A Gaussian Tree Approximation for Integer Least-Squares

Jacob Goldberger
School of Engineering
Bar-Ilan University
goldbej@eng.biu.ac.il

Amir Leshem
School of Engineering
Bar-Ilan University
leshema@eng.biu.ac.il

Abstract

This paper proposes a new algorithm for the linear least squares problem where the unknown variables are constrained to be in a finite set. The factor graph that corresponds to this problem is very loopy; in fact, it is a complete graph. Hence, applying the Belief Propagation (BP) algorithm yields very poor results. The algorithm described here is based on an optimal tree approximation of the Gaussian density of the unconstrained linear system. It is shown that even though the approximation is not directly applied to the exact discrete distribution, applying the BP algorithm to the modified factor graph outperforms current methods in terms of both performance and complexity. The improved performance of the proposed algorithm is demonstrated on the problem of MIMO detection.

1 Introduction

Finding the linear least squares fit to data is a well-known problem, with applications in almost every field of science. When there are no restrictions on the variables, the problem has a closed form solution. In many cases, a-priori knowledge on the values of the variables is available. One example is the existence of priors, which leads to Bayesian estimators. Another example of great interest in many applications is when the variables are constrained to a discrete finite set. This problem has many diverse applications such as decoding of multi-input-multi-output (MIMO) digital communication systems, GPS system ambiguity resolution [15] and many lattice problems in computer science, such as finding the closest vector in a lattice to a given point in R^n [1], and vector subset sum problems which have applications in cryptography [11]. In contrast to the continuous linear least squares problem, this problem is known to be NP hard.

This paper concentrates on the MIMO application. It should be noted, however, that the proposed method is general and can be applied to any integer linear least-square problem. A multiple-input-multiple-output (MIMO) is a communication system with n transmit antennas and m receive antennas. The tap gain from transmit antenna i to receive antenna j is denoted by $\mathbf{H}_{ij}$. In each use of the MIMO channel a vector $x = (x_1, ..., x_n)^\top$ is independently selected from a finite set of points $\mathcal{A}$ according to the data to be transmitted, so that $x \in \mathcal{A}^n$. A standard example of a finite set $\mathcal{A}$ in MIMO communication is $\mathcal{A} = \{-1, 1\}$ or more generally $\mathcal{A} = \{\pm 1, \pm 3, ..., \pm(2k+1)\}$. The received vector y is given by:

$$y = \mathbf{H}x + \epsilon \tag{1}$$

The vector ϵ is an additive noise in which the noise components are assumed to be zero mean, statistically independent Gaussians with a known variance $\sigma^2 I$. The $m \times n$ matrix $\mathbf{H}$ is assumed to be known. (In the MIMO application we further assume that $\mathbf{H}$ comprises iid elements drawn from a normal distribution of unit variance.) The MIMO detection problem consists of finding the unknown transmitted vector x given $\mathbf{H}$ and y. The task, therefore, boils down to solving a linear system in which the unknowns are constrained to a discrete finite set. Since the noise ϵ is assumed

to be additive Gaussian, the optimal maximum likelihood (ML) solution is:

$$\hat{x} = \arg\min_{x \in \mathcal{A}^n} \|\mathbf{H}x - y\|^2 \tag{2}$$

However, going over all the $|\mathcal{A}|^n$ vectors is unfeasible when either n or $|\mathcal{A}|$ are large.

A simple sub-optimal solution is based on a linear decision that ignores the finite set constraint:

$$z = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top y \tag{3}$$

and then, neglecting the correlation between the symbols, finding the closest point in $\mathcal{A}$ for each symbol independently:

$$\hat{x}_i = \arg\min_{a \in \mathcal{A}} |z_i - a| \tag{4}$$

This scheme performs poorly due to its inability to handle ill-conditioned realizations of the matrix $\mathbf{H}$. Somewhat better performance can be obtained by using a minimum mean square error (MMSE) Bayesian estimation on the continuous linear system. Let e be the variance of a uniform distribution over the members of $\mathcal{A}$. We can partially incorporate the information that $x \in \mathcal{A}^n$ by using the prior Gaussian distribution $x \sim \mathcal{N}(0, eI)$. The MMSE estimation becomes:

$$E(x|y) = (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1} \mathbf{H}^\top y \tag{5}$$

and then the finite-set solution is obtained by finding the closest lattice point in each component independently. A vast improvement over the linear approaches described above can be achieved by using sequential decoding:

- Apply MMSE (5) and choose the most reliable symbol, i.e. the symbol that corresponds to the column with the minimal norm of the matrix:
$$(\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1} \mathbf{H}^\top$$
- Make a discrete symbol decision for the most reliable symbol $\hat{x}_i$. Eliminate the detected symbol: $\sum_{j \neq i} \mathbf{h}_j x_j = y - \mathbf{h}_i \hat{x}_i$ ($\mathbf{h}_j$ is the j-th column of $\mathbf{H}$) to obtain a new smaller linear system. Go to the first step to detect the next symbol.

This algorithm, known as MMSE-SIC [5], has the best performance for this family of linear-based algorithms but the price is higher complexity. These linear type algorithms can also easily provide probabilistic (soft-decision) estimates for each symbol. However, there is still a significant gap between the detection performance of the MMSE-SIC algorithm and the performance of the optimal ML detector.

Many alternative structures have been proposed to approach ML detection performance. For example, sphere decoding algorithm (an efficient way to go over all the possible solutions) [7], approaches using the sequential Monte Carlo framework [3] and methods based on semidefinite relaxation [17] have been implemented. Although the detection schemes listed above reduce computational complexity compared to the exhaustive search of ML solution, sphere decoding is still exponential in the average case [9] and the semidefinite relaxation is a high-degree polynomial. Thus, there is still a need for low complexity detection algorithms that can achieve good performance.

This study attempts to solve the integer least-squares problem using the Belief Propagation (BP) paradigm. It is well-known (see e.g. [14]) that a straightforward implementation of the BP algorithm to the MIMO detection problem yields very poor results since there are a large number of short cycles in the underlying factor graph. In this study we introduce a novel approach to utilize the BP paradigm for MIMO detection. The proposed variant of the BP algorithm is both computationally efficient and achieves near optimal results.

2 The Loopy Belief Propagation Approach

Given the constrained linear system $y = \mathbf{H}x + \epsilon$, and a uniform prior distribution on x, the posterior probability function of the discrete random vector x given y is:

$$p(x|y) \propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{H}x - y\|^2) \qquad , \qquad x \in \mathcal{A}^n \tag{6}$$

The notation $\propto$ stands for equality up to a normalization constant. Observing that $\|\mathbf{H}x - y\|^2$ is a quadratic expression, it can be easily verified that $p(x|y)$ is factorized into a product of two- and single-variable potentials:

$$p(x_1, .., x_n|y) \propto \prod_i \psi_i(x_i) \prod_{i<j} \psi_{ij}(x_i, x_j) \tag{7}$$

such that

$$\psi_i(x_i) = \exp(-\frac{1}{2\sigma^2} y^\top \mathbf{h}_i x_i) \qquad , \qquad \psi_{ij}(x_i, x_j) = \exp(-\frac{1}{\sigma^2} \mathbf{h}_i^\top \mathbf{h}_j x_i x_j) \tag{8}$$

where $\mathbf{h}_i$ is the i-th column of the matrix $\mathbf{H}$. Since the obtained factors are simply a function of pairs, we obtain a Markov Random Field (MRF) representation [18]. In the MIMO application the (known) matrix $\mathbf{H}$ is randomly selected and therefore, the MRF graph is usually a completely connected graph.

In a loop-free MRF graph the max-product variant of the BP algorithm always converges to the most likely configuration (which corresponds to ML decoding in our case). For loop-free graphs, BP is essentially a distributed variant of dynamic programming. The BP message update equations only involve passing messages between neighboring nodes. Computationally, it is thus straightforward to apply the same local message updates in graphs with cycles. In most such models, however, this loopy BP algorithm will not compute exact marginal distributions; hence, there is almost no theoretical justification for applying the BP algorithm. (One exception is that, for Gaussian graphs, if BP converges, then the means are correct [16]). However, the BP algorithm applied to loopy graphs has been found to have outstanding empirical success in many applications, e.g., in decoding LDPC codes [6]. The performance of BP in this application may be attributed to the sparsity of the graphs. The cycles in the graph are long, hence the graph have tree-like properties, so that messages are approximately independent and inference may be performed as though the graph was loop-free. The BP algorithm has also been used successfully in image processing and computer vision (e.g. [4]) where the image is represented using a grid-structured MRF that is based on local connections between neighboring nodes.

However, when the graph is not sparse, and is not based on local grid connections, loopy BP almost always fails to converge. Unlike the sparse graphs of LDPC codes, or grid graphs in computer vision applications, the MRF graphs of MIMO channels are *completely connected graphs* and therefore the associated detection performance is poor. This has prevented the BP from being an asset for the MIMO problem. Fig. 1 shows an example of a MIMO real-valued system based on an 8×8 matrix and $\mathcal{A} = \{-1, 1\}$ (see the experiment section for a detailed description of the simulation set-up). As can be seen in Fig. 1, the BP decoder based on the MRF representation (7) has very poor results. Standard techniques to stabilize the BP iterations such as damping the message updates do not help here. Even applying more advanced versions of BP (e.g. Generalized BP and Expectation Propagation) to inference problems on complete MRF graphs yields poor results [12]. The problem here is not in the optimization method but in the cost function that needs to be modified yield a good approximate solution.

There have been several recent attempts to apply BP to the MIMO detection problem with good results (e.g. [8, 10]). However in the methods proposed in [8] and [10] the factorization of the probability function is done in such a way that each factor corresponds to a single linear equation. This leads to a partition of the probability function into factors each of which is a function of all the unknown variables. This leads to exponential computational complexity in computing the BP messages. Shental et. al [14] analyzed the case where the matrix $\mathbf{H}$ is relatively sparse (and has a grid structure). They showed that even under this restricted assumption the BP still does not perform well. As an alternative method they proposed the generalized belief propagation (GBP) algorithm that does work well on the sparse matrix if the algorithm regions are carefully chosen. There are situations where the sparsity assumption makes sense (e.g. 2D intersymbol interference (ISI) channels). However, in the MIMO channel model we assume that the channel matrix elements are iid and Gaussian; hence we cannot assume that the channel matrix $\mathbf{H}$ is sparse.

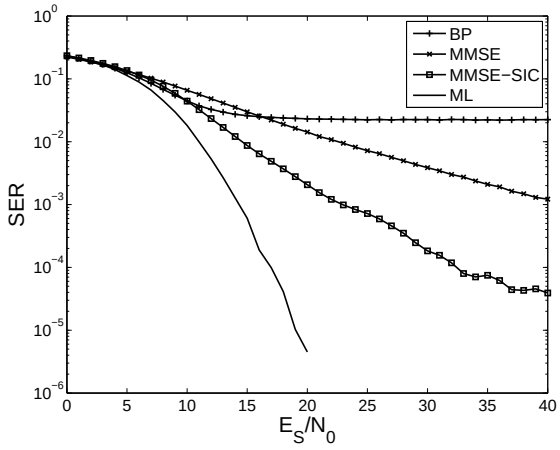


Figure 1: Decoding results for 8×8 system, $\mathcal{A} = \{-1, 1\}$.

3 The Tree Approximation of the Gaussian Density

Our approach is based on an approximation of the exact probability function:

$$p(x_1, .., x_n|y) \propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{H}x - y\|^2) \qquad , \qquad x \in \mathcal{A}^n \tag{9}$$

that enables a successful implementation of the Belief Propagation paradigm. Since the BP algorithm is optimal on loop-free factor graphs (trees) a reasonable approach is finding an optimal tree approximation of the exact distribution (9). Chow and Liu [2] proposed a method to find a tree approximation of a given distribution that has the minimum Kullback-Leibler distance to the actual distribution. They showed that the optimal tree can be learned efficiently via a maximum spanning tree whose edge weights correspond to the mutual information between the two variables corresponding to the edges endpoints. The problem is that the Chow-Liu algorithm is based on the (2-dimensional) marginal distributions. However, finding the marginal distribution of the probability function (9) is, unfortunately, NP hard and it is (equivalent to) our final target.

To overcome this obstacle, our approach is based on applying the Chow-Liu algorithm on the distribution corresponding to the unconstrained linear system. This distribution is Gaussian and therefore it is straightforward in this case to compute the (2-dimensional) marginal distributions. Given the Gaussian tree approximation, the next step of our approach is to apply the finite-set constraint and utilize the Gaussian tree distribution to form a discrete loop free approximation of $p(x|y)$ which can be efficiently globally maximized using the BP algorithm. To motivate this approach we first apply a simplified version to derive the linear solution (4) described in Section 2.

Let $z(y) = (\mathbf{H}^\top \mathbf{H})^{-1}\mathbf{H}^\top y$ be the least-squares estimator (3) and $\mathbf{C} = \sigma^2(\mathbf{H}^\top \mathbf{H})^{-1}$ is its variance. It can be easily verified that $p(x|y)$ (9) can be written as:

$$p(x|y) \propto f(x; z, C) = \exp(-\frac{1}{2}(z - x)^\top C^{-1}(z - x)) \tag{10}$$

where $f(x; z, C)$ is a Gaussian density with mean z and covariance matrix C (to simplify notation we ignore hereafter the constant coefficient of the Gaussian densities). Now, instead of marginalizing the true distribution $p(x|y)$, which is an NP hard problem, we approximate it by the product of the marginals of the Gaussian density $f(x; z, C)$:

$$f(x; z, C) \approx \prod_i f(x_i; z_i, C_{ii}) = \exp(-\frac{(z_i - x_i)^2}{2\mathbf{C}_{ii}}) \tag{11}$$

From the Gaussian approximation (11) we can extract a discrete approximation:

$$\hat{p}(x_i = a|y) \propto f(x_i; z_i, C_{ii}) = \exp(-\frac{(z_i - a)^2}{2\mathbf{C}_{ii}}) \quad , \quad a \in \mathcal{A} \tag{12}$$

Input: A constrained linear LS problem: $\mathbf{H}x + \epsilon = y$, a noise level σ^2 and a finite symbol set $\mathcal{A}$.

Goal: Find (approx. to) $\arg\min_{x \in \mathcal{A}^n} \|\mathbf{H}x - y\|^2$

Algorithm:

- Compute $z = (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1} \mathbf{H}^\top y$ and $\mathbf{C} = \sigma^2 (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1}$.
- Denote:

$$\begin{aligned} f(x_i; z, C) &= \exp(-\frac{1}{2} \frac{(x_i - z_i)^2}{C_{ii}}) \\ f(x_i | x_j; z, C) &= \exp(-\frac{1}{2} \frac{((x_i - z_i) - C_{ij}/C_{jj}(x_j - z_j))^2}{C_{ii} - C_{ij}^2/C_{jj}}) \end{aligned}$$

- compute maximum spanning tree of the n-node graph where the weight of the i-j edge is the square of the correlation coefficient:

$$\rho_{ij}^2 = C_{ij}^2 / (C_{ii} C_{jj})$$

  Assume the tree is rooted at node '1' and denote the parent of node i by $p(i)$.

- Apply BP on the loop free MRF:

$$\hat{p}(x_1, ..., x_n | y) \propto f(x_1; z, C) \prod_{i=2}^{n} f(x_i | x_{p(i)}; z, C) \qquad x_1, ..., x_n \in \mathcal{A}$$

  to find the (approx. to the) most likely configuration.

Figure 2: The Gaussian Tree Approximation (GTA) Algorithm.

Taking the most likely symbol we obtain the sub-optimal linear solution (4).

Motivated by the simple product-of-marginals approximation described above, we suggest approximating the discrete distribution $p(x|y)$ via a tree-based approximation of the Gaussian distribution $f(x; z, C)$. Although the Chow-Liu algorithm was originally stated for discrete distributions, one can easily verify that it also applies for the Gaussian case. Let

$$I(x_i; x_j) = \log C_{ii} + \log C_{jj} - \log \begin{vmatrix} C_{ii} & C_{ij} \\ C_{ji} & C_{jj} \end{vmatrix} = -\log(1 - \rho_{ij}^2)$$

be the mutual information of x_i and x_j based on the Gaussian distribution $f(x; z, C)$, where ρ_{ij} is the correlation coefficient between x_i and x_j. Let $\hat{f}(x)$ be the optimal Chow-Liu tree approximation of $f(x; z, C)$. We can assume, without loss of generality, that $\hat{f}(x)$ is rooted at x_1. $\hat{f}(x)$ is a loop-free Gaussian distribution on $x_1, ..., x_n$, i.e.

$$\hat{f}(x) = f(x_1; z, C) \prod_{i=2}^{n} f(x_i | x_{p(i)}; z, C) \qquad , \qquad x \in R^n \tag{13}$$

where $p(i)$ is the 'parent' of the i-th node in the optimal tree. The Chow-Liu algorithm guarantees that $\hat{f}(x)$ is the optimal Gaussian tree approximation of $f(x; z, C)$ in the sense that the KL divergence $D(f||\hat{f})$ is minimal among all the Gauss-Markov distributions on R^n. We note in passing that applying a monotonic function on the graph weights does not change the topology of the optimal tree. Hence to find the optimal tree we can use the weights ρ_{ij}^2 instead of $-\log(1-\rho_{ij}^2)$. The optimal tree, therefore is one that maximizes the sum of the square correlation coefficients between adjacent nodes.

Our approximation approach is, therefore, based on replacing the true distribution $p(x|y)$ with the following approximation:

$$\hat{p}(x_1,...,x_n|y) \propto \hat{f}(x) = f(x_1;z,C)\prod_{i=2}^{n} f(x_i|x_{p(i)};z,C) \qquad , \qquad x \in \mathcal{A}^n \tag{14}$$

The probability function $\hat{p}(x|y)$ is a loop free factor graph. Hence the BP algorithm can be applied to find its most likely configuration. An optimal BP schedule requires passing a message once for each direction of each edge. The BP messages are first sent from leaf variables to the root and then back to the leaves. We demonstrate empirically in the experiment section that the optimal solution of $\hat{p}(x|y)$ is indeed nearly optimal for $p(x|y)$.

The MMSE Bayesian approach (5) is known to be better than the linear based solution (4). In a similar way we can consider a Bayesian version of the proposed Gaussian tree approximation. We can partially incorporate the information that $x \in \mathcal{A}^n$ by using the prior Gaussian distribution $x \sim \mathcal{N}(0, eI)$ such that $e = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} a^2$. This yields the posterior Gaussian distribution:

$$\begin{aligned} f_{(x|y)}(x|y) &\propto \exp(-\frac{1}{2e}\|x\|^2 - \frac{1}{2\sigma^2}\|\mathbf{H}x - y\|^2) \\ &\propto \exp(-\frac{1}{2}(x - E(x|y))^\top(\mathbf{H}^\top\mathbf{H} + \frac{\sigma^2}{e}I)(x - E(x|y)) \end{aligned} \tag{15}$$

such that $E(x|y) = (\mathbf{H}^\top\mathbf{H} + \frac{\sigma^2}{e}I)^{-1}\mathbf{H}^\top y$. We can apply the Chow-Liu tree approximation on the Gaussian distribution (15) to obtain a 'Bayesian' Gaussian tree approximation for $p(x|y)$. One can expect that this yields is a better approximation of the discrete distribution $p(x|y)$ than the tree distribution that is based on the unconstrained distribution $f(x;z,C)$ since it partially includes the finite-set constraint. We show in Section 4 that the Bayesian version indeed yields better results.

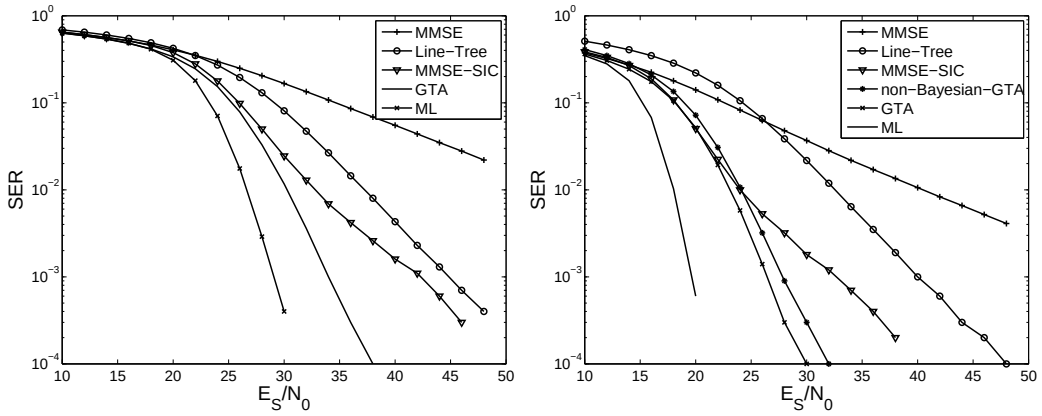
To summarize, our solution to the constrained least squares problem is based on applying BP on a Gaussian tree approximation of the Bayesian version of the continuous least-square case. We dub this method "The Gaussian-Tree-Approximation (GTA) Algorithm". The GTA algorithm is summarized in Fig. 3. We next compute the complexity of the GTA algorithm. The complexity of computing the covariance matrix $(\mathbf{H}^\top\mathbf{H} + \frac{\sigma^2}{e}I)^{-1}$ is $O(n^3)$, the complexity of the Chow-Liu algorithm (based on Prim's algorithm for finding the minimum spanning tree) is $O(n^2)$ and the complexity of the BP algorithm is $O(|\mathcal{A}|^2 n)$.

4 Experimental Results

In this section we provide simulation results for the GTA algorithm over various MIMO systems. We assume a frame length of 100, i.e. the channel matrix $\mathbf{H}$ is constant for 100 channel uses. The channel matrix comprised iid elements drawn from a zero-mean normal distribution of unit variance. We used 10^4 realizations of the channel matrix. This resulted in 10^6 vector messages. The performance of the proposed algorithm is shown as a function of the variance of the additive noise σ^2. The signal-to-noise ratio (SNR) is defined as $10\log_{10}(E_s/N_0)$ where $E_s/N_0 = \frac{ne}{\sigma^2}$ (n is the number of variables, σ^2 is the variance of the Gaussian additive noise, and e is the variance of the uniform distribution over the discrete set $\mathcal{A}$).

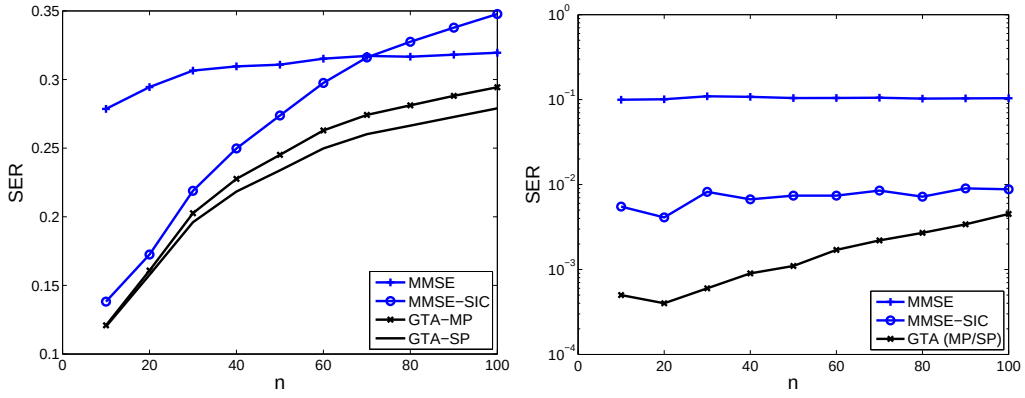
Fig. 3 shows the symbol error rate (SER) versus SNR for a 10×10, $|\mathcal{A}| = 8$, MIMO system and for a 20×20, $|\mathcal{A}| = 4$, MIMO system. Note that the algorithm was applied in Fig. 3 to a real world practical application (MIMO communication) using real world parameters. Unlike other areas (e.g computer vision, bioinformatics) here the real world performance analysis is based on extensive simulations of the communication channel. Note that a 20×20 fully connected MRF is not a small problem and unlike the Potts model that is defined on a grid MRF, the BP and it variants do not work here. The performance of the GTA method was compared to the MMSE and the MMSE-SIC algorithms (see Section 2). The GTA algorithm differs from these algorithms in two ways. The first is a Markovian approximation of $f(x;z,C)$ instead of a product of independent densities. The second aspect is utilizing the optimal tree. To clarify the contribution of each component we modified the GTA algorithm by replaced the Chow-Liu optimal tree by the tree $1 \to 2 \to 3, ..., \to n$. We call this method the 'Line-Tree'. As can be seen from Fig. 3, using the optimal tree is crucial

to obtain improved results. Fig. 3b also shows results of the non-Bayesian variant of the GTA algorithm. As can be seen, the Bayesian version yields better results. In Fig. 3a the two versions yield the same results. It can be seen that the performance of the GTA algorithm is significantly better than the MMSE-SIC (and its computational complexity is much smaller).



(a) $10 \times 10, |\mathcal{A}| = 8$ (b) $20 \times 20, |\mathcal{A}| = 4$

Figure 3: Comparative results of MMSE, MMSE-SIC and variants of the GTA.



(a) noise level: $\sigma^2 = 2.5$ (b) noise level:$\sigma^2 = 0.25$

Figure 4: Comparative results of MMSE, MMSE-SIC and the GTA approximation followed by the sum-product and max-product variants of the BP algorithm. The alphabet size is $|\mathcal{A}| = 8$ and the results are shown as a function of the matrix size $n \times n$.

Fig. 4 depicts comparative performance results as a function of n, the size of the linear system. The alphabet size in all the experiments was $|\mathcal{A}| = 8$ and as in Fig. 3 each experiment was repeated $10^4 \times 10^2$ times. The performance of the GTA method was compared to the MMSE and the MMSE-SIC algorithms (see Section 2). In Fig. 4a the noise variance was set to $\sigma^2 = 2.5$ and in Fig. 4b to $\sigma^2 = 0.25$. In all cases the GTA was found to be better than the MMSE-SIC. The GTA algorithm is based on an optimal Gaussian tree approximation followed by a BP algorithm. There are two variants of the BP, namely the max-product (MP) and the sum-product (SP). Since the performance is measured in symbol error-rate and not frame error-rate the SP should yield improved results. Note that if the exact distribution was loop-free then SP would obviously be the optimal method when the error is measured in number of symbols. However, since the BP is applied to an approximated distribution the superiority of the SP is not straightforward. When the noise level is relatively high the sum-product version is better than the max-product. When the noise level is lower there is no significant difference between the two BP variants. Note that from an algorithmic point of view, the MP unlike the SP, can be easily computed in the log domain.

5 Conclusion

Solving integer linear least squares problems is an important issue in many fields. We proposed a novel technique based on the principle of a tree approximation of the Gaussian distribution that corresponds to the continuous linear problem. The proposed method improved performance compared to all other polynomial algorithms for solving the problem as demonstrated in simulations. As far as we know this is the first successful attempt to apply the BP paradigm to completely connected MRF. A main concept in the GTA model is the interplay between discrete and Gaussian models. Such hybrid ideas can be considered also for discrete inference problems other than least-squares. One example is the work of Opper and Winther who applied an iterative algorithm using a model which is seen as discrete and Gaussian in turn to address Ising model problems [13]. Although the focus of this paper is on an approach based on tree approximation, more complicated approximations such as multi-parent trees have potential to improve performance and can potentially provide a smooth performance-complexity trade-off. Although the proposed method yields improved results, the tree approximation we applied nay not be the best one (finding the best tree for the integer constrained linear problem is NP hard). It is left for future research to search for a better discrete tree approximation for the constrained linear least squares problem.

References

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.

[2] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Info. Theory*, pages 462–467, 1968.

[3] B. Dong, X. Wang, and A. Doucet. A new class of soft MIMO demodulation algorithms. *IEEE Trans. Signal Processing*, pages 2752–63, 2003.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, pages 41–54, 2006.

[5] G.J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell Labs Technical Journal*, 1(2):41–59, 1996.

[6] R. G. Gallager. Low density parity check codes. *IRE Trans. Inform.Theory*, pages 21–28, 1962.

[7] B. M. Hochwald and S. ten Brink. Achieving near-capacity on a multiple antenna channel. *IEEE Trans. Commun.*, pages 389–399, 2003.

[8] J. Hu and T. M. Duman. Graph-based detector for BLAST architecture. *IEEE Communications Society ICC*, 2007.

[9] J. Jalden and B. Ottersten. An exponential lower bound on the expected complexity of sphere decoding. *IEEE Intl. Conf. Acoustic, Speech, Signal Processing*, 2004.

[10] M. Kaynak, T. Duman, and E. Kurtas. Belief propagation over SISO/MIMO frequency selective fading channels. *IEEE Transactions on Wireless Communications*, pages 2001–5, 2007.

[11] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *J. ACM*, 32(1):229–246, 1985.

[12] T. Minka and Y. Qi. Tree-structured approximations by expectation propagation. *Advances in Neural Information Processing Systems*, 2004.

[13] M. Opper and O. Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, pages 2177–2204, 2005.

[14] O. Shental, N. Shental, S. Shamai (Shitz), I. Kanter, A.J. Weiss, and Y. Weiss. Discrete-input two-dimensional gaussian channels with memory: Estimation and information rates via graphical models and statistical mechanics. *Information Theory, IEEE Transactions on*, pages 1500–1513, 2008.

[15] P.J.G. Teunissen. Success probability of integer GPS ambiguity rounding and bootstrapping. *Journal of Geodesy*, 72:606–612, 1998.

[16] Y. Weiss and W.T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, pages 2173–2200, 2001.

[17] A. Wiesel, Y. C. Eldar, and S. Shamai. Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels. *IEEE Signal Processing Letters*, 2005.

[18] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *IJCAI*, 2001.