A message-passing solver for linear systems

Ori Shental, Danny Bickson, Paul H. Siegel, Jack K. Wolf and Danny Dolev

***Abstract*— We develop an efficient distributed message-passing solution for systems of linear equations based upon Gaussian belief propagation that does not involve direct matrix inversion.**

Solving a system of linear equations $\mathbf{Ax} = \mathbf{b}$ is one of the most fundamental problems in algebra, with countless applications in the mathematical sciences and engineering. Given the observation vector $\mathbf{b} \in \mathbb{R}^n, n \in \mathbb{N}^*$, and the data matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, a unique solution, $\mathbf{x} = \mathbf{x}^* \in \mathbb{R}^n$, exists if and only if the data matrix $\mathbf{A}$ is full rank. In this contribution we concentrate on the popular case where the data matrices, $\mathbf{A}$, are also symmetric.

We can translate the problem of solving the linear system from the algebraic domain to the domain of probabilistic inference, as stated in the following theorem.

Proposition 1 (Solution and inference): The computation of the solution vector $\mathbf{x}^*$ is identical to the inference of the vector of marginal means $\mu = \{\mu_1, \dots, \mu_n\}$ over the graph $\mathcal{G}$ with the associated joint Gaussian probability density function $p(\mathbf{x}) \sim \mathcal{N}(\mu \triangleq \mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1})$.

The move to the probabilistic domain calls for the utilization of belief propagation (BP) as an efficient inference engine.

The BP algorithm functions by passing real-valued messages across edges in the graph and consists of two computational rules, namely the 'sum-product rule' and the 'product rule'. Gaussian BP is a special case of continuous BP, where the underlying distribution is Gaussian.

The message $m_{ij}(x_j)$, sent from node i to node j over their shared edge on the graph, is given by

$$m_{ij}(x_j) \propto \int_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k\in \mathrm{N}(i)\setminus j} m_{ki}(x_i)dx_i. \quad (1)$$

The marginals are computed according to the product rule

$$p(x_i) = \alpha\phi_i(x_i) \prod_{k\in \mathrm{N}(i)} m_{ki}(x_i), \quad (2)$$

where the scalar α is a normalization constant. The graph potentials $\psi_{ij}(x_i, x_j)$ and $\phi_i(x_i)$ are simply determined according to a pairwise factorization of the Gaussian distribution. The set of graph nodes $\mathrm{N}(i)$ denotes the set of all the nodes neighboring the ith node. The set $\mathrm{N}(i)\setminus j$ excludes the node j from $\mathrm{N}(i)$.

Looking at the right hand side of the integral-product rule (1), node i needs to first calculate the product of all incoming messages, except for the message coming from node j. Recall that since $p(\mathbf{x})$ is jointly Gaussian, the factorized self potentials $\phi_i(x_i) \propto \mathcal{N}(\mu_{ii}, P_{ii}^{-1})$ and similarly all messages $m_{ki}(x_i) \propto \mathcal{N}(\mu_{ki}, P_{ki}^{-1})$ are of Gaussian form as well.

O. Shental, P. H. Siegel and J. K. Wolf are with CMRR, UCSD (Email: {oshental,psiegel,jwolf}@ucsd.edu). D. Bickson and D. Dolev are with CSE, HUJI, Israel (Email: {daniel51,dolev}@cs.huji.ac.il).

As the terms in the product of the incoming messages and the self potential in the integral-product rule (1) are all a function of the same variable, x_i (associated with the node i), then, according to the lemma of product of Gaussian densities, $\phi_i(x_i)\prod_{k\in\mathrm{N}(i)\setminus j} m_{ki}(x_i)$ is proportional to a certain Gaussian distribution, $\mathcal{N}(\mu_{i\setminus j}, P_{i\setminus j}^{-1})$. Thus, the update rule for the inverse variance is given by (over-braces denote the origin of each of the terms)

$$P_{i\setminus j} = \overbrace{P_{ii}}^{\phi_i(x_i)} + \sum_{k\in\mathrm{N}(i)\setminus j} \overbrace{P_{ki}}^{m_{ki}(x_i)}, \quad (3)$$

where $P_{ii} \triangleq A_{ii}$ is the inverse variance a-priori associated with node i, via the precision of $\phi_i(x_i)$, and P_{ki} are the inverse variances of the messages $m_{ki}(x_i)$. Similarly, we can calculate the mean

$$\mu_{i\setminus j} = P_{i\setminus j}^{-1}\Big(\overbrace{P_{ii}\mu_{ii}}^{\phi_i(x_i)} + \sum_{k\in\mathrm{N}(i)\setminus j} \overbrace{P_{ki}\mu_{ki}}^{m_{ki}(x_i)}\Big), \quad (4)$$

where $\mu_{ii} \triangleq b_i/A_{ii}$ is the mean of the self potential and μ_{ki} are the means of the incoming messages.

Next, we calculate the remaining terms of the message $m_{ij}(x_j)$, including the integration over x_i. After some algebraic manipulation, using the Gaussian integral $\int_{-\infty}^{\infty} \exp{(-ax^2 + bx)}dx = \sqrt{\pi/a}\exp{(b^2/4a)}$, we find that the messages $m_{ij}(x_j)$ are proportional to a normal distribution with precision and mean

$$P_{ij} = -A_{ij}^2 P_{i\setminus j}^{-1}, \quad (5)$$

$$\mu_{ij} = -P_{ij}^{-1} A_{ij}\mu_{i\setminus j}. \quad (6)$$

These two scalars represent the messages propagated in the GaBP-based algorithm.

Finally, computing the product rule (2) is similar to the calculation of the previous product and the resulting mean (4) and precision (3), but including all incoming messages. The marginals are inferred by normalizing the result of this product. Thus, the marginals are found to be Gaussian probability density functions $\mathcal{N}(\mu_i, P_i^{-1})$ with precision and mean

$$P_i = \overbrace{P_{ii}}^{\phi_i(x_i)} + \sum_{k\in\mathrm{N}(i)} \overbrace{P_{ki}}^{m_{ki}(x_i)}, \quad (7)$$

$$\mu_i = P_{i\setminus j}^{-1}\Big(\overbrace{P_{ii}\mu_{ii}}^{\phi_i(x_i)} + \sum_{k\in\mathrm{N}(i)} \overbrace{P_{ki}\mu_{ki}}^{m_{ki}(x_i)}\Big), \quad (8)$$

respectively, where the latter gives the desired solution to the linear system.