 (1) に現れる和も上の積和計算と同種の構造を有している。その場合、acg, adh などの項は各符号語に対する事後確率に対応する。もし、符号が図 2 に示されるようなノード数の少ないグラフ構造（トレリス構造と呼ばれる）を持つならば、事後確率の和が上に述べた再帰的な分配則の適用により効率良く計算できる[4]。隠れマルコフモデルにおける隠れ変数の事後確率計算、畳み込み符号のシンボルワイズ MAP 復号などで利用される前向き・後ろ向きアルゴリズム (BCJR アルゴリズム) では、上述の計算を右方向からも行い、それぞれの計算結果から各シンボルの事後確率分布を計算する。

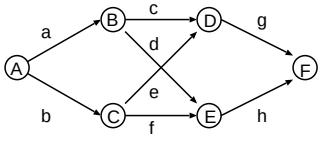


図 2: 積和 $acg+adh+beg+bfh$ の因子分解を表すグラフ

上述の計算手法は、より広いクラスのグラフに適用できる。今、5 つの離散値を取る確率変数 A, B, C, D, E の結合確率分布が

$$P(A,B,C,D,E) = P(A)P(B|A)P(D|A) \times P(C|D)P(E|D)$$

と与えられているものとしよう。図 3 は、この結合分布における確率変数間の依存関係を表すベイジアンネットワーク [8] である。

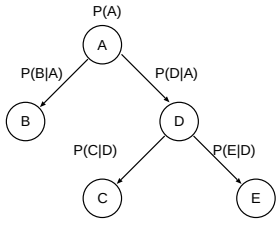


図 3: 結合分布 $P(A, B, C, D, E)$ の構造を表すベイジアンネットワーク

ここで、確率変数 C の観測値 c が得られたものとし、観測後の確率変数 A の事後確率分布 $P(A|C = c) = P(A, C = c)/P(C = c)$ を計算したいものと仮定する。これを計算するには、まず $P(A, C = c)$ を求め、それを規格化するように $P(C = c)$ を求めればよい。また、$P(A, C = c)$ は、結合分布に含まれる A, C 以外の確率変数を周辺化（マージナライズ）することにより計算することができる。すなわち、結合分布 $P(A, B, C, D, E)$ について $P(A, C = c) = \sum_{B,D,E} P(A, B, C = c, D, E)$ を計算すればよい。記法 $\sum_{B,D,E}$ は、変数 B, D, E の取り得る全ての値で和を取ることを意味する。これを計算するには、

$$P(A,C=c) = \sum_{B,D,E} P(A)P(B|A)P(D|A) \times P(C=c|D)P(E|D) \quad (2)$$

と直接計算するよりも

$$P(A,C=c) = P(A)\sum_{B}\{P(B|A)\} \times \sum_{D}\{P(D|A)P(C=c|D)\sum_{E}P(E|D)\} \quad (3)$$

と変形して計算する方が、より少ない算術計算で目的とする $P(A, C = c)$ が得られる。図 2 の例と同様に、この場合も分配則を利用して演算回数の削減が可能となることがわかる。すなわち、結合分布がうまく因子分解できる構造を持っていれば、その構造に従って分配則を利用することにより効率良く、結合分布の周辺化を実行することができる。

Sum-product アルゴリズムは、この結合分布の周辺化をベイジアンネットワーク上での分散アルゴリズムの形で実現したものである。各ノードでは、隣接したノードから送られてくる局所積和（この例の場合だと例えば $\sum_E P(E|D)$ に相当する）を基づいて送り先ノードの変数以外を周辺化した局所積和を計算し、それを隣接した送り先ノードに送信する。例えば、式 (3) の場合には、ノード D はノード C から $P(C = c|D)$ を、ノード E から $\sum_E P(E|D)$ をメッセージとして受け取る。ノード D は、これらを元にノード A に送るメッセージ $M_{A\to D} = \sum_D\{P(D|A)P(C = c|D)\sum_E P(E|D)\}$ を計算し、それをノード A に送る。このメッセージ $M_{A\to D}$ は 、送り先のノード変数 A 以外の変数を周辺化した局所積和である。十分なメッセージ交換の後、隣接したノードから送られてくる全てのメッセージ（局所積和）の積を取ることにより所望の周辺化確率分布が得られる。上の例では、ノード A は、$P(A), \sum_B\{P(B|A)\}, M_{A\to D}$ をそれぞれ自分、ノード

[4]この積和計算において、積を和に、和を最大値に置き換えることにより Viterbi アルゴリズムが得られる [5]。

B, ノード D から受け取る。これらの積を計算することにより $P(A, C = c)$ を得る。

ここで議論した観測値に基づく事後確率分布の計算は、シンボルワイズ MAP 復号法の対数事後確率比計算 (1) に現れる計算と同種の計算である。符号の復号の場合に利用されるグラフには、トレリス、タナーグラフなどがあるが、sum-product アルゴリズムの基本的な計算原理は上に述べたとおりである。

ただし、sum-product アルゴリズムにより正確な事後確率分布が計算できるのは、計算に利用される有向グラフがループを含まない場合に限られる。すなわち、符号のタナーグラフ内にループが存在する場合には、sum-product アルゴリズムは正確な MAP 復号法ではなく、近似 MAP 復号法となる。残念ながら、一般にループを含まないタナーグラフを持つ符号は、ループを持つ符号に比べて最尤復号特性が劣っており、通信路符号化の観点からは興味の対象にはならない。LDPC 符号の場合、符号長が十分に長ければ、そのタナーグラフは短いループをほとんど含まない。このような場合には、sum-product アルゴリズムにより良い復号性能が得られることが経験的に知られている [3]。しかし、タナーグラフにループがあり、通信路の SN 比が低い場合、sum-product アルゴリズムの挙動は複雑であり、その収束性、近似度、その非線形ダイナミクスなど、これからの解決が待たれる問題も多い。

Sum-product アルゴリズムの 1 ラウンドの計算量はタナーグラフに含まれる辺の数に比例する。ここで、1 ラウンドとは、タナーグラフにおいて 2 回のメッセージ伝達が行われる期間を指す。タナーグラフの辺の数は、検査行列に含まれる 1 の数に等しい。LDPC 符号の場合、検査行列に含まれる 1 の数は $O(N)$(N は符号長) であることから、1 ラウンドあたりの復号計算量は $O(N)$ となる。したがって、最大繰り返し数を定数として定めれば、sum-product アルゴリズムによる復号計算量は $O(N)$ となる。また、sum-product アルゴリズムは本質的に並列アルゴリズムであるため、並列分散型ハードウェアによる実装に適している。

参考文献

[1] R.G.Gallager,“Low density parity check codes,” in Research Monograph series. Cambridge, MIT Press (1963).

[2] R.M.Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, Vol.27, pp.533–547 (1981).

[3] D. J. C. MacKay,“Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inform.Theory*, Vol.45, pp.399–431 (1999).

[4] T.J.Richardson, M.A. Shokrollahi, and R.L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inform.Theory*, Vol.47, pp.619–637 (2001).

[5] S.M.Aji and R.J.McEliece, “The generalized distributive law,” *IEEE Trans. Inform. Theory*, Vol.46, pp.325–343 (2000).

[6] F.R.Kschischang, B.J.Frey and H.A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, Vol.47, pp.498–519 (2001).

[7] N.Wiberg, “Codes and decoding on general graph,” Ph.D thesis, Dept. of Electrical Engineering, Liköping University, Sweden (1996).

[8] B.J.Frey, “Graphical models for machine learning and digital communication,” MIT Press (1998).

Enhancing Belief Propagation Decoding of Low Density Parity Check Codes with Reliability Based Decoding

Marc Fossorier（ハワイ大）

概 要

In this study, probabilistic reliability based decoding is combined with belief propagation (BP) decoding for low-density parity check (LDPC) codes. At each iteration, the soft output values delivered by the BP algorithm are used as reliability values to construct a short list of codewords among which the most likely codeword is identified. This approach allows to bridge the error performance gap between belief propagation decoding which remains suboptimum, and maximum likelihood decoding which is too complex to be implemented for the codes considered. Simulations results are given for Gallager LDPC codes and different set cyclic (DSC) codes of hundreds of information bits.

1 Introduction

Iterative decoding of low-density parity check (LDPC) codes, first introduced in [1], has received renewed interest over the past few years. This is mainly due to the excellent error performances achieved by such codes for long enough lengths, and their relatively simple iterative decoding methods based on belief propagation (BP) [2, 3].

Although BP iterative decoding of such codes does not achieve maximum likelihood decoding (MLD), it has been shown both theoretically and experimentally that its application to improved constructions derived from Gallager original codes [4, 5, 6, 7] can closely approach channel capacity for the additive white Gaussian noise (AWGN) channel. Interestingly, other families of codes such as quasi-cyclic based codes [8, 9] and finite geometry codes [10, 11, 12] can also be decoded iteratively with BP. These results are quite surprising as the BP algorithm was originally designed to work on graphs without cycles, while the Tanner graph representation [13] of all codes of practical interest contains cycles [14].

To guarantee that the influence of cycles in the Tanner graph representation of the codes considered is small enough, very long code lengths may been needed [4]. As a result, although BP decoding of codes of medium lengths (say between hundred and few thousand bits) performs quite well, it may still fall short of MLD due to the non negligible influence of cycles in the Tanner graph representation of the code.

In this study, we propose to use the fact that BP iterative decoding of LDPC codes converges quite slowly in general. At each iteration, the soft output values delivered by the BP algorithm are viewed as reliability values by a list-based soft decision decoding algorithm. Due to the slow convergence, sufficiently different lists are expected to be constructed at each iteration and the most likely codeword in the union of these lists is delivered as final solution. Indeed, if BP converges to the MLD codeword, then this codeword is also found by this combined approach. In order to keep a manageable decoding complexity, we choose the ordered statistic decoding (OSD) algorithm of [15] in this study.

2 Decoding Algorithm

A detailed description of the combined algorithm as well as possible refinements and the corresponding computational complexity analysis can be found in [16]. In the following, the basic algorithm is presented for a code of length N and dimension K.

Define Ω_n as the set of check sums containing bit n, and $\omega(m)$ as the set of bits forming check sum m. The iterative BP decoding algorithm has two alternating parts, in which certain quantities q_{mn} and r_{mn}, are iteratively updated. The quantity q_{mn}^x is meant to be the probability that bit n is taking the value x, given the information obtained via checks other than check m (i.e. $\Omega_n \setminus \{m\}$). The quantity r_{mn}^x is meant to be the probability that check m is satisfied if bit n is considered fixed at x and the other bits have a separable distribution given by the probabilities $\{q_{mn'} : n' \in \omega(m) \setminus \{n\}\}$.

Initially, for $n = 1, \cdots, N$, the bit decisions $\hat{x}_{mn}$ and $\hat{x}_n$ are initialized to the symbols z_n corresponding to the hard decision decoding of the noisy received values y_n, and the variables q_{mn}^0 and q_{mn}^1 are initialized to the a-priori probabilities f_n^0 and $f_n^1 = 1 - f_n^0$ associated with the corresponding values z_n and $z_n \oplus 1$. The value D_{min}

is initialized to a large number, and we set $\hat{\mathbf{c}} = \mathbf{0}$. Then each iteration is processed as follows:

- *BP Decoding:*

  Step 1:

  For each n and each m, define $\delta q_{mn} = q^0_{mn} - q^1_{mn}$, and for $x = 0, 1$ compute:

  (a)
  $$\delta r_{mn} = \prod_{n' \in \omega(m)} \delta q_{mn'},$$

  (b)
  $$r^x_{mn} = (1/2)(1 + (-1)^x \delta r_{mn}) \, .$$

  Step 2:

  (a) For each n and each m, and for $x = 0, 1$, update:
  $$q^x_{mn} = \alpha_{mn} f^x_n \prod_{m' \in \Omega_n \setminus m} r^x_{m'n},$$
  where α_{mn} is chosen such that $q^0_{mn} + q^1_{mn} = 1$.

  (b) For each n and $x = 0, 1$, update the "pseudo-posterior probabilities" q^0_n and q^1_n given by:
  $$q^x_n = \alpha_n f^x_n \prod_{m \in \Omega_n} r^x_{mn},$$
  where α_n is chosen such that $q^0_n + q^1_n = 1$.

  Step 3:

  (a) Create $\hat{\mathbf{x}} = [\hat{x}_n]$ such that $\hat{x}_n = 1$ if $q^1_n > 0.5$, and $\hat{x}_n = 0$ if $q_n \leq 0.5$.

  (b) Do the following:

  - If $\mathbf{H}\hat{\mathbf{x}} = \mathbf{0}$ then the decoding algorithm halts and $\hat{\mathbf{x}}$ is considered as a valid decoding result.
  - Else, the algorithm starts the "Order-i Reprocessing".

- *Order-i Reprocessing:*

  Step 1:

  Based on the reliability values $\max\{q^0_n, q^1_n\}$, determine the K corresponding most reliable independent positions (MRIP's). Define B_K as the set of locations of these MRIP's.

  Step 2:

  (a) Take $I(B_K) = \{\hat{x}_j : j \in B_K\}$ created at Step 3-(a) of BP decoding as initial information sequence.

  (b) For $0 \leq l \leq i$, make all possible $\binom{K}{l}$ changes of l values in $I(B_K)$ and for each change, determine the corresponding codeword $\mathbf{c}$.

  Step 3:

  Let $\hat{\mathbf{c}}$ denote the codeword with minimum discrepancy D_{min} constructed so far by the iterative combined decoding algorithm.

  (a) For each codeword $\mathbf{c} \neq \hat{\mathbf{c}}$ constructed at Step 2-(b), evaluate the discrepancy:
  $$D(\mathbf{y}, \mathbf{c}) = \sum_{i: c_i \neq z_i} |y_i|$$
  between $\mathbf{c}$ and the original noisy received sequence $\mathbf{y}$.

  If $D(\mathbf{y}, \mathbf{c}) < D_{min}$, set $D_{min} = D(\mathbf{y}, \mathbf{c})$ and $\hat{\mathbf{c}} = \mathbf{c}$.

  (b) Start the next iteration with BP decoding.

The algorithm stops if some maximum number of iterations is reached and the codeword $\hat{\mathbf{c}}$ is delivered as the final solution.

The OSD can also be processed based on the initial values f^0_n and f^1_n, in which case iteration-1 becomes equivalent to the conventional order-i reprocessing of [15]. Furthermore, stopping rules can also be introduced in BP decoding (e.g. syndrome test at Step 3-(b)) and in the order-i reprocessing [16].

3 Simulation Results

Simulation results for LDPC codes and different set cyclic (DSC) codes of hundreds of information bits have been obtained. Figure 1 depicts the simulation results for the (3,6) (i.e. each bit is associated with three orthogonal check sums of Hamming weight six) Gallager code with parameters $(N, K) = (504,252)$. In this figure, BP decoding is compared with the combined scheme proposed above for $i = 1$ and $i = 2$. We observe that for the WER $10^{-3.5}$, order-1 reprocessing of at most 50 and 200 BP decoding iterations achieves 0.55 dB and 0.6 dB coding gain, respectively, over BP decoding of at most 200 iterations. At the same WER, order-2 reprocessing of at most 50 BP decoding iterations achieves a 0.75 dB coding gain over BP decoding of at most 200 iterations. Similar results were obtained for a (3,6) (1008,504) LDPC code [16].

Figure 2 depicts the simulation results for the (273,191) DSC code and its (3,6) and (4,8) LDPC code counterparts. We observe that a much larger coding gain is achieved by the proposed scheme for decoding the LDPC codes than for decoding the DSC code. At the WER 10^{-4}, a coding gain of 0.9 dB is achieved by the proposed scheme to decode the (4,8) LDPC code. A similar behavior was observed for the (1057,813) DSC code and its (3,6) and (4,8) LDPC code counterparts [16]. Importantly, based on the results of Figure 1, the need to distinguish between "good codes" and "good iteratively decodable codes" can be emphasized.

4 Conclusion

In this study, we have shown that for LDPC codes of short to medium length, a non negligible gap exists between the error performance of BP decoding and that of MLD. We have proposed a simple combined method to bridge this gap based on the combination of BP decoding with a list-based soft decision decoding algorithm. This method can be viewed either as an improved decoding scheme, or a way to measure the suboptimality of BP decoding.

Although the OSD algorithm was considered in this study, the same approach can be applied with other list-based algorithms as well as with reduced complexity versions of BP decoding. In [17], this combined approach has been generalized to iterative decoding of "turbo-like" codes [18, 19] and its effectiveness has been related to the convergence behavior of the iterative decoding considered.

5 Acknowledgments

The author wishes to thank Michael Tanner for numerous valuable comments which have been conveyed in this article.

参考文献

[1] R.G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp.21-28, Jan. 1968.

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

[3] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399-431, Mar. 1999.

[4] T. Richardson and R. Urbanke, "The Capacity of Low-Density Parity Check Codes under Message-Passing Decoding," *IEEE Trans. Inform. Theory*, vol.47, pp.599-681, Feb. 2001.

[5] D.J.C. MacKay, S.T. Wilson and M.C. Davey, "Comparison of Constructions of Irregular Gallager Codes," *IEEE Trans. Commun.*, vol. 47, pp. 1449-1454, Oct. 1999.

[6] T. Richardson, A. Shokrollahi and R. Urbanke, "Design of Capacity Approaching Irregular Low-Density Parity Check Codes," *IEEE Trans. Inform. Theory*, vol.47, pp.619-637, Feb. 2001.

[7] S.Y. Chung, G.D. Forney, Jr., T.J. Richardson and R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit," *IEEE Commun. Letters*, vol.5, pp. 58-60, Feb. 2001.

[8] R.M. Tanner, "A Transform Theory for a Class of Group-Invariant Codes," *IEEE Trans. Inform. Theory*, vol.34, pp. 752-775, July 1988.

[9] R.M. Tanner, "Spectral Graphs for Quasi-Cyclic LDPC Codes," *Proc. 2001 IEEE Intern. Symp. Inform. Theory*, Washington, D.C., USA, p. 226.

[10] R. Lucas, M. Fossorier, Y. Kou and S. Lin. "Iterative Decoding of One-Step Majority Logic Decodable Codes Based on Belief Propagation," *IEEE Trans. Commun.*, vol. 48, pp. 931-937, June 2000.

[11] Y. Kou, S. Lin and M. Fossorier, "Low Density Parity Check Codes Based on Finite Geometries: A Rediscovery and More," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2711-2736, Nov. 2001.

[12] P. Vontobel and R.M. Tanner, "Construction of Codes Based on Finite Generalized Quadrangles for Iterative Decoding," *Proc. 2001 IEEE Intern. Symp. Inform. Theory*, Washington, D.C., USA, p. 223.

[13] R.M. Tanner, "A Recursive Approach to Low Complexity Codes, *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, Sept. 1981.

[14] T. Etzion, A. Trachtenberg and A. Vardy, "Which Codes Have Cycle-Free Tanner Graphs?", *IEEE Trans. Inform. Theory*, vol. 45, pp. 2173-2181, Sept. 1999.

[15] M. Fossorier and S. Lin, "Soft-Decision Decoding of Linear Block Codes based on Ordered Statistics," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1379-1396, Sept. 1995.

[16] M. Fossorier, "Iterative Reliability-Based Decoding of Low Density Parity Check Codes," *IEEE J. Sel. Areas in Commun.*, vol.19, pp.908-917, May 2001.

[17] M. Isaka, M. Fossorier and H. Imai, "Iterative Reliability-Based Decoding of Turbo-like Codes," submitted to *IEEE Trans. Commun.*, Nov. 2001.

[18] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," *IEEE Trans. Commun.*, vol. 44, pp.1261-1271, Oct. 1996.

[19] D. Divsalar, H. Jin and R. McEliece, "Coding Theorems for 'Turbo-Like' Codes," *The Proc. 36-th Allerton Conf.*, Monticello, USA, pp. 201-210, Sept. 1998.

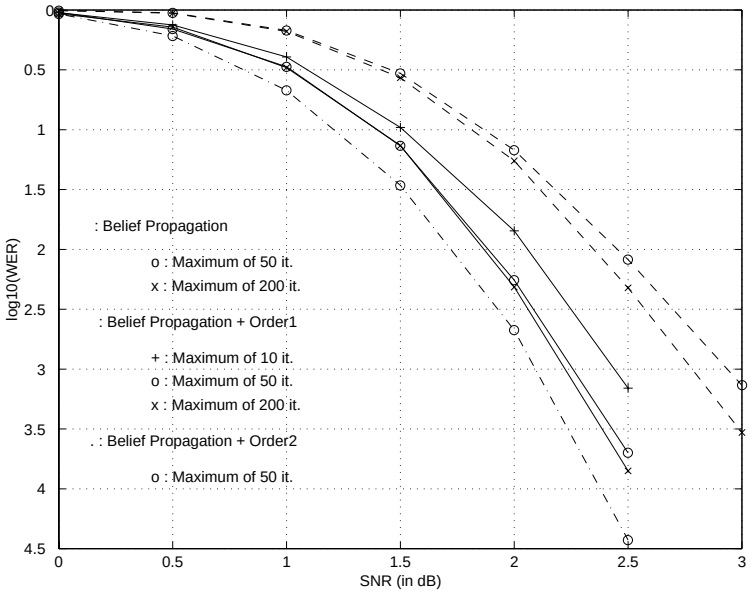


Figure 1: Word error rate comparison between BP decoding and BP decoding in conjunction with order-1 reprocessing for the (3,6) (504,252) LDPC code.

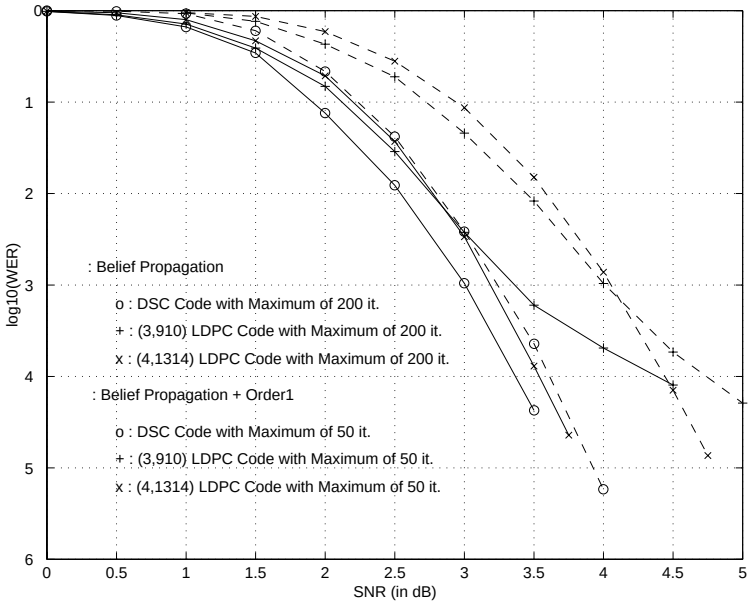


Figure 2: Word error rate comparison between BP decoding and BP decoding in conjunction with order-1 reprocessing for the (273,191) DSC code and its (3,6) and (4,8) LDPC code counterparts.

微分幾何から見た確率推論と復号

松嶋 敏泰（早稲田大学）

1 はじめに

誤り訂正符号の復号問題は受信系列から情報系列を推定する問題と考えることができ，その本質は受信系列を与えられたもとでの情報系列の事後確率計算と同等の問題に帰着される．また，BN(Bayesian network)[14] を用いた知識表現の分野おいて，証拠 (evidence) が与えられもとで各命題の確信度 (belief) の更新を行うことは，ある確率変数の値が与えられたもとでのその他の確率変数の事後周辺確率を求めていることに他ならない．よって両者は共通の問題と考えることができ，近年 Turbo 復号アルゴリズム [4][7] や LDPC 符号で用いられる sum-products アルゴリズム [6][15] は BN の代表的推論アルゴリズムである BP(Belief Propagation)[14] と等価なアルゴリズムとして共通の視点から議論されている．このような事後確率の計算問題は符号理論における復号問題や人工知能分野における不確実な知識を扱う問題のみならず，統計学をはじめ，統計力学の分野，学習理論の分野，パターン認識の分野など 様々な分野において共通で重要な問題となっている．

本稿では，まず誤り訂正符号の復号問題を事後確率の計算問題として整理し，確率推論と本質的に同じ 問題であることを述べる．次に確率変数間の従属関係が部分的に存在する確率分布のグラフを用いた代表的表現法についてまとめる．そして，そのグラフ上で情報を伝搬させることにより効率的に事後確率を計算するアルゴリズムについて概観する．最後に微分幾何を用いた事後周辺確率の計算についての解釈と最近の研究を簡単に紹介する．

2 誤り訂正符号の復号と事後確率計算

情報系列 $\mathbf{u} = (u_1, \cdots, u_K)$, $u_k \in A$ より符号語 $C(\mathbf{u})$ を生成し，通信路を通して受信側へ送り，受信系列 $\mathbf{y}$ が受信されたとする．復号の問題は，受信系列を $\mathbf{y}$ としたもとで，受信系列 $\mathbf{y}$ から符号語 $C(\mathbf{u})$ または情報系列 $\mathbf{u}$ を推定する問題に帰着される．推定は情報系列全体 $\mathbf{u}$ をブロックとして推定する場合と各情報シンボル u_k を推定する場合に大別される．ブロック誤り率を最小化する推定 (復号) 法は受信系列を与えられたもとでの事後確率 $P(\mathbf{u}|\mathbf{y})$ を最大化する情報系列 $\hat{\mathbf{u}}$ を推定値とすることになる．これは $\mathbf{u}$ の事前確率が一様な場合は，尤度 $P(\mathbf{y}|\mathbf{u})$ を最大化する情報系列と一致し，通常これを最尤復号と呼んでいる．一方，シンボル誤り率を最小化する推定法は事後確率 $P(u_k|\mathbf{y})$ を最大化する情報シンボル $\hat{u}_k$ を推定値とすることになる．通常これを MAP(Maximum a posteriori probability) 復号 (正確には情報シンボルに対する最大事後確率復号) または MPM(Maximum posterior marginal) 復号などと呼んでいる．

$\mathbf{u}$ を推定する場合も u_k を推定する場合もそれぞれの確率変数の事後確率を計算し，その最大値を探索するという意味で基本的には同じ 手続きで復号は行われるが，計算の困難な部分は両者で異なっている．一般に情報系列全体の事後確率 $P(\mathbf{u}|\mathbf{y})$ は比較的容易に計算可能であるが，最大値をとる $\hat{\mathbf{u}}$ を $|A|^K$ の候補から探索する問題は指数オーダ $O(|A|^K)$ の計算量が必要とされる．この状況は尤度最大化の問題でも同様で，通信路復号化の多くの研究はこの最適な最尤復号に近い誤り率を，少ない計算量で実現する事に向けられてきた．

逆に，個々の情報シンボル u_k を推定する場合において，最大値を見つけることは $|A| - 1$ の比較で容易であるが，各情報シンボルの事後確率 $P(u_k|\mathbf{y})$ を計算することは，事後結合確率 $P(\mathbf{u}|\mathbf{y})$ から周辺確率を計算することに対応し，一般には指数オーダ $O(|A|^K)$ の計算量となる．この事後周辺確率計算を，少ない計算量で近似計算することに成功したある種の反復アルゴリズムが，Turbo 復号や LDPC 符号における sum-products アルゴリズムである．

このように事後確率の値を求めることがその問題の主要な課題となっている問題は，様々な分野で見受けられる．BN をはじめとする確率推論の分野おいて，証拠が与えられもとで各命題の確信度の更新を行うことは主要な問題であり，この問題は，ある確率変数の値が与えられたもとでのその他の確率変数の事後周辺確率を求めている上記の復号問題と同等の問題となっている．

3 確率変数間の相互関係のグラフ表現

事後周辺確率の計算は一般に多くの計算量を必要とするが，確率変数間の相互の関連が部分的にのみ存在する確率分布においてはその計算が容易になる可能性がある．相互の関連が部分的とは，系全体の確率構造を表す結合確率が，確率変数の部分集合の関数の積で表される場合を指す．その関数としては，例えば条件付き確率やポテンシャル関数が用いられている．

これらの部分的に確率変数が関連を持つ分布の確率構造を表現するには，確率変数を節に対応させ，関連のある確率変数の節間を枝で結んだグラフを用いると分かりやすく表現できる．用いられるグラフ表現は分野によって様々であるが，まず無向グラフと有向グラフに大別される．前者としては MRF[11]，moral グラフ [12] 等があり，後者として代表的なのが BN である．BN は系全体の確率を条件付き確率の積で表し，条件付き確率を親節から子節への有向枝に対応させグラフ化している．正確には BN は cycle が無い有向グラフ (DAG: Directed Acyclic Graph) で定義され，知識処理の分野で広く用いられている．

また，符号のグラフ表現としては，白丸の節で確率変

数を表し，黒丸の節で関連の有ることを表す Tanner グラフ [11] が従来から提案されていた．Tanner グラフは近年 factor グラフとして一般化されている [11]．また，畳み込み符号や一部の線形符号は，トレリスによってその構造を簡素に表現可能であり，従来から広く用いられている．

このようなグラフ表現は，部分的に確率変数が関連を持つ分布の確率構造を視覚的に分かりやすくする為だけではなく，このグラフを用いて事後確率計算等を効率良く行うアルゴリズムを構成するために役立っている．

4 グラフ上での伝播を用いた効率的事後確率計算アルゴリズム

BN における確信度更新，つまり事後周辺確率の計算における効率的アルゴリズムはとしては BP が代表的である．BP では有向グラフの順方向へメッセージ π のと逆方向へのメッセージ λ 伝搬させることにより各節の事後周辺確率を求めている．この BP はも事後周辺確率を部分的な確率計算に分割して行う効率的アルゴリズムと解釈することができる．BN は DAG として定義されているが，BP を用いることができるグラフは DAG の部分クラスである Polytree(一般木) と呼ばれる loop の無い DAG のクラスに限定されている [14].

BP の性能が保証できるクラスはこの Polytree に限られるが，計算量は確率変数の数とアルファベットの大きさの多項式オーダとなる効率の良いアルゴリズムとなっている．Polytree より広いクラスの DAG に対しても正確さの保証のある計算アルゴリズムが幾つか提案されているが計算量は増加してしまう [14] [12]．この問題を計算量から考えると，DAG の一般的クラスにおける事後周辺確率計算問題は NP 困難となることが計算理論分野で有名な 3-SAT 問題を用いて証明されている [5].

その他のグラフ表現として factor グラフにおいては，効率的な事後周辺確率計算アルゴリズムとして sum-products アルゴリズムがある．グラフ上で 2 種類の情報を 2 種類の節を通して流し計算をするアルゴリズムで，節に集まってきた情報を和と積で計算することでこのように呼ばれる．BP と本質的には同じアルゴリズムであることが示されており，このアルゴリズムも factor グラフに loop がない場合のみ正しい計算の保証がある．

BP や sum-products アルゴリズムが正しく事後周辺確率を計算できる保証は，この木構造上で条件付き独立の性質をうまく用いた点にある．BP と同等なアルゴリズムは様々な分野で用いられている．例えば，隠れマルコフモデルにおいて，観測値が与えられたもとでの各時点の状態の事後確率を求める効率的アルゴリズムとして，forward-backward アルゴリズム [3] があるがこれも BP と同等である．

また畳み込み符号を BN で表現すると Polytree で表現可能なため，BP アルゴリズムで正確な事後周辺確率が計算できることになる．この BP がトレリス符号に対し正確な MAP 復号を実現する BCJR[2] アルゴリズムと同等であり，forward-backward アルゴリズムそのものであることは良く知られている．畳み込み符号の MAP 復号は隠れマルコフモデルの状態推定と同等な問題であるので，2つのアルゴリズムが一致することは当然の帰結といえる．さらに semi-ring 上の演算の定義を変えれば forward-backward アルゴリズムは Viterbi アルゴリズムとも等価となる [8] [11]．これらのアルゴリズムはすべて BP の特殊形と解釈されるので，符号を BN で表現した場合に Polytree で表現可能であれば，その符号に対して BP と等価な効率よい復号アルゴリズムが構成できることになる．

しかし，オリジナルの Turbo 符号 (並列連接畳み込み符号) をはじめ多くの符号は，DAG による表現は可能であるが Polytree で表現されないことが多い．Turbo 符号全体の DAG は loop があるが，それぞれの畳み込み符号を中心とした 2 つの部分グラフに分割すれば，それぞれは Polytree となり，BP と等価な BCJR アルゴリズムを用いることができる．Turbo 復号ではそれぞれの部分グラフに BCJR アルゴリズムを交互に用い反復することにより，事後周辺確率の近似値を求めていることになる．これは loop があり Polytree とはなっていない符号の DAG に対して，グラフを Polytree となる部分グラフに分割し，まず一方の Polytree 上で BP により各節の暫定的な事後確率計算を行う．次にその Polytree 上で得られた情報系列に関する暫定的な事後確率を用いてもう一方の Polytree 上でまた BP を行い，それを反復することで暫定事後確率の値を更新し収束させていると考えることができる．

また，Gallager により提案され最近再発見された LDPC（低密度パリティ検査）符号 [6][15] は，2 元符号で検査行列 H の各列の 1 の数を J 個と一定数の小さい数に制限した符号である．この符号も DN や factor グラフで表現すると loop のあるグラフとなってしまう．そこで，turbo 復号と同様に符号を表現したグラフを幾つかの loop のない部分グラフに分解し BP または sum-products アルゴリズムで反復計算し復号を行っている．sum-products アルゴリズムを用いる場合，節に集まってくる情報の数が J となることで和の計算の次元が J となり，効率的計算が可能となっている．

このように一般の DAG に BP を無理やりに用いた場合の性能は保証されるのであろうか．残念ながら，一般的には正しい事後確率を計算することの保証どころか，収束性も保証されていない．しかし，これらの 2 つの符号をはじめいくつかの符号に対しては，復号誤り特性に関する多くのシミュレーションの結果から，これらの復号法の良好な性能が確認されている．理論的面からは，BP の演算を変換した BP と同質なアルゴリズムにおいて，例えば，loop が一つでアルファベットが 2 元の DAG に対しアルゴリズムは収束し，収束結果は真の最大事後確率による推定と一致するという意味で正当性が示されている [18]．この結果を拡張して単一 loop が複数存在する DAG に対しても，ある範囲内で正当性の保証が得られることも最近報告されている．

5 微分幾何からみた事後確率計算

n 次元のパラメータ θ で表現される確率分布 $P(X,\theta)$ 全体の集合 S を考えてみる．この S は n 次元多様体をなし，θ はその局所座標系の一つとなる．この空間にある種の微分幾何学的構造を導入したものが情報幾何学 [1] と呼ばれ，情報理論，統計的推論，ニューラルネット等の分野で応用されている．最近，Turbo 復号や LDPC 符号における sum-products アルゴリズムについてもこの情報幾何を用いた研究 [9][16][17] が行われている．

座標系や確率変数のスケール等に不変性を仮定すると，計量として Fisher 情報量や双対な接続が導かれる．また，距離の役目をする分離度として KL 情報量とその双対の情報量が導かれ，e 座標系と m 座標系と呼ばれる双対の座標系や，e 射影と m 射影と呼ばれる双対の射影も導かれる．

このような微分幾何的な視点から本稿の中心的問題である事後周辺確率の計算を眺めてみよう．事後確率分布 $P(X|Y=y)$ は，多様体上のある点として表される事前分布 $P(X,Y)$ から $P(Y=y)$ の制約を満たす部分空間上への e 射影と考えることができる．またある確率分布から周辺分布を求めることは m 射影とみなすことができる．つまり，事後周辺確率の計算は e 射影と m 射影の 2 つの基本的射影により簡素に表現できることになる．

BP のような効率的事後周辺確率の計算は，上記の視点からはどのように見えるのであろうか．ある部分空間上での e 射影と m 射影を繰り返すことで，本来は空間全体での e 射影と m 射影により求まる事後周辺確率を求めていると考えられる．各部分空間の関係が条件付独立を満たしている場合，この射影により正しい事後周辺分布を求めることができる．

Turbo 復号や LDPC 符号における sum-products アルゴリズムのように loop のある BN に BP を適用した場合について，情報幾何を用いた解析が最近なされている [9]. この研究では m 射影の挙動に注目することにより，停留点の安定性や解のずれなどについて考察されている．

また，対数線形モデルと呼ばれる分布のクラス上で e 射影と m 射影を繰り返す視点から，周辺事後確率を計算するいくつかのアルゴリズムが提案 [16] されている．これらは BP や sum-products アルゴリズムの一般化と見なすことができ，より広い確率分布のクラスや符号に適用可能なアルゴリズムになっている．

参考文献

[1] S. Amari, *Differential-geometry in statistical inference*, Springer lecture notes in statistics vol.28, Springer, 1985.

[2] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate* IEEE Trans. IT, Vol.20, 1974.

[3] L.E. Baum and T. Petrie, *Statistical inference for probabilistic functions on finite state markov chains* Annals of Mathematical Statistics, Vol.37, 1966.

[4] C. Berrou, A. Glavieux and P. Thitimajshima, *Near Shannon limit error-correcting coding and decoding*, in Proc. IEEE Int. Conf. Commun., 1993.

[5] P. Dagum, *Approximating Probabilistic Inference in Bayesian belief networks is NP-hard* Artificial Intelligence, Vol.42, 1990.

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

[7] J. Hagenauer, E. Offer and L. Papke, *Iterative decoding of binary block and convolutional codes*, IEEE Trans. IT, Vol.42, 1996.

[8] C. Heegard and S.B. Wicker, *Turbo coding* Kluwer Academic Publishers, 1999.

[9] 池田思朗，田中利幸, 甘利俊一, ターボ符号の情報幾何, IBIS2001, 2001.

[10] 井坂元彦, 今井秀樹, *Shannon* 限界への道標*: "Parallel concatenated (Turbo) coding", "Turbo (Iterative) decoding"*とその周辺, 信学技報 IT98-51, 1998.

[11] F.R. Kschischang and B.J. Frey, *Iterative decoding of compound codes by probability propagation in graphical models*, IEEE J. Sel. Areas Commun., Vol.16 No.2, 1998.

[12] S.L. Lauritzen and D.J. Spiegelhalter, *Local computation with probabilities on graphical structures and their application to expert systems*, J.R. Statist. Soc., Vol.50 No.2, 1988.

[13] R.J. McElice, D.J.C. MacKay and J. Cheng, *Turbo decoding as an instance of Pearl's "Belief Propagation"*, IEEE J. Sel. Areas Commun., Vol.16 No.2, 1998.

[14] J. Pearl, *Probabilistic reasoning in intelligent systems* Morgan Kaufmann, 1988.

[15] D.J.C. MacKay, *Good Error-Correcting Codes Based on Very Sparse Matrices* , IEEE IT., Vol.45 No.2, 1999.

[16] T. Matsushima, T.K. Matsushima and S. Hirasawa, *An Iterative Algorithm for Calculating Posterior Probability and Model Representation*, Proc. IEEE Int. Sympo. on Information Theory, 2001.

[17] 渡辺曜大, 今井秀樹, **グラフィカルモデルの情報幾何**, 第 23 回情報理論とその応用シンポジューム, 2000.

[18] Y. Weiss, *Belief propagation and revision in networks with loop*, M.I.T A.I. Memo No.1616, 1997.

SITA ニューズレター・ホームページＵＲＬの変更について

SITA ニューズレター 37 号から、閲覧用ホームページの URL が
http://www.ogawa.nuee.nagoya-u.ac.jp/~yamazato/sitanl/
に変更となりました。ここでは、36 号以降を掲載しています。

22 号〜35 号のバックナンバーについては、暫定的に
http://sita-nl.ics.nitech.ac.jp/
で閲覧が可能です。

編 集 後 記

９月以降，わが家の地上波テレビの映りが良くなりました．航空機による反射が減少したためでしょう．ワイキキで「ほら，また飛んだ！」と K 大学の Sa 先生が言われていたのを思い出します．特に島を結ぶ航空便は激減です．日本からの観光客数も，通常の３分の１以下まで落ち込みましたが，ホノルルマラソン以降は，除々に増加してきました．いつになったら，テレビ映りが悪くなってくれるのか．

（岡）

編 集 担 当 者

岡 育生（編集理事）
〒 558-8585 大阪市住吉区杉本 3-3-138
大阪市立大学工学部情報工学科
Tel. 06-6605-2779
Fax. 06-6690-5382
Email oka@info.eng.osaka-cu.ac.jp

山里 敬也（編集幹事）
〒 464-8603 名古屋市千種区不老町
名古屋大学情報メディア教育センター
Tel. 052-789-2743
Fax. 052-789-3173
E-mail yamazato@nuee.nagoya-u.ac.jp

杉村 立夫（編集理事）
〒 380-8553 長野市若里 4-17-1
信州大学工学部電気電子工学科
Tel. 026-269-5237
Fax. 026-268-5220
Email tsugimu@gipwc.shinshu-u.ac.jp

野上 保之（編集幹事）
〒 700-8530 岡山市津島中 3-1-1
岡山大学工学部通信ネットワーク工学科
Tel. 086-251-8128
Fax. 086-251-8127
E-mail nogami@cne.okayama-u.ac.jp

情報理論とその応用学会事務局
〒 113-8656 東京都文京区本郷 7-3-1
東京大学工学部計数工学科
数理第 3 研究室内，山本博資　気付
Tel: 03-5841-6930（直通）
Fax: 03-5841-8605
E-mail: sita-ofice@hy.t.u-tokyo.ac.jp