# DENSITY EVOLUTION FOR EXPECTATION PROPAGATION

*John MacLaren Walsh, Ph. D.*

Drexel University
Department of Electrical & Computer Engineering
Philadelphia, PA 19104

## ABSTRACT

Expectation propagation (EP) [1, 2, 3, 4] is a theoretical extension of the belief propagation family of message passing algorithms [5, 6] for statistical inference which allows for efficient handling of models with continuous random variables as well as second or higher order correlation via the use of standard exponential families of probability measures [7, 8, 9]. Here we provide theoretically rigorous justifications for the use of density evolution [10, 11] to analyze the convergence and performance behavior of the family of algorithms in the large system regime by extending and expanding on the corresponding results for belief propagation decoding and turbo decoding.

***Index Terms***— expectation propagation, Bayes procedures, distributed iterative decoding and estimation, belief propagation

## 1. INTRODUCTION AND NOTATION

In this paper we will mathematically study the performance and convergence behavior of a family of iterative algorithms for Bayesian statistical inference known as expectation propagation (henceforth EP) [1, 2]. These algorithms extend belief propagation [5, 6] to efficiently handle continuous random variables and to allow for second and higher order statistical dependence among the random variables. Due to the rich family of algorithms which can be considered as special cases of belief propagation and thus EP, e.g. the turbo decoder/equalizer [13], the LDPC decoder [14], the Kalman filter, and the forward backward algorithm, a wealth of literature exists studying the performance and convergence properties of instances of EP.

One such body of convergence and performance analysis work which has had incredible theoretical impact is called density evolution [15, 11, 10]. Density evolution studies the performance and convergence of the turbo and LDPC decoders in the limit that the block length grows arbitrarily large. Of particular interest in this paper is the extension of the ideas of density evolution [15, 11, 10] from the decoding of turbo and LDPC codes to EP. With this in mind, we provide a review of EP in section 2, followed by a list of conditions under which density evolution can be applied to EP in section 3. The paper concludes with a list of possible practical applications of this theoretical work.

## 2. EXPECTATION PROPAGATION

EP is a distributed iterative method for statistical inference, which approximates the a posteriori distribution for some high-dimensional vector of parameters $\boldsymbol{\theta}$ given observations $\mathbf{r}$. The parameters $\boldsymbol{\theta}$ are known to be members of the set $\mathcal{Q}$ which is the Cartesian product of the sets $\{\mathcal{Q}_\mathtt{i}\}$

$$\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2 \times \cdots \times \mathcal{Q}_\mathtt{K}$$

so that $\boldsymbol{\theta}$ is a vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_\mathtt{K}]$, with $\theta_\mathtt{i} \in \mathcal{Q}_\mathtt{i}$ for all $\mathtt{i} \in \{1, \ldots, \mathtt{K}\}$. We will assume that $\mathcal{Q}$ has been endowed with a measure $\mathsf{d}\boldsymbol{\theta}$ that is a product of measures $\mathsf{d}\theta_\mathtt{i}$ on the $\mathcal{Q}_\mathtt{i}$s. Typically, $\mathsf{d}\theta_\mathtt{i}$ will be a counting measure if $\mathcal{Q}_\mathtt{i}$ is countable, and will be Lebesgue measure otherwise.

In statistical inference problems to which EP may be applied, there is a multiplicative factoring of the Radon Nikodym derivative of the joint probability distribution for $\mathbf{r}$ and $\boldsymbol{\theta}$ with respect to $\mathsf{d}\boldsymbol{\theta}$,

$$\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r}, \boldsymbol{\theta}) := \prod_{\mathtt{a}=1}^{\mathtt{A}} \mathsf{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_\mathtt{a}), \quad \boldsymbol{\theta}_\mathtt{a} \subset \boldsymbol{\theta}$$

where $\boldsymbol{\theta}_\mathtt{a} \subset \boldsymbol{\theta}$ indicates that $\boldsymbol{\theta}_\mathtt{a}$ is the (smaller dimensional) vector created by removing some of the elements of $\boldsymbol{\theta}$. EP exploits this multiplicative factoring in order to approximate $\mathsf{p}_{\boldsymbol{\theta}|\mathbf{r}}(\boldsymbol{\theta}|\mathbf{r})$ with the product of minimal standard exponential family densities $\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r})}(\boldsymbol{\theta}_\mathtt{a})$

$$\mathsf{p}_{\boldsymbol{\theta}|\mathbf{r}}(\boldsymbol{\theta}|\mathbf{r}) \approx \mathsf{c}(\boldsymbol{\lambda}^{(\mathtt{k})}) \prod_{\mathtt{a}=1}^{\mathtt{A}} \mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r})}(\boldsymbol{\theta}_\mathtt{a}) \tag{1}$$

Here $\boldsymbol{\lambda}^{(\mathtt{k})} := \left[\boldsymbol{\lambda}_1^{(\mathtt{k})}, \ldots, \boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}, \ldots, \boldsymbol{\lambda}_\mathtt{A}^{(\mathtt{k})}\right]$ is a vector of adjustable parameters which EP iteratively refines over time $\mathtt{k}$ to improve the approximation in (1) for a particular observed value $\mathbf{r} = \mathbf{r}$. Also, $\mathsf{c}$ is a normalization constant defined by

$$\mathsf{c}(\boldsymbol{\lambda}^{(\mathtt{k})}) := \left( \int_{\mathcal{Q}} \prod_{\mathtt{a}=1}^{\mathtt{A}} \exp\left(\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}^{(\mathtt{k})} \cdot \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\right) \mathsf{d}\boldsymbol{\theta} \right)^{-1}$$

and the gs are standard exponential family probability densities [7, 8]

$$\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r})}(\boldsymbol{\theta}_\mathtt{a}) := \exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \cdot \boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r}) - \psi_{\mathbf{t}_\mathtt{a}}(\boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r}))\right)$$

with the log partition function defined by

$$\psi_{\mathbf{t}_\mathtt{a}}(\boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r})) := \log\left( \int \exp(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \cdot \boldsymbol{\lambda}_\mathtt{a}^{(\mathtt{k})}(\mathbf{r})) \mathsf{d}\boldsymbol{\theta}_\mathtt{a} \right)$$

EP attempts to choose the best approximation in (1) by iteratively refining the $\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}$s by solving the equations

$$\frac{\int \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \exp(\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}} \cdot \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})) \prod_{\mathtt{c}\neq\mathtt{a}} \exp\left(\mathbf{t}_\mathtt{c}(\boldsymbol{\theta}_\mathtt{c}) \cdot \boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}\right) \mathsf{d}\boldsymbol{\theta}}{\int \exp(\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}} \cdot \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})) \prod_{\mathtt{c}\neq\mathtt{a}} \exp\left(\mathbf{t}_\mathtt{c}(\boldsymbol{\theta}_\mathtt{c}) \cdot \boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}\right) \mathsf{d}\boldsymbol{\theta}} = \frac{\int \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \mathsf{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \prod_{\mathtt{c}\neq\mathtt{a}} \exp\left(\mathbf{t}_\mathtt{c}(\boldsymbol{\theta}_\mathtt{c}) \cdot \boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}\right) \mathsf{d}\boldsymbol{\theta}}{\int \mathsf{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \prod_{\mathtt{c}\neq\mathtt{a}} \exp\left(\mathbf{t}_\mathtt{c}(\boldsymbol{\theta}_\mathtt{c}) \cdot \boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}\right) \mathsf{d}\boldsymbol{\theta}} \tag{2}$$

for $\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}$ in terms of $\{\boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}} \,|\, \mathtt{c} \neq \mathtt{a}\}$. The order in which these equations are solved is referred to as scheduling, and there are several

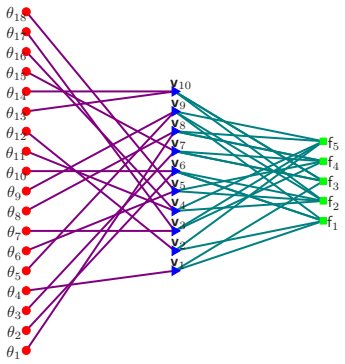

**Fig. 1**. A parameter basis factor graph.

possibilities. In **parallel scheduling**, (2) is solved to get $\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}^{(\mathtt{k}+1)}$ from $\boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}^{(\mathtt{k})}, \mathtt{c} \neq \mathtt{a}$ for each $\mathtt{a}$. In **serial scheduling**, (2) is solved to get $\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}^{(\mathtt{k}+1)}$ from $\boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}^{(\mathtt{k})}, \boldsymbol{\lambda}_{\mathtt{c}',\mathbf{r}}^{(\mathtt{k}+1)}, \mathtt{c} < \mathtt{a}, \mathtt{c}' > \mathtt{a}$ for each $\mathtt{a}$. Finally, in **random scheduling**, (2) is solved to update $\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}$ from the most recently updated values of $\boldsymbol{\lambda}_{\mathtt{c},\mathbf{r}}, \mathtt{c} \neq \mathtt{a}$.

In this article, we will focus on parallel scheduling since we wish to emphasize the distributed, parallelized, nature of EP. In fact, under some additional assumptions, solving (2) iteratively can be equated to a message passing algorithm on a statistics factor graph. Situations in which these assumptions are satisfied will constitute the cases of interest for this article, so we wish to highlight them now.

**As. 1** (Sufficiency)**:** The factors $\mathtt{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_\mathtt{a})$ depend on the parameters only through $\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})$, so that $\mathtt{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_\mathtt{a}) = \hat{\mathtt{f}}_{\mathtt{a},\mathbf{r}}(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}))$ for all $\boldsymbol{\theta}_\mathtt{a}$.

Assumption 1 refocuses our interest on the functions $\{\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\}$ which the factors depend on the parameters through. Typically, the same function $\mathtt{t}_\mathtt{i}(\boldsymbol{\theta}_\mathtt{i})$ will appear in several vector functions $\mathbf{t}_\mathtt{a}$ (i.e. with different $\mathtt{a}$s). Take each element $\mathtt{t}_\mathtt{i}$ of the vectors $\{\mathbf{t}_\mathtt{a} | \mathtt{a} \in \{1,\dots,\mathtt{A}\}\}$ and collect them without repetition into the vector $\mathbf{t}$, so that $\mathbf{t}_\mathtt{a} \subset \mathbf{t}$ for all $\mathtt{a}$. Next, break $\mathbf{t}$ up into the disjoint concatenation of the vectors $\mathbf{v}_\mathtt{j}$ with $\mathtt{j} \in \{1,\dots,\mathtt{S}\}$, in such a way so that $\mathtt{t}_\mathtt{i}, \mathtt{t}_\mathtt{l} \in \mathbf{v}_\mathtt{j}$ and $\mathtt{t}_\mathtt{l} \in \mathbf{t}_\mathtt{a}$ implies that $\mathtt{t}_\mathtt{i} \in \mathbf{t}_\mathtt{a}$. We call the $\mathbf{v}_\mathtt{j}$s elementary basis functions. Finally, denote the vector of arguments of $\mathbf{v}_\mathtt{j}$ by $\boldsymbol{\vartheta}_\mathtt{j}$, so that $\boldsymbol{\vartheta}_\mathtt{j} \subset \boldsymbol{\theta}$, and $\boldsymbol{\vartheta}_\mathtt{j} \subset \boldsymbol{\theta}_\mathtt{a}$ for all $\mathtt{a}$ such that $\mathbf{v}_\mathtt{j} \subset \mathbf{t}_\mathtt{a}$.

We may now depict the interdependence between the parameters $\{\theta_\mathtt{i}\}$, the elementary basis functions $\{\mathbf{v}_\mathtt{j}\}$, and the factors $\mathtt{f}_\mathtt{a}$ with a tri-partite graph called a parameter basis factor graph. In this graph, the left nodes are the parameters $\{\theta_\mathtt{i}\}$, the middle nodes are the elementary basis functions $\{\mathbf{v}_\mathtt{j}\}$, and the right nodes are the factors $\mathtt{f}_\mathtt{a}$. An edge connects $\theta_\mathtt{i}$ with $\mathbf{v}_\mathtt{j}$ if and only $\theta_\mathtt{i} \in \boldsymbol{\vartheta}_\mathtt{j}$. Similarly, an edge connections $\mathbf{v}_\mathtt{j}$ with $\mathtt{f}_\mathtt{a}$ if and only if $\mathbf{v}_\mathtt{j} \subset \mathbf{t}_\mathtt{a}$. In the following development, we will denote the indices $\mathtt{a} \in \{1,\dots,\mathtt{A}\}$ of factor nodes which share an edge with the basis node $\mathbf{v}_\mathtt{j}$ by $\mathcal{F}(\mathtt{j})$. Furthermore, we will denote the indices $\mathtt{j} \in \{1,\dots,\mathtt{S}\}$ of the basis nodes which share an edge with the factor node $\mathtt{f}_\mathtt{a}$ by $\mathcal{S}(\mathtt{a})$. We shall call the bipartite subgraph formed by only the basis and factor nodes and edges between them the basis factor graph. An example of a parameter basis factor graph is shown in figure 1.

**As. 2** (Reciprocity)**:** There is a product measure $\mathtt{d}\mathbf{t} = \prod_\mathtt{i} \mathtt{d}\mathbf{v}_\mathtt{i}$ on $\mathbf{t}(\mathcal{Q})$ such that the Radon Nikodym derivative $\mu$ of the induced measure via the inverse image $\mathbf{t}^{-1}(\mathcal{A})$ from $\mathtt{d}\boldsymbol{\theta}$, with respect to the $\mathtt{d}\mathbf{t}$ measure of $\mathcal{A}$ factors into the product of functions of $\mathbf{v}_\mathtt{j}$, i.e. $\mu(\mathbf{t}) = \prod_\mathtt{j} \mu_\mathtt{j}(\mathbf{v}_\mathtt{j})$.

Perhaps the easiest way to satisfy assumption 2 is to have each parameter $\theta_\mathtt{i}$ appear in only one $\boldsymbol{\vartheta}_\mathtt{j}$, so that the parameter nodes in the parameter basis factor graph all have degree one. This is the situation depicted in Figure 1.

Under as. 1 and 2, EP can be shown to be equivalent to a message passing algorithm on the basis factor graph. In particular, these assumptions imply that (2) simplifies to

$$\begin{aligned} &\frac{\int \mathbf{v}_\mathtt{j}(\boldsymbol{\vartheta}_\mathtt{j}) \mathtt{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \exp\left(\sum_{\mathtt{l} \in \mathcal{S}(\mathtt{a})} \mathbf{v}_\mathtt{l}(\boldsymbol{\vartheta}_\mathtt{l}) \cdot \mathbf{m}_{\mathtt{l}\to\mathtt{a}}\right) \mathtt{d}\boldsymbol{\theta}_\mathtt{a}}{\int \mathtt{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \exp\left(\sum_{\mathtt{l} \in \mathcal{S}(\mathtt{a})} \mathbf{v}_\mathtt{l}(\boldsymbol{\vartheta}_\mathtt{l}) \cdot \mathbf{m}_{\mathtt{l}\to\mathtt{a}}\right) \mathtt{d}\boldsymbol{\theta}_\mathtt{a}} \\ &= \frac{\int \mathbf{v}_\mathtt{j}(\boldsymbol{\vartheta}_\mathtt{j}) \exp\left(\mathbf{v}_\mathtt{j}(\boldsymbol{\vartheta}_\mathtt{j}) \cdot (\mathbf{n}_{\mathtt{a}\to\mathtt{j}} + \mathbf{m}_{\mathtt{j}\to\mathtt{a}})\right) \mathtt{d}\boldsymbol{\vartheta}_\mathtt{j}}{\int \exp\left(\mathbf{v}_\mathtt{j}(\boldsymbol{\vartheta}_\mathtt{j}) \cdot (\mathbf{n}_{\mathtt{a}\to\mathtt{j}} + \mathbf{m}_{\mathtt{j}\to\mathtt{a}})\right) \mathtt{d}\boldsymbol{\vartheta}_\mathtt{j}} \end{aligned} \tag{3}$$

where we have introduced the left going messages $\mathbf{n}_{\mathtt{a}\to\mathtt{j}} = [\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}]_\mathtt{j}$, that are the components of $\boldsymbol{\lambda}_{\mathtt{a},\mathbf{r}}$ multiplying the elements $\mathbf{v}_\mathtt{j}$ in $\mathbf{t}_\mathtt{a}$. Equation (3) also introduces the right going messages

$$\mathbf{m}_{\mathtt{j}\to\mathtt{a}} = \sum_{\mathtt{c} \in \mathcal{F}(\mathtt{j}) \setminus \{\mathtt{a}\}} \mathbf{n}_{\mathtt{a}\to\mathtt{j}} \tag{4}$$

Emphasizing again the sequential aspects of the algorithm, under the parallel scheduling, for each $\mathtt{a} \in \{1,\dots,\mathtt{A}\}$ equation (3) is solved to update $\mathbf{n}_{\mathtt{a}\to\mathtt{j}}$ for all $\mathtt{j} \in \mathcal{S}(\mathtt{a})$. Then, for each $\mathtt{j} \in \{1,\dots,\mathtt{S}\}$ equation (4) is solved to update $\mathbf{m}_{\mathtt{j}\to\mathtt{a}}$ for all $\mathtt{a} \in \mathcal{F}(\mathtt{j})$. This two step sequence is then repeated either for a fixed number of iterations or until convergence is reached.

## 3. DENSITY EVOLUTION FOR HOMOGENOUS EP

Here we provide an extension of density evolution analysis from belief propagation decoding [10] and turbo decoding [12, 11] to general EP. To simplify the results, we consider some special cases of EP which satisfy some additional assumptions, which collectively we will refer to as *the homogeneity assumptions*.

**As. 3** (Regular Parameter Nodes)**:** Each parameter in the parameter basis factor graph has degree one. In other word, each parameter $\theta_\mathtt{i}$ appears as an argument of only one elementary basis vector $\mathbf{v}_\mathtt{j}$.

**As. 4** (Regular Basis Nodes)**:** Each elementary basis vector $\mathbf{v}_\mathtt{j}$ appears in 1 degree 1 factor node ($\mathtt{f}_\mathtt{j}$) and $\mathtt{d}_\mathtt{p}$ degree $\mathtt{d}_\mathtt{f}$ factor nodes, and is made of the same function, so that $\mathbf{v}_\mathtt{j}(\boldsymbol{\vartheta}_\mathtt{j}) = \mathbf{v}(\boldsymbol{\vartheta}_\mathtt{j})$ for all $\boldsymbol{\vartheta}_\mathtt{j}$ for some vector function $\mathbf{v}$.

**As. 5** (Symmetry)**:** For the purposes of characterizing the performance and convergence of the particular instance of EP, it suffices to condition on $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \mathcal{R}$ where

$$\mathcal{R} := \{\boldsymbol{\theta}^0 \,|\boldsymbol{\vartheta}_\mathtt{j} = \boldsymbol{\vartheta}_\mathtt{l} \forall \mathtt{j}, \mathtt{l} \in \{1,\dots,\mathtt{S}\}\}$$

Assumption 5 is equivalent to the assumption that the all zero code word was transmitted in error control/correction decoding problems. In estimation problems, it relies on symmetry of the performance of the algorithm by requiring, say, that the distribution of the messages passed to $\mathbf{v}_j$ after $k$ iterations be the same when $\boldsymbol{\theta} = \boldsymbol{\theta}^0 \in \mathcal{R}$ as when $\boldsymbol{\theta} = \boldsymbol{\theta}^1$ not a member of $\mathcal{R}$ but $\boldsymbol{\vartheta}_j^0 = \boldsymbol{\vartheta}_j^1$.

**As. 6** (Marginal Messages)**:** The marginal a posteriori density

$$\frac{\int f_a(\mathbf{r}_a, \boldsymbol{\theta}_a) \prod_{i \in \mathcal{S}(a) \setminus \{j\}} \exp\left(\mathbf{v}(\boldsymbol{\vartheta}_i) \cdot \boldsymbol{\gamma}_i\right) d\boldsymbol{\theta}_c \setminus \boldsymbol{\vartheta}_j}{\int f_a(\mathbf{r}_a, \boldsymbol{\theta}_a) \prod_{i \in \mathcal{S}(a) \setminus \{j\}} \exp\left(\mathbf{v}(\boldsymbol{\vartheta}_i) \cdot \boldsymbol{\gamma}_i\right) d\boldsymbol{\theta}_c}$$

is a minimal standard exponential family with sufficient statistics $\mathbf{v}(\cdot)$.

**As. 7** (Independent Factors)**:** The factors of degree $d_f$ depend on different observations which are conditionally independent given $\boldsymbol{\theta} = \boldsymbol{\theta}$, so that the observation vector $\mathbf{r}$ can be broken down into the disjoint[1] concatenation and reordering of smaller vectors $\{\mathbf{r}_a | a \in \{S+1, \ldots, A\}\}$, which are conditionally i.i.d. given $\boldsymbol{\theta} = \boldsymbol{\theta}_0$

$$p_{\mathbf{r}|\boldsymbol{\theta}}(\mathbf{r}|\boldsymbol{\theta}) = \prod_{a=1}^{A} p_{\mathbf{r}_a|\boldsymbol{\theta}_a}(\mathbf{r}_a|\boldsymbol{\theta}_a)$$

with $f_{a,\mathbf{r}}$ a function of only $\mathbf{r}_a$ and $\boldsymbol{\theta}_a$, i.e. $f_{a,\mathbf{r}}(\boldsymbol{\theta}_a) = f_{a,\mathbf{r}_a}(\boldsymbol{\theta}_a)$, and $p_{\mathbf{r}_a|\boldsymbol{\theta}_a}(\mathbf{r}_a|\boldsymbol{\theta}_a) = h(\mathbf{r}_a|\boldsymbol{\theta}_a) \forall \mathbf{r}_a, \boldsymbol{\theta}_a$ for all $a \in \{S+1, \ldots, A\}$.

**As. 8** (Regular Factor Nodes)**:** Each degree 1 factor node is the same function, so that

$$f_{a,\mathbf{r}_a}(\boldsymbol{\theta}_a) = w_{\mathbf{r}_a}(\boldsymbol{\theta}_a), \quad \hat{f}_{a,\mathbf{r}_a}(\mathbf{t}_a) = \hat{w}_{\mathbf{r}_a}(\mathbf{t}_a)$$

for all $a \in \{1, \ldots, S\}$. Furthermore, each degree $d_f$ factor $f_a$ is the same function, so that

$$f_{a,\mathbf{r}_a}(\boldsymbol{\theta}_a) = f_{\mathbf{r}_a}(\boldsymbol{\theta}_a), \quad \hat{f}_{a,\mathbf{r}_a}(\mathbf{t}_a) = \hat{f}_{\mathbf{r}_a}(\mathbf{t}_a)$$

for some functions $f$ and $\hat{f}$ for all $a$. Also, the functions $f$ and $\hat{f}$ are insensitive to the ordering of the elementary basis vectors, so that

$$f_{\mathbf{r}_a}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_a)) = f_{\mathbf{r}_a}(\boldsymbol{\theta}_a), \quad \hat{f}_{\mathbf{r}_a}(\boldsymbol{\pi}_{\mathbf{v}}(\mathbf{t}_a)) = \hat{f}_{\mathbf{r}_a}(\mathbf{t}_a)$$

where $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is a permutation on the elementary basis vector ordering of $\boldsymbol{\theta}_a$, i.e. $\boldsymbol{\pi}_{\boldsymbol{\theta}}([\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \ldots, \boldsymbol{\vartheta}_{S_a}]) = [\boldsymbol{\vartheta}_{j_1}, \boldsymbol{\vartheta}_{j_2}, \ldots, \boldsymbol{\vartheta}_{j_{S_a}}]$ and $\boldsymbol{\pi}_{\mathbf{v}}([\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{S_a}]) = [\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \ldots, \mathbf{v}_{j_{S_a}}]$ for $(j_1, \ldots, j_{S_a})$ is any reordering of the integers from 1 to $S_a$.

**As. 9** (Randomly Chosen Edges)**:** The subset of edges in the basis factor graph connected to degree $d_f$ factor nodes are chosen by picking randomly from $\mathcal{G}(N, K, d_p, d_f)$, the set of bipartite graphs with $N$ left nodes each of degree $d_p$ and $K$ right nodes each of degree $d_f$.

To highlight the purpose of these assumptions we must first become acquainted with some terminology. Define the *computation neighborhood* of a basis node $\{\mathbf{v}_j\}$[2] $\mathcal{C}_k(\mathbf{v}_j)$ of depth $k$ to be subgraph of the basis factor graph of nodes and edges no more than $k$ edges away from $\mathbf{v}_j$. An example of a computation neighborhood is shown in Fig. 2.

[1]Note that the degree 1 factor nodes have indices $a \in \{1, \ldots, S\}$ while the degree $d_f$ factor nodes have indices $a \in \{S+1, \ldots, A\}$.

[2]This is adapted from the term *decoding neighborhood* from [10] because we wish to consider estimation problems as well as decoding problems.

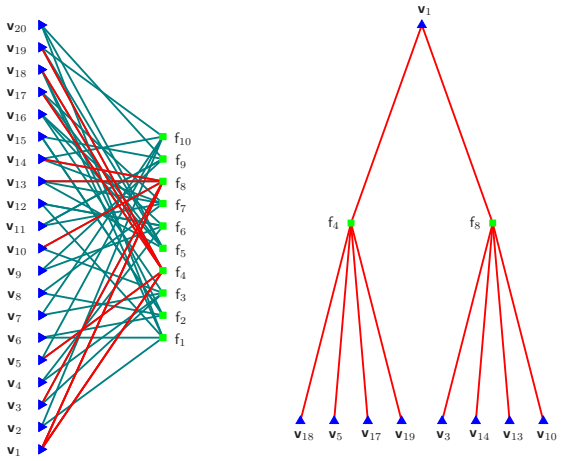


**Fig. 2**. The computation neighborhood of $\mathbf{v}_1$ of length 2.

The computation neighborhood is an important concept in EP because $\mathcal{C}_{2k}(\mathbf{v}_j)$ contains all of the factor nodes which influence the estimate of the a posteriori density provided by the EP algorithm after $k$ iterations.

As we shall prove in the upcoming theorem, As. 3-9 see to it that, with probability $\to 1$ as the size of the parameter basis factor graph $K \to \infty$, for each $k$, the messages passed in the basis factor graph during the $k$th iteration are identically distributed according to a single probability distribution. Furthermore, again with probability $\to 1$ as $K \to \infty$, at any given iteration $k$ all of the incoming messages to any particular basis node $\mathbf{v}_j$ (i.e. all incoming messages at the root of the computation neighborhood $\mathcal{C}_k(\mathbf{v}_j)$) are independent. This allows one to determine the performance of EP at any given parameter node by keeping track of the way a single probability distribution evolves over iterations.

To state this result mathematically, it is necessary to introduce the operators $\bigotimes^{d_p-1}, y\otimes, \mathfrak{F}_{\boldsymbol{\theta}_0}$ which we presently define. Given a probability density function $g$, we will denote by $\bigotimes^{d_p-1} g$ the $d_p - 1$ fold convolution of $g$ with itself. Furthermore, denote by $y\otimes$ convolution with the probability density, given $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \mathcal{R}$, for the vector

$$\boldsymbol{\Lambda}_{\mathbf{v}}^{-1}(\boldsymbol{\lambda}) \left( \frac{\int \mathbf{v}(\boldsymbol{\vartheta}_j) w(\mathbf{r}_j, \boldsymbol{\vartheta}_j) d\boldsymbol{\vartheta}_j}{\int w(\mathbf{r}_j, \boldsymbol{\vartheta}_j) d\boldsymbol{\vartheta}_j} \right)$$

where $w$ was the function for the degree one factor nodes. Here, we have introduced the canonical transformation from the theory of statistical exponential families [7] which is bijective for minimal sufficient exponential families

$$\boldsymbol{\Lambda}_{\mathbf{v}_j}(\boldsymbol{\lambda}) := \frac{\int \mathbf{v}_j(\boldsymbol{\vartheta}_j) \exp(\mathbf{v}_j(\boldsymbol{\vartheta}_j) \cdot \boldsymbol{\lambda}) d\boldsymbol{\vartheta}_j}{\int \exp(\mathbf{v}_j(\boldsymbol{\vartheta}_j) \cdot \boldsymbol{\lambda}) d\boldsymbol{\vartheta}_j}$$

Finally, denote by $\mathfrak{F}_{a \to j, \boldsymbol{\theta}_0}$ the map which takes the probability density $g_{\boldsymbol{\lambda}}$ to the probability density, given $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \mathcal{R}$, for the vector

$$\boldsymbol{\Lambda}_{\mathbf{v}_j}^{-1} \left( \frac{\int_{\mathcal{Q}_a} \mathbf{v}_j(\boldsymbol{\vartheta}_j) f_{a,\mathbf{r}}(\boldsymbol{\theta}_a) \prod_{i \in \mathcal{S}(a) \setminus \{j\}} \exp\left(\mathbf{v}(\boldsymbol{\vartheta}_i) \cdot \boldsymbol{\lambda}_i\right) d\boldsymbol{\theta}_a}{\int_{\mathcal{Q}_a} f_{a,\mathbf{r}}(\boldsymbol{\theta}_a) \prod_{i \in \mathcal{S}(a) \setminus \{j\}} \exp\left(\mathbf{v}(\boldsymbol{\vartheta}_i) \cdot \boldsymbol{\lambda}_i\right) d\boldsymbol{\theta}_a} \right)$$

where $\boldsymbol{\lambda}_{\mathtt{i}}$, $\mathtt{i} \in \mathcal{S}(\mathtt{a}) \setminus \{\mathtt{j}\}$ are i.i.d. samples from the probability distribution $\mathtt{g}_{\boldsymbol{\lambda}}$.

With these definitions we are now ready to prove that density evolution is a viable way of assessing the performance and convergence of EP under the As. 3-9 in an asymptotic sense.

**Thm. 1** (Density Evolution for EP)**:** Let the number of parameters and observations grow in such a way that As. 3-9 are all satisfied for each $\mathtt{K}$ as $\mathtt{K} \to \infty$. Then, with probability $\to 1$ as $\mathtt{K} \to \infty$, at any given iteration $\mathtt{k}$, given $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \mathcal{R}$, the incoming messages $\{\mathbf{n}_{\mathtt{a}\to\mathtt{j}}|\mathtt{a} \in \mathcal{F}(\mathtt{j})\}$ to any basis node $\mathbf{v}_{\mathtt{j}}$ are independent and identically distributed samples from a probability distribution $\mathtt{q}_{\mathtt{k}}$ obeying the recursion

$$\mathtt{q}_{\mathtt{k}+1} = \left( \mathfrak{F}_{\boldsymbol{\theta}_0} \circ \mathtt{y} \otimes \circ \bigotimes^{\mathtt{d_p}-1} \right) \mathtt{q}_{\mathtt{k}} \tag{5}$$

where $\mathfrak{F}_{\boldsymbol{\theta}_0}$ can be chosen as any $\mathfrak{F}_{\mathtt{a}\to\mathtt{j},\boldsymbol{\theta}_0}$.

**Proof**: It was shown in [10] that for any fixed $\mathtt{k}$, the assumption 9 gaurantees that the computation neighborhood $\mathcal{C}_{\mathtt{k}}(\mathbf{v}_{\mathtt{j}})$ is a tree with probability $\to 1$ as $\mathtt{K} \to \infty$. Only those factor nodes which are in the computation tree $\mathcal{C}_{2\mathtt{k}}(\mathbf{v}_{\mathtt{j}})$ affect those messages passed to $\mathbf{v}_{\mathtt{j}}$ during the $\mathtt{k}$th iteration. Furthermore, As. 5 and 7 require the observations $\mathbf{r}_{\mathtt{a}}$ included in this computation tree to be independently and identically distributed given $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \mathcal{R}$. This gives that the message passed at the top edges of the computation tree (to $\mathbf{v}_{\mathtt{j}}$) are independent. As. 3,4, and 8 which guarantee that all of the factors and basis functions, and thus the message creating functions at the factor nodes, are the same, then imply that not only are these incoming messages to any basis node $\mathbf{v}_{\mathtt{j}}$ independent, but they are also identically distributed. Thus, the probability density for the message at the output of the factor node can be calculated by taking the probability distribution corresponding to $\mathtt{d_f} - 1$ i.i.d. samples from the input message distribution, and putting it through the factor message update (3). Solving (3) is then equivalent to calculating $\mathbf{H}_{\mathtt{a}\to\mathtt{j}}(\mathbf{r}, \boldsymbol{\lambda})$ where $\boldsymbol{\lambda}$ are the samples from the incoming message probability distribution.

At the basis node, the message passing rules (4) require that all but one of the incoming messages be summed to create each output message. Since we have just proved that these messages are independent and identically distributed, this operation creates an outgoing (to the factor nodes) message with probability density that is the convolution of the incoming distribution with itself $\mathtt{d_p} - 1$ times and then with the degree one factor node message density. □

**Cor. 1** (Parameter Distribution Estimate)**:** Given $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \mathcal{R}$, the estimate $\hat{\mathtt{p}}^{\mathtt{k}}_{\boldsymbol{\vartheta}_{\mathtt{j}}|\mathbf{r}}$ provided by EP for the a posteriori distribution $\mathtt{p}_{\boldsymbol{\vartheta}_{\mathtt{j}}|\mathbf{r}}$ after $\mathtt{k}$ iterations is given by

$$\hat{\mathtt{p}}^{\mathtt{j}}_{\boldsymbol{\vartheta}_{\mathtt{j}}|\mathbf{r}} = \exp\left(\mathbf{v}(\boldsymbol{\vartheta}_{\mathtt{j}}) \cdot \boldsymbol{\lambda} - \psi_{\mathbf{v}}(\boldsymbol{\lambda})\right)$$

where $\boldsymbol{\lambda}$ is distributed according to the probability density function $\mathtt{h}_{\mathtt{j}}$ defined by $\mathtt{h}_{\mathtt{k}} := \bigotimes^{\mathtt{d_p}} \mathtt{q}_{\mathtt{k}}$ and $\mathtt{q}_{\mathtt{k}}$ is defined by the recursion (5).

Theorem 1 and Corollary 1 are immensely useful, because they show that determining the performance and convergence of EP under the homogeneity As. 3-9 is equivalent to studying the convergence properties of the iterated map (5). This is, in essence, the theory of density evolution.

## 4. CONCLUSIONS

This paper showed that density evolution style analysis can be applied to general expectation propagation under some homogeneity and symmetry assumptions. Future work will then apply density evolution to determine the performance of particular instances of expectation propagation, for example ones used for distributed estimation in sensor networks.

## 5. REFERENCES


[1] T. P. Minka, *A Family of Algorithms for Approximate Bayesian Inference*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[2] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertainty in AI'01*, 2001.

[3] J. M. Walsh and P. A. Regalia, "Iterative constrained maximum likelihood estimation via expectation propagation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[4] J. M. Walsh, "Dual Optimality Frameworks for Expectation Propagation," in *IEEE Conference on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, June 2006.

[5] J. Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*, Morgan Kaufmann Publishers, 1988.

[6] F. R. Kshischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[7] Lawrence D. Brown, *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*, Institute of Mathematical Statistics, 1986.

[8] S. Amari, *Methods of Information Geometry*, vol. 191, AMS Translations of Mathematical Monographs, 2004.

[9] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," Tech. Rep., Department of Statistics, University of California, Berkeley.

[10] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes.," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[12] H. El Gamal and A. R. Hammons, Jr., "Analyzing the turbo decoder using the gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[13] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes.," in *ICC 93*, Geneva, May 1993, vol. 2, pp. 1064–1070.

[14] R. G. Gallager, "Low-density parity-check codes.," *IRE Trans. Information Theory*, vol. 2, pp. 21 28, 1962.

[15] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution.," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 891–907, May 2001.