Advanced Messaging Passing Algorithms

Dong Liu, Lars K. Rasmussen

October 10, 2019

Contents

1 Problem

The problem setting is that for observable signal $\boldsymbol{y}$, there is a true underlining signal $\boldsymbol{x}$ that results in the observation $\boldsymbol{y}$. Here $\boldsymbol{x}$ is assumed to lie in a discrete finite set, i.e. $\boldsymbol{x} \in \mathcal{A}^N$ and we denote its i-th element $x_i \in \mathcal{A}$. The observation lies in continuous real-valued space $\boldsymbol{y} \in \mathbb{R}^N$. The further assumption about the

relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$ is linear transformation with Gaussian noise $\boldsymbol{w}$, i.e. $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I})$. This can be formulated as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}, \boldsymbol{x} \in \mathcal{A}^N. \tag{1}$$

The optimal way to estimate $\boldsymbol{x}$ is to look for the maximum a posterior (MAP) to

$$p(\boldsymbol{x}|\boldsymbol{y}) \sim \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2 \boldsymbol{I}) \delta_{\boldsymbol{x} \in \mathcal{A}}. \tag{2}$$

But the search space can be too large to do MAP estimation as the search space increases exponentially with length of $\boldsymbol{x}$, i.e. N. To avoid the complex computation, approximation methods are usually carried out. Since the original posterior can be factorized into

$$p(\boldsymbol{x}|\boldsymbol{y}) = \prod_i t_i(\boldsymbol{x}). \tag{3}$$

Then, we can formulate a approximated distribution

$$q(\boldsymbol{x}) = \prod_i \tilde{t}_i(\boldsymbol{x}), \tag{4}$$

and estimate $\boldsymbol{x}$ by using $q(\boldsymbol{x})$.

2 Expectation Propagation (EP)

For the EP algorithm, we basically follow [2] to establish a baseline algorithm. In this algorithm, the approximated distribution is assumed to be:

$$q(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2 \boldsymbol{I}) \prod_{i=1}^{N} \exp\left(\gamma_i x_i - \frac{1}{2}\Lambda_i x_i^2\right), \tag{5}$$

where $\gamma_i \in \mathbb{R}$ and $\Lambda_i \in \mathbb{R}^+$, $\forall i$, are the parameters to be estimated in EP. It is clear that $q(\boldsymbol{x})$ in (5) belongs to exponential family, Gaussian distribution to be specific:

$$q(\boldsymbol{x}) \sim \mathcal{N}\left(\boldsymbol{x} : \boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \tag{6}$$

where we have its distribution parameter as

$$\boldsymbol{\Sigma} = \left(\sigma_w^2 \boldsymbol{H}^T \boldsymbol{H} + \mathrm{diag}(\boldsymbol{\Lambda})\right)^{-1}, \tag{7}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\left(\sigma_w^2 \boldsymbol{H}^T \boldsymbol{y} + \boldsymbol{\gamma}\right), \tag{8}$$

with $\mathrm{diag}(\Lambda)$ being a diagnal matrix of which $\Lambda_i, i = 1, 2, \cdots, N$, are diagonal elements, $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_N]^T$.

EP algorithm keeps updating the parameters of $q(\boldsymbol{x})$ until it meets the stop criteria, after which the most probable $\boldsymbol{x}$ is estimated by using updated $q(\boldsymbol{x})$. To be specific, the iterations of EP can be detailed as follows:

- Update the marginal distribution of the cavity distribution at iteration l:

$$q^{(l)\setminus i}(u_i) = \frac{q^{(l)}(u_i)}{\exp\left(\gamma_i^{(l)} u_i - \frac{1}{2}\Lambda_i^{(l)} u_i^2\right)}, \tag{9}$$

  which is a Gaussian distribution. Superscript $^{(l)}$ denotes the iteration. Let us denote this distribution as $q^{(l)\setminus i}(u_i) = \mathcal{N}\left(u_i : t_i^{(l)}, h_i^{2(l)}\right)$, with

$$h_i^{2(l)} = \frac{\sigma_i^{2(l)}}{1 - \sigma_i^{2(l)}\Lambda_i^{(l)}}, \tag{10}$$

$$t_i^{(l)} = h_i^{2(l)}\left(\frac{\mu_i^{(l)}}{\sigma_i^{2(l)}} - \gamma_i^{(l)}\right). \tag{11}$$

- Corresponding marginal distribution of cavity distribution of $p(\boldsymbol{x})$:

$$\hat{p}^{(l)}(u_i) \sim q^{(l)\setminus i}(u_i)\delta_{u_i \in \mathcal{A}_i}. \tag{12}$$

- Update the statistics parameter $(\boldsymbol{\gamma}, \boldsymbol{\Lambda})$ of q, where the marginal distribution:

$$q^{(l+1)\setminus i}(u_i) \sim q^{(l)\setminus i}(u_i)\exp\left\{\gamma_i^{(l+1)} u_i - \frac{1}{2}\Lambda_i^{(l+1)} u_i^2\right\}. \tag{13}$$

  Here $q^{(t+1)\setminus i}(u_i)$ is parameterized by $(\boldsymbol{\gamma}, \boldsymbol{\Lambda})$. This pair can be updated by:

$$\Lambda_i^{(l+1)} = \frac{1}{\sigma_{p_i}^{2(l)}} - \frac{1}{h_i^{2(l)}} \tag{14}$$

$$\gamma_i^{(l+1)} = \frac{\mu_{p_i}^{(t)}}{\sigma_{p_i}^{2(t)}} - \frac{t_i^{(t)}}{h_i^{2(t)}}. \tag{15}$$

When stop criteria is met, estimation is made by using $q(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{x} : \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$, where

$$\boldsymbol{\Sigma}^* = \left(\sigma_w^2 \boldsymbol{H}^T \boldsymbol{H} + \mathrm{diag}(\boldsymbol{\Lambda}^*)\right)^{-1}, \tag{16}$$

$$\boldsymbol{\mu}^* = \boldsymbol{\Sigma}^*\left(\sigma_w^2 \boldsymbol{H}^T \boldsymbol{y} + \boldsymbol{\gamma}^*\right) \tag{17}$$

are returned by the above EP iterations. The decision is made by:

$$\hat{\mu}_i = \underset{\mu_i \in \mathcal{A}}{\mathrm{argmin}} \, |\mu_i - \mu_i^*| \tag{18}$$

3 Power EP

In this section, we explain a variant of EP algorithm, known as power EP in literature [10]. Power EP has similar procedure of update as EP has. We would mainly discuss the difference here.

The power EP algorithm stems from a alpha-deiverge:

$$D_\alpha(p\|q) = \frac{4}{1-\alpha^2}\left(1 - \int_x p^{(1+\alpha)/2} q^{(1-\alpha)/2}\right) \tag{19}$$

The meta algorithm of power EP can be summarize in high level steps as follows.

- Initialize $q(\boldsymbol{x}) \sim t_0(\boldsymbol{x}) \prod_i \tilde{t}_i(x_i)$
- Repeat:

$$q^{\backslash i}(\boldsymbol{x}) = q(\boldsymbol{x})/\tilde{t}_i(x_i), \forall i, \tag{20}$$

$$q^{\text{new}}(\boldsymbol{x}) = \underset{q}{\text{argmin}}\, D_\alpha\left(t_i(x_i)q^{\backslash i}(\boldsymbol{x}) \| \tilde{t}_i(x_i)q^{\backslash i}(\boldsymbol{x})\right) \tag{21}$$

$$\tilde{t}_i^{\text{new}}(x_i) = q^{\text{new}}(\boldsymbol{x})/q^{\backslash i}(\boldsymbol{x}), \forall i. \tag{22}$$

To apply the above power EP algorithm into the problem in Section 1. We also need the equivalence condition as follows:

$$\underset{q}{\text{argmin}}\, D_\alpha\left(t_i(x_i)q^{\backslash i}(\boldsymbol{x}) \| \tilde{t}_i(x_i)q^{\backslash i}(\boldsymbol{x})\right) = \underset{q}{\text{argmin}}\, KL\left(f_i(x_i)q^{\backslash \tilde{f}_i}(\boldsymbol{x}) \| q(\boldsymbol{x})\right), \tag{23}$$

where

$$f_i(x_i) = [t_i(x_i)]^{1/n} \tag{24}$$

$$\alpha = 2/n - 1. \tag{25}$$

By applying the above rules to the problem in Section 1, then

$$f_i(x_i) = [t_i(x_i)]^{1/n} = [\delta_{x_i \in \mathcal{A}}]^{1/n} = \delta_{x_i \in \mathcal{A}}, \tag{26}$$

$$\tilde{f}_i(x_i) = [\tilde{t}_i(x_i)]^{1/n} = \exp\left(\frac{\gamma_i x_i - \frac{1}{2}\Lambda_i x_i^2}{n}\right). \tag{27}$$

Assume $\varphi(x_i)$ is the sufficient statistics of variables x_i. Solving problem in (23) gives

$$\mathbb{E}_{q^{\backslash \tilde{f}_i}(\boldsymbol{x})\tilde{f}_i^{\text{new}}(x_i)}[\varphi(x_i)] = \mathbb{E}_{q^{\backslash \tilde{f}_i}(\boldsymbol{x})f_i(x_i)}[\varphi(x_i)], \tag{28}$$

which is equivalent to

$$\int_{\boldsymbol{x}} q^{\backslash \tilde{f}_i}(x)\tilde{f}_i^{\text{new}}(x_i)\varphi(x_i)d\boldsymbol{x} = \int_{\boldsymbol{x}} q^{\backslash \tilde{f}_i}(x)f_i(x_i)\varphi(x_i)d\boldsymbol{x} \tag{29}$$

$$= \int_{\boldsymbol{x}} q^{\backslash \tilde{f}_i}(x)\delta_{x_i \in \mathcal{A}}\varphi(x_i)d\boldsymbol{x}. \tag{30}$$

Simplifying the above on marginalization gives

$$\int_{x_i} q_i^{\backslash \tilde{f}_i}(x_i)\tilde{f}_i^{\text{new}}(x_i)\varphi(x_i) = \int_{x_i} \delta_{x_i \in \mathcal{A}} q_i^{\backslash \tilde{f}_i}(x_i)\varphi(x_i)dx_i, \tag{31}$$

where

$$q_i^{\setminus \tilde{f}_i}(x_i) = \int \frac{q(\boldsymbol{x})}{\tilde{f}_i(x_i)} dx_1 dx_2 \cdots dx_{i-1} dx_{i+1} \cdots dx_n. \tag{32}$$

Similar to Section 2, we use notation

$$\int_{x_i} \delta_{x_i \in \mathcal{A}} q_i^{\setminus \tilde{f}_i}(x_i) \varphi(x_i) dx_i = [\mu_{p_i}, \sigma_{p_i}^2]^T, \forall i. \tag{33}$$

We summarize the algorithm steps for power EP as follow:

- Update the marginal distribution of the cavity distribution at iteration t:

$$q^{(l)\setminus i}(u_i) = \frac{q^{(l)(u_i)}}{\tilde{f}_i(x_i)} \sim \mathcal{N}\left(x_i; t_i, h_i^2\right), \tag{34}$$

  where

$$h_i^{2(l)} = \frac{\sigma_i^{2(l)}}{1 - \sigma_i^{2(l)} \Lambda_i^{(l)}/n}, \tag{35}$$

$$t_i^{(l)} = h_i^{2(l)} \left(\frac{\mu_i^{(l)}}{\sigma_i^{2(l)}} - \gamma_i^{(l)}/n \right). \tag{36}$$

- Compute the statics mean $\mu_{p_i}^{(l)}$ and variance $\sigma_{p_i}^{2(l)}$ according to (33) at iteration l.

- Update the statistics parameter $(\boldsymbol{\gamma}, \boldsymbol{\Lambda})$ of q, such that the statics of

$$q^{(l+1)\setminus \tilde{f}_i}(x_i) \tilde{f}_i^{(l+1)} \sim \exp\left\{ -\frac{(x_i - t_i)^2}{2h_i^2} \right\} \exp\left\{ \gamma_i^{(l+1)} u_i - \frac{1}{2} \Lambda_i^{(l+1)} u_i^2 \right\}, \tag{37}$$

  is the same as $\mu_{p_i}^{(l)}$, $\sigma_{p_i}^{2(l)}$. Solving this moments matching problem gives

$$\Lambda_i^{(l+1)} = n \left[\frac{1}{\sigma_{p_i}^{2(l)}} - \frac{1}{h_i^{2(l)}} \right] \tag{38}$$

$$\gamma_i^{(l+1)} = n \left[\frac{\mu_{p_i}^{(t)}}{\sigma_{p_i}^{2(t)}} - \frac{t_i^{(t)}}{h_i^{2(t)}} \right]. \tag{39}$$

The detection is similar to EP after power EP iterations.

4 Improvement on EP

In this section, some improvement on EP is used following Opper's work in [13]. The intuition of improving EP comes from that the partition function of EP approximation should be the as close as possible to the partition of original

function. In this section we still follow the same notation for model distribution and approximate distortion as previous sections:

$$\text{Model distribution }: p(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2 \boldsymbol{I}) \prod_{i=1}^{N} \delta_{x_i \in \mathcal{A}} = t_0(\boldsymbol{x}) \prod_{i=1}^{N} t_i(x_i) \tag{40}$$

$$\text{Approximate }: q(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2 \boldsymbol{I}) \prod_{i=1}^{N} \exp\left(\gamma_i x_i - \frac{1}{2}\Lambda_i x_i^2\right) = t_0(\boldsymbol{x}) \prod_{i=1}^{N} \tilde{t}_i(x_i) \tag{41}$$

The i-th title distribution is as:

$$\hat{p}_i(\boldsymbol{x}) \sim \frac{q(\boldsymbol{x})}{\tilde{t}_i(x_i)} t_i(x_i). \tag{42}$$

The first order correction is to estimate by the following approximation:

$$p(\boldsymbol{x}) \sim \sum_i \hat{p}_i(\boldsymbol{x}) - (N-1) q(\boldsymbol{x}). \tag{43}$$

We need to compute the 1st moment of above for approximation, since that is what we need for estimate $\boldsymbol{x}$. Then the problem is boiled down to the moment computation:

$$\begin{aligned} \hat{\mu}_{1EP} &= \int_{x_i} \left[\sum_{i=1}^{N} \hat{p}_i(x_i) - (N-1) q(x_i) \right] x_i dx_i \\ &= \left[\sum_{n=1}^{N} \int_{x_i} \hat{p}_i(x_i) dx_i \right] - (N-1) \int_{x_i} q(x_i) x_i dx_i, \end{aligned} \tag{44}$$

where $\hat{p}_i(x_i)$ is the ith marginal of $\hat{p}_i(\boldsymbol{x})$, and $q(x_i)$ is the ith marginal of $q(\boldsymbol{x})$. Let us use the notation as:

$$\int \hat{p}_i(x_i) x_i dx_i = \mu_{\hat{p}_i(x_i)}, \tag{45}$$

$$\int \hat{p}_i(x_j) x_j dx_j = \mu_j, \tag{46}$$

$$\tag{47}$$

where $i \neq j$. Due to the fact that q is Gaussian, $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $q(x_i) = \mathcal{N}(x_i; \mu_i, \Sigma_{ii})$, the Equation (44) becomes

$$\hat{\mu}_{1EP} = \mu_{\hat{p}_i(x_i)} + (N-1)\mu_i - (N-1)\mu_i = \mu_{\hat{p}_i(x_i)}. \tag{48}$$

Then the detection via first order corrected EP is

$$\hat{x}_i = \underset{x_i}{\operatorname{argmin}} \, |x_i - \mu_{\hat{p}_i(x_i)}|. \tag{49}$$

5 Stochastic EP

In this section, we discuss an another variant of EP, named stochastic EP, which is proposed in [7]. The algorithm comes with the intuition of updating an approximate distribution in a stochastic way.

In stochastic EP, all approximate factors of $q(\boldsymbol{x})$ share the same parameter γ, Λ. In this case, the approximate distribution $q(\boldsymbol{x})$ is different from the previous sections. Here q has the form of:

$$q(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2 \boldsymbol{I}) \prod_{i=1}^{N} e^{\gamma x_i - \frac{1}{2}\Lambda x_i^2} \sim \mathcal{N}(\boldsymbol{x} : \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{50}$$

where

$$\boldsymbol{\Sigma} = (\sigma_w^{-2} \boldsymbol{H}^T \boldsymbol{H} + \Sigma \boldsymbol{I})^{-1}, \tag{51}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}(\sigma_w^{-2} \boldsymbol{H}^T \boldsymbol{y} + \gamma \mathbf{1}), \tag{52}$$

in which $\boldsymbol{I}$ is unitary matrix and $\mathbf{1}$ is a vectors with all elements of ones. This setting of models has the following pros and cons:

Pros: Model complexity does not scale with the number of fixed points/length of $\boldsymbol{x}$. In another word, large size of $\boldsymbol{H}$ does not necessarily bring large number of parameters to estimate.

Cons: Due to the simplicity of model setting, it may not bring accurate enough detection.

The algorithm steps are the same as EP but the update functions differ:

- Update the marginal distribution of the cavity distribution at iteration t:

$$q^{(l)\setminus i}(u_i) = \frac{q^{(l)(u_i)}}{\exp\left(\gamma^{(l)} u_i - \frac{1}{2}\Lambda^{(l)} u_i^2\right)}, \tag{53}$$

  which $q^{(l)\setminus i}(u_i) = \mathcal{N}\left(u_i : t_i^{(l)}, h_i^{2(l)}\right)$, with

$$h_i^{2(l)} = \frac{\sigma_i^{2(l)}}{1 - \sigma_i^{2(l)} \Lambda^{(l)}}, \tag{54}$$

$$t_i^{(l)} = h_i^{2(l)} \left(\frac{\mu_i^{(l)}}{\sigma_i^{2(l)}} - \gamma^{(l)} \right). \tag{55}$$

- Compute the mean $\mu_{p_i}^2$ and variance $\sigma_{p_i}^2$ in the same way as in Section 2.
- Update the statistics parameter (γ, Λ) of q, in the same way as in Section 2.

Note above only use one data sample to update per iteration. Since one data sample contain only limited information, it is suggested to do soft update with:

$$\tilde{t}_i^{(l+1)}(x_i) \leftarrow [\tilde{t}_i^{(l)}(x_i)]^{1-\varepsilon}[\tilde{t}_i^{(l+1)}(x_i)]^{\varepsilon}. \tag{56}$$

Then the final update step in Stochastic EP becomes:

$$\Lambda^{(l+1)} = \varepsilon\left[\frac{1}{\sigma_{p_i}^{2(l)}} - \frac{1}{h_i^{2(l)}}\right] + (1-\varepsilon)\Lambda^{(l)} \tag{57}$$

$$\gamma^{(l+1)} = \varepsilon\left[\frac{\mu_{p_i}^{(t)}}{\sigma_{p_i}^{2(t)}} - \frac{t_i^{(t)}}{h_i^{2(t)}}\right] + (1-\varepsilon)\Lambda^{(l)}. \tag{58}$$

Here the parameter ε can be chosen to depend on the information that each iteration of stochastic EP uses from dataset. Say $\varepsilon = 1/N$ if each iteration uses one data sample to update, or $\varepsilon =$ batch size$/N$ in batch fashion.

6 Expectation Consistency (EC)

In this section, we present the expectation consistency (EC) algorithm. EC is originally proposed by Opper in [12], where the approximation is obtained by maintaining partition/normalization function. The application of EC for MIMO is tested in [3].

The original distribution is the same as previous sections for EP and its variants, with explicit partition Z:

$$p(\boldsymbol{x}) = \frac{1}{Z}\mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2\boldsymbol{I})\frac{1}{N}\prod_{i=1}^{N}\delta_{x_i\in\mathcal{A}}. \tag{59}$$

In EC, two approximations are maintained to $p(\boldsymbol{x})$ equivalently:

$$f_q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2\boldsymbol{I}), \tag{60}$$

$$f_r(\boldsymbol{x}) = \frac{1}{N}\prod_{i=1}^{N}\delta_{x_i\in\mathcal{A}}. \tag{61}$$

The three distributions are required to maintained equivalent sufficient statistics:

$$\varphi(\boldsymbol{x}) = \left[x_1, x_2, \cdots, x_N, -\frac{x_1^2}{2}, -\frac{x_2^2}{2}, \cdots, -\frac{x_N^2}{2}\right]^T. \tag{62}$$

Let us denote the parameters corresponding to the above statistics of distribution $q(\boldsymbol{x})$, $r(\boldsymbol{x})$, $s(\boldsymbol{x})$ are

$$\boldsymbol{\lambda}_q = [\gamma_{q,1}, \gamma_{q,2}, \cdots, \gamma_{q,N}, \Lambda_{q,1}, \Lambda_{q,2}, \cdots, \Lambda_{q,N}] = [\boldsymbol{\gamma}_q, \Lambda_q]^T, \tag{63}$$

$$\boldsymbol{\lambda}_r = [\gamma_{r,1}, \gamma_{r,2}, \cdots, \gamma_{r,N}, \Lambda_{r,1}, \Lambda_{r,2}, \cdots, \Lambda_{r,N}] = [\boldsymbol{\gamma}_r, \Lambda_r]^T, \tag{64}$$

$$\boldsymbol{\lambda}_s = [\gamma_{s,1}, \gamma_{s,2}, \cdots, \gamma_{s,N}, \Lambda_{s,1}, \Lambda_{s,2}, \cdots, \Lambda_{s,N}] = [\boldsymbol{\gamma}_s, \Lambda_s]^T. \tag{65}$$

The distributions corresponding to these parameters are:

$$q(\boldsymbol{x}) \sim f_q(\boldsymbol{x}) \exp(\boldsymbol{\lambda}_q^T \varphi(\boldsymbol{x})) = f_q(\boldsymbol{x}) \exp\left(\boldsymbol{\gamma}_q^T \boldsymbol{x} - \frac{\boldsymbol{x}^T \operatorname{diag}(\boldsymbol{\Lambda}_q)\boldsymbol{x}}{2}\right), \tag{66}$$

$$r(\boldsymbol{x}) \sim \exp\left(\boldsymbol{\gamma}_r^T \boldsymbol{x} - \frac{\boldsymbol{x}^T \operatorname{diag}(\boldsymbol{\Lambda}_r)\boldsymbol{x}}{2}\right) \prod_{i=1}^{N} \delta_{x_i \in \mathcal{A}}, \tag{67}$$

$$s(\boldsymbol{x}) \sim \exp\left(\boldsymbol{\gamma}_s^T \boldsymbol{x} - \frac{\boldsymbol{x}^T \operatorname{diag}(\boldsymbol{\Lambda}_s)\boldsymbol{x}}{2}\right). \tag{68}$$

The goal of EC is to achieve

$$\mathbb{E}_{q(\boldsymbol{x})}[x_i] = \mathbb{E}_{r(\boldsymbol{x})}[x_i] = \mathbb{E}_{s(\boldsymbol{x})}[x_i], \tag{69}$$

$$\mathbb{E}_{q(\boldsymbol{x})}[x_i^2] = \mathbb{E}_{r(\boldsymbol{x})}[x_i^2] = \mathbb{E}_{s(\boldsymbol{x})}[x_i^2]. \tag{70}$$

The steps of EC is as follows:

- Initialize $\boldsymbol{\gamma}_q$, $\boldsymbol{\Lambda}_q$
- Repeat the following iterations:
  - Given $\boldsymbol{\gamma}_q^{(l-1)}$, $\boldsymbol{\Lambda}_q^{(l-1)}$, compute $\mathbb{E}_{q(\boldsymbol{x})}[x_i]$, $\mathbb{E}_{q(\boldsymbol{x})}[x_i^2]$
  - Compute $\boldsymbol{\gamma}_s^{(l)}$, $\boldsymbol{\Lambda}_s^{(l)}$ by solving $\mathbb{E}_{s(\boldsymbol{x})}[x_i] = \mathbb{E}_{q(\boldsymbol{x})}[x_i]$ and $\mathbb{E}_{s(\boldsymbol{x})}[x_i^2] = \mathbb{E}_{q(\boldsymbol{x})}[x_i^2]$
  - Update $\boldsymbol{\gamma}_r^{(l)} = \boldsymbol{\gamma}_s^{(l)} - \boldsymbol{\gamma}_q^{(l)}$, $\boldsymbol{\Lambda}_r^{(l)} = \boldsymbol{\Lambda}_s^{(l)} - \boldsymbol{\Lambda}_q^{(l)}$
  - Given $\boldsymbol{\gamma}_r^{(l-1)}$, $\boldsymbol{\Lambda}_r^{(l-1)}$, compute $\mathbb{E}_{r(\boldsymbol{x})}[x_i]$, $\mathbb{E}_{r(\boldsymbol{x})}[x_i^2]$
  - Compute $\boldsymbol{\gamma}_s^{(l)}$, $\boldsymbol{\Lambda}_s^{(l)}$ by solving $\mathbb{E}_{s(\boldsymbol{x})}[x_i] = \mathbb{E}_{r(\boldsymbol{x})}[x_i]$ and $\mathbb{E}_{s(\boldsymbol{x})}[x_i^2] = \mathbb{E}_{r(\boldsymbol{x})}[x_i^2]$
  - Update

$$\boldsymbol{\gamma}_q^{(l)} = \beta\left(\boldsymbol{\gamma}_s^{(l)} - \boldsymbol{\gamma}_r^{(l)}\right) + (1-\beta)\boldsymbol{\gamma}_q^{(l-1)}, \tag{71}$$

$$\boldsymbol{\Lambda}_q^{(l)} = \beta\left(\boldsymbol{\Lambda}_s^{(l)} - \boldsymbol{\Lambda}_r^{(l)}\right) + (1-\beta)\boldsymbol{\Lambda}_q^{(l-1)} \tag{72}$$

7 Some Techniques to Gain Detection Performance

In this section, we discuss some techniques that may be used to help the detection performance.

7.1 Pre-processing by QR decomposition

In previous sections, we discuss the EP algorithm and variants of EP based algorithms, which actually do belief propagation in order to do estimation. It is known that (loopy) belief propagation and its variants have better inference capability on tree-structured graph. The more loopy a graph is, the more degeneration the performance of belief propagation would encounter. This inspires us to do some pre-processing for the linear model in (1), by QR decomposition. According to QR decomposition, any real square matrix can be decomposed into multiplication of an orthogonal matrix and an triangular matrix. Then let us do the QR decomposition of $\boldsymbol{H}$ in (1)

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w} = \boldsymbol{Q}\boldsymbol{R}\boldsymbol{x} + \boldsymbol{w}. \tag{73}$$

Since $\boldsymbol{Q}$ is an orthogonal matrix, $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$. Then

$$\tilde{\boldsymbol{y}} = \boldsymbol{R}\boldsymbol{x} + \tilde{\boldsymbol{w}}, \tag{74}$$

where

$$\tilde{\boldsymbol{y}} = \boldsymbol{Q}^T\boldsymbol{y}, \tag{75}$$

$$\tilde{\boldsymbol{w}} = \boldsymbol{Q}^T\boldsymbol{w}. \tag{76}$$

With this pre-processing trick, the graph representing problem (74) has less loops that that representing problem (1), due to the fact that $\boldsymbol{R}$ is sparser than $\boldsymbol{H}$.

7.2 Ensemble/Boosting

Ensemble method is a meta learning algorithm in statistics and machine learning [6, 8, 14]. The basic idea is to use multiple learning algorithms to obtain better predictive performance than that could be obtained from any of constituent learning algorithm alone. Boosting is actually a type of assembling methods, which incrementally builds an ensemble by training each new model to emphasize the gap missed by previous models.

So far, since we have introduced the a verity of estimation methods for the problem defined in Section 1. Therefore, we can use the ensemble framework to obtain ensembled estimation model if any standalone model discussed in previous sections does not has performance fulfilled requirements. We omit the detailed discussion on ensemble/boosting here, and would give specific steps when it is used in our experiments.

8 α belief propagation

8.1 Preliminary

In this section, we provide the preliminaries that are needed in this paper. We introduce the α-divergence and a graphical model that we are going to use to explain α-BP.

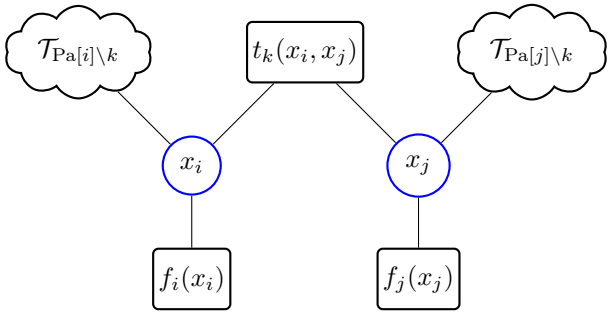


Figure 1: Factor graph illustration of Equation 81.

8.2 Divergence Measures

We are going to minimize α-divergence between p and q, which is defined as follows according to [15][11]:

$$\mathcal{D}_\alpha(p\|q) = \frac{\int_{\boldsymbol{x}} \alpha p(\boldsymbol{x}) + (1-\alpha)q(\boldsymbol{x}) - p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}}{\alpha(1-\alpha)}, \tag{77}$$

where α is the parameter of α-divergence, distribution p and q are unnormalized, i.e. $\int_{\boldsymbol{x}} p(\boldsymbol{x})d\boldsymbol{x} \neq 1$, $\int_{\boldsymbol{x}} q(\boldsymbol{x})d\boldsymbol{x} \neq 1$.

The classic KL divergence is defined as

$$KL(p\|q) = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x} + \int q(\boldsymbol{x}) - p(\boldsymbol{x}) d\boldsymbol{x} \tag{78}$$

where the $\int q(\boldsymbol{x}) - p(\boldsymbol{x})d\boldsymbol{x}$ is a correction factor to accommodate unnormalized p and q. The KL divergence is a special case of α-divergence, since $\lim_{\alpha\to 1} \mathcal{D}_\alpha(p\|q) = KL(p\|q)$ and $\lim_{\alpha\to 0} \mathcal{D}_\alpha(p\|q) = KL(q\|p)$, by applying L'Hôpital's rule to Equation 77.

Both α-divergence and KL divergence are equal to zero if $p = q$, and they are non-negative (therefore satisfy the basic property of error measure). Denote KL-projection by

$$\mathrm{proj}[p] = \operatorname*{argmin}_{q\in\mathcal{F}} KL(p\|q), \tag{79}$$

where $\mathcal{F}$ is the distribution family of q.

According to the stationary point equivalence Theorem in [11], $\mathrm{proj}[p^\alpha q^{1-\alpha}]$ and $\mathcal{D}_\alpha(p\|q)$ have same stationary points. A heuristic scheme to find q minimizing $\mathcal{D}_\alpha(p\|q)$ is to find its stationary point by a fixed-point iteration:

$$q(\boldsymbol{x})^{\mathrm{new}} = \mathrm{proj}[p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha}]. \tag{80}$$

8.3 A Graphic Model

We introduce a pairwise Markov random field (MRF) $p(\boldsymbol{x})$ to explain our algorithm. Variable $\boldsymbol{x} \in \mathcal{A}^N$, where $\mathcal{A}$ is a discrete finite set or subset of $\mathbb{R}$ and

N is a positive integer. We factorize the distribution $p(\boldsymbol{x})$ as

$$p(\boldsymbol{x}) \propto \prod_{i=1}^{N} f_i(x_i) \prod_{k \in \mathcal{K}} t_k(x_i, x_j), \tag{81}$$

where f_i is the *singleton factor*, t_k is *pairwise factor*, $\mathcal{K}$ is the index set of all pairwise factors, and $\propto$ denotes the fact that the only difference between two sides of $\propto$ is a constant factor.

The factor graph representing Equation 81 is shown in Figure 1. In the figure, Pa$[i]$ is the index set of pairwise factors connecting to variable node x_i, i.e. Pa$[i]$ is subset of $\mathcal{K}$, $\setminus$ denotes exclusion. $\mathcal{T}_{\text{Pa}[i]\setminus k}$ is the product of all pairwise factors connecting to x_i except for t_k:

$$\mathcal{T}_{\text{Pa}[i]\setminus k} = \prod_{n \in \text{Pa}[i]\setminus k} t_n. \tag{82}$$

8.4 α-BP as Fully-Factorized Approximation

In this section, we will show why α-BP as a message-passing algorithm can be used as a fully-factorized approximation to the original distribution $p(\boldsymbol{x})$.

Fully Factorized Surrogate:

Now we formulate a surrogate distribution as

$$q(\boldsymbol{x}) \propto \prod_{i=1}^{N} \tilde{f}_i(x_i) \prod_{k \in \mathcal{K}} \tilde{t}_k(x_i, x_j), \boldsymbol{x} \in \mathcal{A}^N \tag{83}$$

to approximate $p(\boldsymbol{x})$. The surrogate distribution would be used to estimate inference problems of $p(\boldsymbol{x})$. We further assume that $q(\boldsymbol{x})$ can be fully factorized, which means that $\tilde{t}_k(x_i, x_j)$ can be factorized as two independent functions of x_i, x_j respectively. We denote this factorization as

$$\tilde{t}_k(x_i, x_j) = m_{k\to i}(x_i) m_{k\to j}(x_j). \tag{84}$$

We use the notation $m_{k\to i}(x_i)$ to denote the factor as a function of x_i due to the intuitive fact that $m_{k\to i}$ is also the message from the factor $t_k(x_i, x_j)$ to variable node x_i. Similarly we have factor $m_{k\to j}(x_j)$. Then the marginal can be formulated straightforwardly as

$$q_i(x_i) \propto \tilde{f}_i(x_i) \prod_{k \in \text{Pa}[i]} m_{k\to i}(x_i). \tag{85}$$

Local α-Divergence Minimization:

Now, we are going to use the heuristic scheme as in Equation 80 to minimize the information loss by using tractable $q(\boldsymbol{x})$ to represent $p(\boldsymbol{x})$. The information loss is measured by α-divergence $\mathcal{D}_\alpha(p(\boldsymbol{x}) \| q(\boldsymbol{x}))$.

We do factor-wise refinement to update the factors of $q(\boldsymbol{x})$ such that $q(\boldsymbol{x})$ approaches $p(\boldsymbol{x})$ asymptotically similar to [11, 9]. Without losing generality, we

begin to refine factor $\tilde{t}_k(x_i, x_j)$. Define $q^{\setminus k}(\boldsymbol{x})$ as all other factors except for $\tilde{t}_k(x_i, x_j)$

$$q^{\setminus k}(\boldsymbol{x}) = q(\boldsymbol{x})/\tilde{t}_k(x_i, x_j) \propto \prod_i \tilde{f}_i(x_i) \prod_{n \in \mathcal{K} \setminus k} \tilde{t}_n(x_i, x_j). \tag{86}$$

Similarly, we have $p^{\setminus k}(\boldsymbol{x})$ as all other factors except for $t_k(x_i, x_j)$. Assume that we already have had $q^{\setminus k}(\boldsymbol{x})$ as a good approximation of $p^{\setminus k}(\boldsymbol{x})$, i.e. $q^{\setminus k}(\boldsymbol{x}) \simeq p^{\setminus k}(\boldsymbol{x})$, it is $\tilde{t}_k(x_i, x_j)$ that remains to be refined. Then the problem $\underset{\tilde{t}_k^{\text{new}}}{\text{argmin}}\, \mathcal{D}_\alpha \left(p^{\setminus k} t_k \| q^{\setminus k} \tilde{t}_k^{\text{new}}\right)$ becomes

$$\underset{\tilde{t}_k^{\text{new}}(x_i, x_j)}{\text{argmin}}\, \mathcal{D}_\alpha \left(q^{\setminus k}(\boldsymbol{x}) t_k(x_i, x_j) \| q^{\setminus k}(\boldsymbol{x}) \tilde{t}_k^{\text{new}}(x_i, x_j)\right), \tag{87}$$

which searches for new factor $\tilde{t}_k^{\text{new}}$ such the above divergence is minimized. Using Equation 80, the above problem is equivalent to

$$\begin{aligned} & q^{\setminus k}(\boldsymbol{x}) \tilde{t}_k^{\text{new}}(x_i, x_j) \\ & \propto \text{proj} \left[\left(q^{\setminus k}(\boldsymbol{x}) t_k(x_i, x_j)\right)^\alpha \left(q^{\setminus k}(\boldsymbol{x}) \tilde{t}_k(x_i, x_j)\right)^{1-\alpha} \right] \\ & \propto \text{proj} \left[q^{\setminus k}(\boldsymbol{x}) t_k(x_i, x_j)^\alpha \tilde{t}_k(x_i, x_j)^{1-\alpha} \right]. \end{aligned} \tag{88}$$

Let us refine one message per time in factor $\tilde{t}_k$. Without lose of generality, we update $m_{k \to i}$ and denote

$$\tilde{t}_k^{\text{new}}(x_i, x_j) = m_{k \to i}^{\text{new}}(x_i) m_{k \to j}(x_j). \tag{89}$$

Since KL-projection to a fully factorized distribution reduces to matching the marginals, Equation 88 is reduced to

$$\sum_{\boldsymbol{x} \setminus x_i} q^{\setminus k}(\boldsymbol{x}) \tilde{t}_k^{\text{new}}(x_i, x_j) \propto \sum_{\boldsymbol{x} \setminus x_i} q^{\setminus k}(\boldsymbol{x}) t_k(x_i, x_j)^\alpha \tilde{t}_k(x_i, x_j)^{1-\alpha}. \tag{90}$$

We use summation here. But it should be replaced by integral if $\mathcal{A}$ is a continuous set. Solving Equation 90 gives the message passing rule as

$$\begin{aligned} m_{k \to i}^{\text{new}}(x_i) \propto & \left[\sum_{x_j} t_k(x_i, x_j)^\alpha m_{k \to j}(x_j)^{1-\alpha} m_{j \to k}(x_j) \right] \\ & \cdot m_{k \to i}(x_i)^{1-\alpha}, \end{aligned} \tag{91}$$

where

$$m_{j \to k}(x_j) = \tilde{f}_j(x_j) \prod_{n \in \text{Pa}[j] \setminus k} m_{n \to j}(x_j). \tag{92}$$

Similarly, the message from t_k to x_j, $m_{k \to j}(x_j)$, can be updated in similar way.

As for the singleton factor $\tilde{f}_i(x_i)$, we can do the refinement procedure on $\tilde{f}_i(x_i)$ in the same way as we have done for $\tilde{t}_k(x_i, x_j)$. This gives us the update rule of $\tilde{f}_i(x_i)$ as

$$\tilde{f}_i^{\text{new}}(x_i) \propto f_i(x_i)^{\alpha} \tilde{f}_i(x_i)^{1-\alpha}, \tag{93}$$

which is the belief from factor $f_i(x_i)$ to variable x_i. Note, if we initialize $\tilde{f}_i(x_i) = f_i(x_i)$, then it remains the same in all iterations.

8.5 Remarks on α-BP

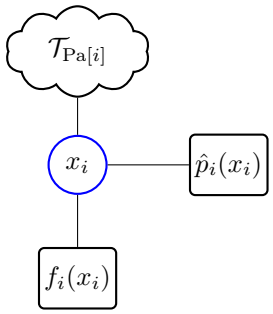


Figure 2: Factor graph illustration with prior factor.

As discussed in Section 8.1, $KL(p\|q)$ is the special case of $\mathcal{D}_\alpha(p\|q)$ when $\alpha \to 1$. When applying $\alpha = 1$ to Equation 91, it gives

$$m_{k\to i}^{\text{new}}(x_i) \propto \sum_{x_j} t_k(x_i, x_j) m_{j\to k}(x_j), \tag{94}$$

which is exactly the messages of BP in Chapter 8 of [1]. From this point of view, α-BP generalizes BP.

Inspired by [5] and assembling methods [6], we can add an extra singleton factor to each x_i as prior information that is obtained from other (usually weak) methods. This factor stands for our belief from exterior estimation. Then run our α-BP. Denote the prior by $\hat{p}_i(x_i)$ for variable node x_i, then the factor graph including this prior belief can be represented as in Figure 2.

We summarize the method into the pseudo-code in Algorithm 1. Though we explain the method with a binary MRF, it is straightforward to replace the factor t_k by a factor involving more than two variables and applies α-BP to general factor graphs.

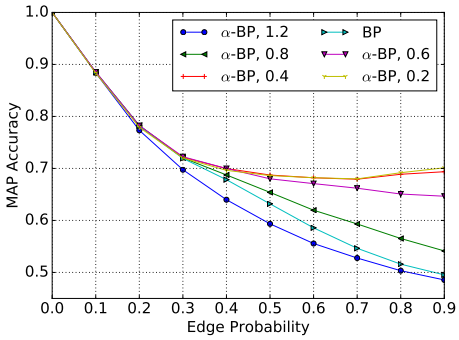
8.6 Experimental Results of α-BP

In this section, we report numerical results on the α-BP. It is well known that performance of BP and its variants deteriorate significantly when loops appear in factor graph. We would like to see if α-BP could relief the deterioration

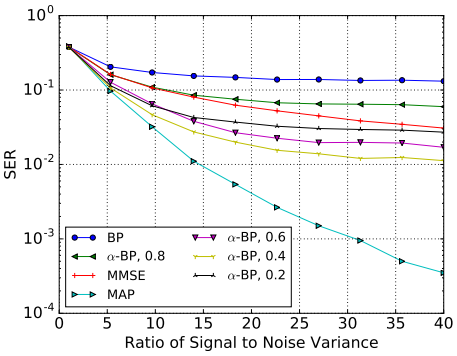
Algorithm 1 Algorithm of α-BP

Input: Factor graph of $p(\boldsymbol{x})$

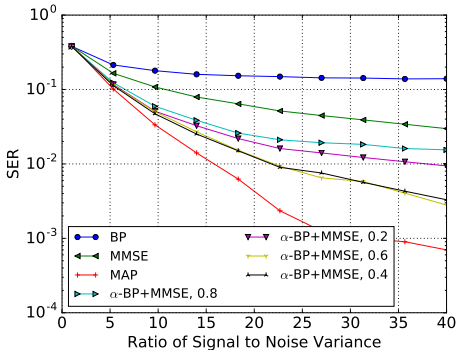
Initialize $q(\boldsymbol{x})$
if Prior belief on x_i available **then**
  Add prior factor as Figure 2
end if
while not converge **do**
  for each edge of factor graph **do**
    Message update by Equation 91 or Equation 93
  end for
end while
return $q(\boldsymbol{x})$



(a) Mismatch between MAP and α-BP on binary MRF



(b) MIMO detection: α-BP without prior



(c) MIMO detection: α-BP with prior

Figure 3: Numerical Results of α-BP on: (a) binary MRF, (b) and (c) MIMO detection.

brought by loops in inference. Thus we firstly test the performance of α-BP for MAP inference in a MRF with $\mathcal{A} = \{-1, 1\}$ (Ising model), where we adjust how loopy its corresponding factor graph is.

In addition, we apply the α-BP to a MIMO detection problem, to explore its performance in comparison with (loopy) BP and minimum mean square error (MMSE). At the end, the prior factor trick is used according to discussion in Subsection 8.5. This turns out to be MAP inference problem as well.

For the MAP inference, the most probable estimation by α-BP, $\hat{\boldsymbol{x}} = [\hat{x}_1, \cdots, \hat{x}_N]$, is obtained by

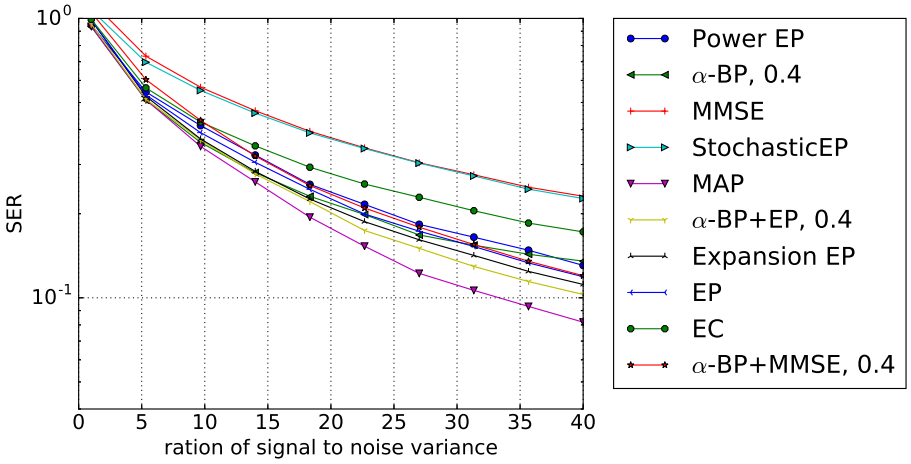
$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}}\ \tilde{f}_i(x_i) \prod_{k \in \mathrm{Pa}[i]} m_{k \to i}(x_i), x_i \in \mathcal{A}. \tag{95}$$

With $t_k(x_i, x_j) = e^{-2J_{i,j}x_i x_j}$ and $f_i(x_i) = e^{-J_{i,i}-b_i x_i}$, Equation 81 can be reformulated as

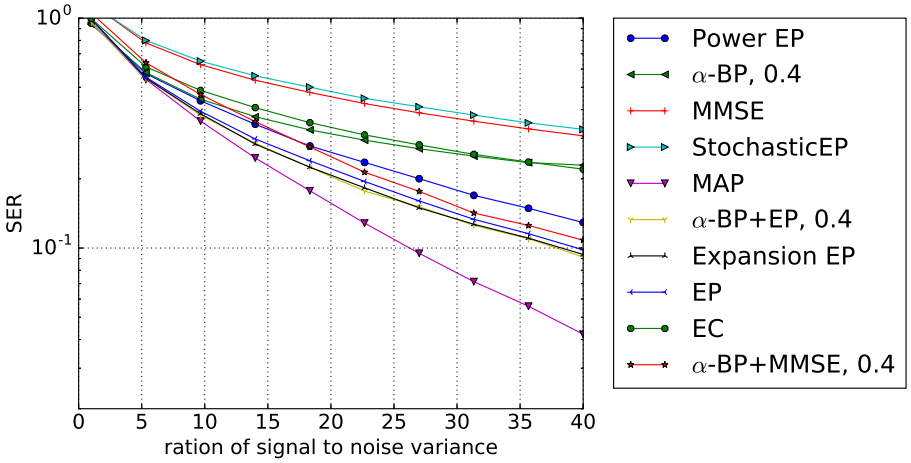
$$p(\boldsymbol{x}) \propto \exp\{-\boldsymbol{x}^T \boldsymbol{J} \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}\}, \boldsymbol{x} \in \mathcal{A}^N, \tag{96}$$

where $\boldsymbol{x}^T$ is transform of $\boldsymbol{x}$, $J_{i,j}$ is element of symmetrix matrix $\boldsymbol{J}$ at i-th row and j-th column, $\boldsymbol{b} = [b_1, \cdots, b_N]^T$.

For this experiment, we set $\mathcal{A} = \{-1, 1\}$ and $N = 9$. Bias $\boldsymbol{b}$ is sampled from Gaussian, $b_i \sim \mathcal{N}(0, (1/4)^2)$. Since $\boldsymbol{J}$ decides the loopy level of its corresponding factor graph, we use the Erdos-Rényi model [4] to construct its connectivity. Namely, an element of $\boldsymbol{J}$ is set as non-zero, $J_{i,j} = J_{j,i} \sim \mathcal{N}(0, 1)$, with an *Edge Probability*. Otherwise, $J_{i,j} = J_{j,i} = 0$, which means this is no connection between variable node x_i and x_j. For each test value of Edge Probability, 5000 binary MRF models are generated randomly and mismatch between $\operatorname{argmax}_{\boldsymbol{x}} p(\boldsymbol{x})$ and $\left\{\boldsymbol{x} \middle| \operatorname{argmax}_{x_i} q_i(x_i)\right\}$ is computed in each realization. The results are shown in Figure 3a. As the Edge Probability increases, graphs become loopier and α-BP (also BP) has more mismatch with MAP inference. In general, α-BP with $\alpha > 1$ underperforms BP, and α-BP with $\alpha < 1$ outperforms BP. For $\alpha = 0.2, 0.4, 0.6$, α-BP stops deteriorating for Edge Probability increasing over 0.35, while BP continues giving even worse approximations.



(a) Scenario: input size 4, ouput size 4.



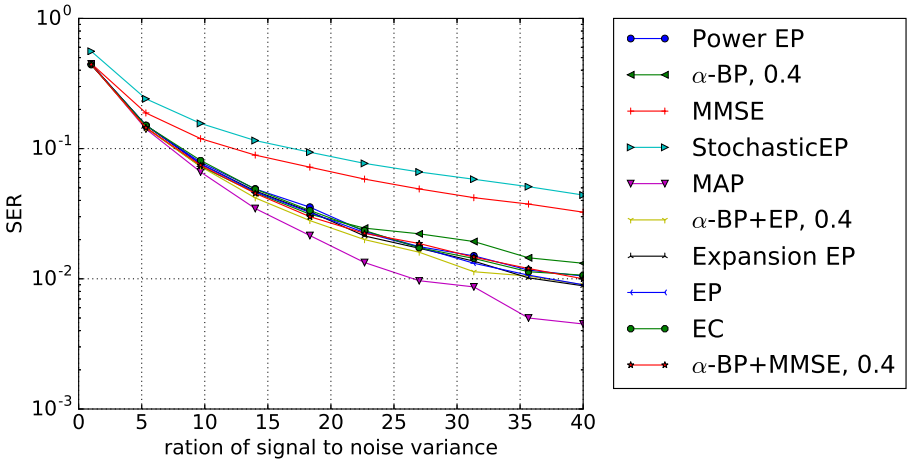
(b) Scenario: input size 8, ouput size 8.

Figure 4: Numerical results of discussed algorithms, constellation size 4.

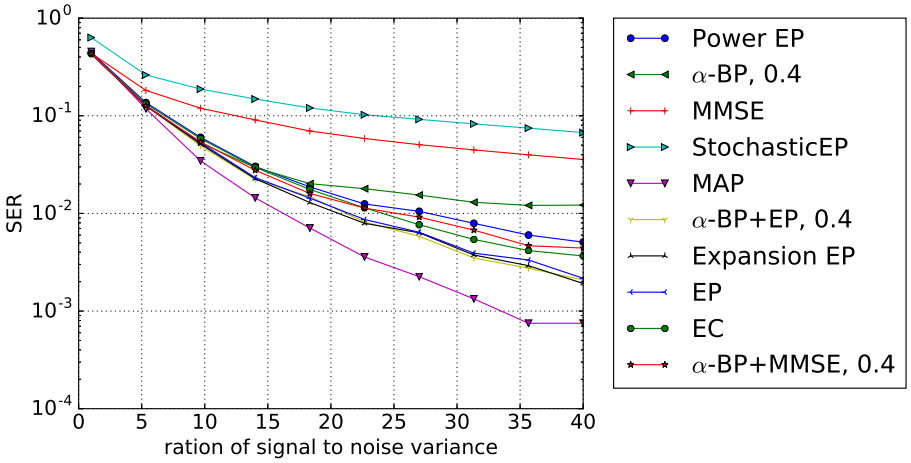
9 Numerical Results

In this section, we made a comparison between the algorithms discussed in this report. The software written for this set of experiments is implemented in Python. The algorithms implemented includes MMSE, expectation propagation, stochastic expectation propagation, power expectation propagation, expansion expectation propagation (correction of expectation propagation), and some improvement tricks. The maximum a posterior is used shown numerically as a reference bound for all algorithms.

The legends explanation for corresponding algorithm are shown as in Table 1. The quantity comes with α-BP is the α value used in experiment for the algorithm of α-BP and its ensembles. In Table 1, MAP stands for maximum a



(a) Scenario: input size 4, ouput size 4.



(b) Scenario: input size 4, ouput size 4.

Figure 5: Numerical results of discussed algorithms, constellation size 2.

posterior, which is the optimal solution for problem (1), i.e.

$$\underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{x}|\boldsymbol{y}) = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \mathcal{N}(\boldsymbol{y} : \boldsymbol{H}\boldsymbol{x}, \sigma_w^2 \boldsymbol{I}) \delta_{\boldsymbol{x}\in\mathcal{A}}. \tag{97}$$

Table 1: Legend Explanation for Experiment Section

MMSE	Minimum Mean Square Error
EP	Expectation Propagation (Section 2)
Power EP	Power Expectation Propagation (Section 3)
Expansion EP	Expansion Expectation Propagation (The First-order Expansion in Section 4)
Stochastic EP	Stochastic Expectation Propagation (Section 5)
EC	Expectation Consistency (Section 6)
α-BP	α Belief Propagation (Section 8)
α-BP+MMSE	α-BP assembled with MMSE as prior (Section 7.2,8.5)
α-BP+EP	α-BP assembled with EP as prior (Similar to the α-BP+MMSE)
MAP	Maximum a Posterior

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[2] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz. Expectation propagation detection for high-order high-dimensional mimo systems. *IEEE Transactions on Communications*, 62(8):2840–2849, Aug 2014.

[3] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz. Probabilistic mimo symbol detection with expectation consistency approximate inference. *IEEE Transactions on Vehicular Technology*, 67(4):3481–3494, April 2018.

[4] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[5] J. Goldberger and A. Leshem. Pseudo prior belief propagation for densely connected discrete graphs. In *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, pages 1–5, Jan 2010.

[6] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[7] Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2323–2331. Curran Associates, Inc., 2015.

[8] Richard Maclin and David W. Opitz. Popular ensemble methods: An empirical study. *CoRR*, abs/1106.0257, 2011.

[9] Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[10] Tom Minka. Power ep. Technical Report MSR-TR-2004-149, January 2004.

[11] Tom Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, January 2005.

[12] M. Opper and O. Winther. Expectation consistent approximate inference, 2005.

[13] Manfred Opper, Ulrich Paquet, and Ole Winther. Improving on expectation propagation. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1241–1248. Curran Associates, Inc., 2009.

[14] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, Feb 2010.

[15] Huaiyu Zhu and Richard Rohwer. Information geometric measurements of generalisation. Technical report, 1995.