

Gaussian Belief Propagation

小松和暉

1 はじめに

Gaussian Belief Propagation (GaBP) について解説をする. GaBP は以下のような線形モデルで送信ベクトル \mathbf{x} を推定するために用いられる.

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \quad (1)$$

ここで, それぞれの変数のサイズは観測ベクトル \mathbf{y} は $N \times 1$, チャネル行列 \mathbf{H} は $N \times M$, 送信ベクトル \mathbf{x} は $M \times 1$, 雑音ベクトル \mathbf{z} は $N \times 1$ である. ただし $M < N$ とし, 雑音ベクトルは要素が互いに独立な複素ガウス分布に従うとする. このようなモデルは, たとえば CDMA や MIMO などに現れる. ここではチャネル行列 \mathbf{H} は既知であるとする. もちろん, ゼロフォーシング (ZF) や MMSE により \mathbf{x} を推定できるが, チャネル行列 \mathbf{H} が大きな場合には計算量の問題が生じる. 具体的には ZF や MMSE では逆行列の計算が必要であるため, チャネル行列 \mathbf{H} が特別な構造を持たない限りは $\mathcal{O}(M^3 + M^2N)$ の計算量が必要である. GaBP はこの問題を一回の繰り返しあたり $\mathcal{O}(MN)$ で解くことができる.

2 GaBP

まず, LDPC の sum-product 復号や GaBP は「複数の制約条件を満たすような良さそうな解を探す」アルゴリズムである. たとえば, LDPC の sum-product 復号では「パリティ方程式が成り立つ」ことが制約条件であり, 関数ノードは制約条件を満たすかどうかの対数尤度比を計算する. GaBP では「 $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$ が成り立つ」ことが制約条件であり, 関数ノードではその対数尤度比を計算する. このように, sum-product 復号と GaBP はその制約条件が異なるだけで, 似たアルゴリズムである. そのため, LDPC の sum-product 復号法の導出過程がわかれば, GaBP の導出もそれほど難しくない. ここでは, LDPC の sum-product 復号法に似せるために, $x = \{+1, -1\}$ という BPSK 信号とする. GaBP で最終的に計算したい値は, 受信ベクトル \mathbf{y} が観測されたときの送信ベクトルの i ビット目の対数尤度比 $\text{LLR}(x_i|\mathbf{y})$ である. これは, 次のようにベイズの定理により変換できる.

$$\begin{aligned} \text{LLR}(x_i|\mathbf{y}) &= \log \frac{p(\mathbf{y}|x_i = +1)p(x_i = +1)}{p(\mathbf{y}|x_i = -1)p(x_i = -1)} \\ &= \log \frac{p(\mathbf{y}|x_i = +1)}{p(\mathbf{y}|x_i = -1)} + \log \frac{p(x_i = +1)}{p(x_i = -1)} \quad (2) \\ &= \log \frac{p(\mathbf{y}|x_i = +1)}{p(\mathbf{y}|x_i = -1)} \end{aligned}$$

最後の式へは事前情報がない場合, $p(x_i = +1) = p(x_i = -1) = 0.5$ であることを利用して第二項を 0 とした. こ

こで, 雑音の分布が独立であるため, 観測ベクトルの各要素で展開すれば以下を得る.

$$p(\mathbf{y}|x_i) = \prod_{j=1}^N p(y_j|x_i) \quad (3)$$

つまり, 計算したい LLR 値は

$$\begin{aligned} \text{LLR}(x_i|\mathbf{y}) &= \log \frac{\prod_{j=1}^N p(y_j|x_i = +1)}{\prod_{j=1}^N p(y_j|x_i = -1)} \quad (4) \\ &= \sum_{j=1}^N \text{LLR}(y_j|x_i) \end{aligned}$$

となる. これが GaBP の変数ノードでの処理である.

あとは, この各観測値の対数尤度比 $\text{LLR}(y_j|x_i)$ が計算できればよい. この値を計算することが関数ノードでの処理である. そして関数ノードでは, 制約条件「 $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$ が成り立つ」かどうかの対数尤度比を計算し, それを $\text{LLR}(y_j|x_i)$ として返す. ここで, ソフトビットという概念を導入する. ソフトビットとは LLR 値から推定されるビット値であり, $+1$ や -1 の尤度が高ければ高いほど $+1$ や -1 に近づく値であり, 次のように計算できる.

$$\begin{aligned} \hat{x}_i &= (+1) \cdot p(x_i = +1|\mathbf{y}) + (-1) \cdot p(x_i = -1|\mathbf{y}) \\ &= \tanh \left(\frac{\text{LLR}(x_i|\mathbf{y})}{2} \right) \quad (5) \end{aligned}$$

ここで $q = p(x_i = -1|\mathbf{y})$ とすれば $p(x_i = +1|\mathbf{y}) = 1 - q$ であり, $\hat{x}_i = 1 - 2q$ である. このソフトビットは現在の送信ベクトルのビットの推定値であるため, このソフトビットを用いて受信信号から干渉除去を行う. 具体的には, 現在 i 番目のビットに関して $\text{LLR}(y_j|x_i)$ を計算したいとする. このとき, j 番目の観測値は以下のように記述できる.

$$y_j = \sum_{k=1}^M h_{jk}x_k + z_j \quad (6)$$

今, i 番目のビットのみに着目したいため, それ以外のビットは干渉になってしまう. そのためソフトビットを用いて次のように干渉除去を行う.

$$\begin{aligned} y'_{ji} &= y_j - \sum_{k=1, k \neq i}^M h_{jk}\hat{x}_k \\ &= h_{ji}x_i + \sum_{k=1, k \neq i}^M h_{jk}(x_k - \hat{x}_k) + z \quad (7) \end{aligned}$$

$h_{ji}x_i$ を左辺に移項すれば、以下の式が得られる.

$$y'_{ji} - h_{ji}x_i = \sum_{k=1, k \neq i}^M h_{jk}(x_k - \hat{x}_k) + z \quad (8)$$

右辺の干渉除去された信号の残留干渉の期待値と分散は次の通りである. ただし, $q = p(x_i = -1|\mathbf{y})$ であり $\hat{x}_i = 1 - 2q$ である.

$$\begin{aligned} \mathbb{E}[x_i - \hat{x}_i] &= p(x_i = 1|\mathbf{y})(1 - \hat{x}_i) + p(x_i = -1|\mathbf{y})(-1 - \hat{x}_i) \\ &= (1 - q)(1 - (1 - 2q)) + q(-1 - (1 - 2q)) \\ &= (1 - 2q) - (1 - 2q) = 0 \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbb{E}[|x_i - \hat{x}_i|^2] &= p(x_i = 1|\mathbf{y})|1 - \hat{x}_i|^2 + p(x_i = -1|\mathbf{y})|-1 - \hat{x}_i|^2 \\ &= (1 - q)|1 - (1 - 2q)|^2 + q|-1 - (1 - 2q)|^2 \\ &= 4q - 4q^2 = 1 - (1 - 4q + 4q^2) = 1 - (1 - 2q)^2 \\ &= 1 - \hat{x}_i^2 \end{aligned} \quad (10)$$

そのため, 左辺や右辺全体の期待値や分散は

$$\mathbb{E}[y'_{ji} - h_{ji}x_i] = 0 \quad (11)$$

$$\sigma_{ji}^2 = \mathbb{V}[y'_{ji} - h_{ji}x_i] = \sum_{k=1, k \neq i}^M |h_{jk}|^2(1 - \hat{x}_k^2) + \mathbb{V}[z] \quad (12)$$

となる. $\text{LLR}(y_j|x_i)$ を計算するためには, この $y'_{ji} - h_{ji}x_i$ の確率分布を求める必要がある. ここで, 干渉除去されて残留している部分は多数の確率変数の和であるため, 中心極限定理によりガウス分布に近似する. そのため, 確率変数 $y'_{ji} - h_{ji}x_i$ の確率密度関数は

$$p(y'_{ji} - h_{ji}x_i) = \frac{1}{\pi\sigma_{ji}^2} \exp\left\{-\frac{|y'_{ji} - h_{ji}x_i|^2}{\sigma_{ji}^2}\right\} \quad (13)$$

となる. つまり, 計算したい LLR 値は以下の通りである.

$$\begin{aligned} \text{LLR}(y_j|x_i) &= \log \frac{p(y'_{ji} - h_{ji}(+1))}{p(y'_{ji} - h_{ji}(-1))} \\ &= \frac{|y'_{ji} + h_{ji}|^2 - |y'_{ji} - h_{ji}|^2}{\sigma_{ji}^2} \\ &= \frac{4\Re[y'_{ji}h_{ji}^*]}{\sigma_{ji}^2} \end{aligned} \quad (14)$$

以上をすべてまとめれば, GaBP の更新式は次の通りである.

- 変数ノード

$$\text{LLR}(x_i|\mathbf{y}) = \sum_{j=1}^N \text{LLR}(y_j|x_i) \quad (15)$$

- 関数ノード

$$\hat{x}_i = \tanh\left(\frac{\text{LLR}(x_i|\mathbf{y})}{2}\right) \quad (16)$$

$$y'_{ji} = y_j - \sum_{k=1, k \neq i}^M h_{jk}\hat{x}_k \quad (17)$$

$$\sigma_{ji}^2 = \sum_{k=1, k \neq i}^M |h_{jk}|^2(1 - \hat{x}_k^2) + \mathbb{V}[z] \quad (18)$$

$$\text{LLR}(y_j|x_i) = \frac{4\Re[y'_{ji}h_{ji}^*]}{\sigma_{ji}^2} \quad (19)$$

しかし, このままでは関数ノードで生成するメッセージ数が 1 ノードあたり M 個であり, 各メッセージの計算には $\mathcal{O}(M)$ の計算量が必要なため, 全関数ノードの計算量が $\mathcal{O}(M^2N)$ となる. そのため, 関数ノードで生成するメッセージは次の二つの値に変更する.

$$y'_j = y_j - \sum_{k=1}^M h_{jk}\hat{x}_k \quad (20)$$

$$\sigma_j^2 = \sum_{k=1}^M |h_{jk}|^2(1 - \hat{x}_k^2) + \mathbb{V}[z] \quad (21)$$

これらと式 (17) 及び式 (18) の違いは, 総和記号に $k = i$ の項を含めるかどうかである. そして, 変数ノードでは次の計算を行う.

$$y'_{ji} = y'_j + h_{ji}\hat{x}_i \quad (22)$$

$$\sigma_{ji}^2 = \sigma_j^2 - |h_{ji}|^2(1 - \hat{x}_i^2) \quad (23)$$

$$\text{LLR}(y_j|x_i) = \frac{4\Re[y'_{ji}h_{ji}^*]}{\sigma_{ji}^2} \quad (24)$$

$$\text{LLR}(x_i|\mathbf{y}) = \sum_{j=1}^N \text{LLR}(y_j|x_i) \quad (25)$$

この変更によって, 関数ノードでも変数ノードでも, その計算量は一回あたり $\mathcal{O}(NM)$ になる.

参考文献

- [1] 大鐘武雄, 西村寿彦, 小川恭孝, “[招待講演] 確率伝搬法による多次元信号検出,” 信学技報, vol. 117, no. 395, SIP2017-92, pp. 175–175, 2018 年 1 月.
スライド URL: <https://www.ieice.org/ess/sita/forum/article/2018/201801262000.pdf>