Belief Propagation, Dykstra's Algorithm, and Iterated Information Projections

John MacLaren Walsh, *Member, IEEE,* Phillip A. Regalia *Fellow, IEEE,*

***Abstract*—Belief propagation is shown to be an instance of a hybrid between two projection algorithms in the convex programming literature: Dykstra's algorithm with cyclic Bregman projections, and an alternating Bregman projections algorithm. Via this connection, new results concerning the convergence and performance of belief propagation can be proven by exploiting the corresponding literature about the two projections algorithms it hybridizes. In this regard, it is identified that the lack of guaranteed convergence for belief propagation results from the asymmetry of its Bregman divergence by proving that when the associated hybrid projection algorithm generalization is used with a symmetric Bregman divergence, it always converges. Additionally, by characterizing factorizations that are close to acyclic in a manner independent of their girth, a new collection of distributions for which belief propagation is guaranteed to perform well is identified using the new projection algorithm framework.**

***Index Terms*—belief propagation; convex programming; information projections; information geometry; projections algorithms**

I. Introduction

The venerable belief propagation algorithm [1] has emerged over the past two decades as a robust and widely applicable estimation procedure in an impressive array of problem settings. The algorithm may be formulated as a "fast" method for calculating marginal probabilities [2] when a certain factor graph assumes a forest structure, yet its real power lies in its success when applied to factor graphs displaying loops. The seminal practical confirmation occurred with the advent of turbo codes [3], [4] which breached previous barriers in approaching the Shannon limit of reliable communications over a noisy channel. Further validation followed with the resurrection [5] of low density parity check codes [6], and in relatively short time successful applications of belief propagation have been reported in network diagnostics [7], sensor self-localization [8], distributed inference in sensor networks [9], [10], and multi-user communications [11], [12], [13], among other problems underlying multi-terminal information theory (see, e.g., the references in [14], [15]). Additional terrain has been charted with extensions to continuous probability distributions using expectation propagation [16], [9], which encompasses belief propagation in particular settings.

The impressive range of applications is tempered by the iterative nature of belief propagation, since convergence of the procedure has not been proved in the general case. The various formal results available appeal to asymptotic approximations [17], [18], [19], [20] in the problem dimension, isolating conditions under which the factor graph girth (which is the number of iterations before the shortest loop in the factor graph closes itself) becomes arbitrarily large; the loops nearly disappear and the dependence graph becomes locally a tree, on which belief propagation performs an exact marginal probability calculation. Although such results confirm the favorable behavior in such asymptotic settings, the approximations are known to break down for more reasonable problem sizes, and indeed, specific instances of misconvergence and chaotic behavior are catalogued in [21].

The intent of this work is to rephrase belief propagation as an iterative projection algorithm using information geometry. Various information-geometric interpretations of belief propagation have emerged over the years: early instances are identified in Moher and Gulliver [22] and Grant [23] in the study of turbo decoding [24] and bit-interleaved coded modulation [25], and tacitly underlies Richardson's treatment [26], as clarified by Ikeda *et al.* [27], [28]. The identified shortcomings of those prior approaches concerns the difficulty in phrasing extrinsic information extraction (the key "recipe" ingredient behind such iterative schemes) as projections on invariant sets [29]. The situation is to be contrasted with existing information projection algorithms of Csiszár [30], [31] and Dykstra [32], which are provably convergent, and have since been extended to projection algorithms for convex programming using Bregman divergences [33], [34], [35], [36]. We review several of these Bregman projections algorithms for convex programming in §II.

Our main result, presented in §III, rephrases the belief propagation algorithm in a form showing greater affinity with several of these convex programming algorithms, as first recognized in [37], [16] in the context of expectation propagation. In particular, we show that belief propagation may be interpreted as an instance of a hybrid between two iterative projections convex programming algorithms: alternating Bregman projections, and Dykstra's algorithm with cyclic Bregman projections. This more general hybrid projections algorithm is then dubbed the belief propagation Bregman projections algorithm.

More recent work by Alberge [24] has also noted the similarity between iterative decoding and Dykstra's algorithm, but

Manuscript received June 10, 2008, revised August 27, 2009, revised again January 13, 2010.

J. M. Walsh is with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA. J. M. Walsh was supported by the National Science Foundation under grant CCF-0728496. P. A. Regalia is with the Catholic University of America, Washington, DC, and an Adjunct Researcher with the Institut Telecom SudParis in Evry, France. P. A. Regalia was supported by the National Science Foundation under grant CCF-0728521. Contact J. M. Walsh at jwalsh@ece.drexel.edu for information regarding this paper.

arrives at a projection algorithm formulation which requires that the sets on which to project depend on the point that is to be projected, and thus deviates significantly from Dykstra's algorithm which does not allow this. Our alternate formulation, the belief propagation Bregman projections algorithm, does not suffer this problem, and can be explained exclusively in terms of an adaptation of Dykstra's algorithm with Bregman projections on invariant sets.

While belief propagation is not properly a convex optimization procedure, two advantages of fitting belief propagation within this more general class of projections algorithms are nonetheless demonstrated. The first shows that the lack of guaranteed convergence for belief propagation stems from the asymmetry of its Bregman divergence. We prove this by showing that when the associated hybrid projection algorithm generalization is used with a *symmetric* Bregman divergence instead of the relative entropy, it always converges. Furthermore, it is empirically observed that this modified algorithm converges near a two-step projection reminiscent of marginalization in several example cases, although the convergent point does not always have this property, as another example shows. Narrowing down the set of joint distributions for which belief propagation proves effective leads to the second instance that validates the new framework, which is discussed in §V. This second instance broadens the class of known factorizations of joint distributions for which conventional belief propagation provides estimates close to the true marginals after a finite number of iterations. This is achieved by characterizing factorizations that are close to acyclic in a manner independent of their girth by appealing to the continuity of the presented hybrid projections algorithm. Although we consider the binary random vector case for pedagogical simplicity, the results extend to arbitrary discrete random variables and some continuous random variables via expectation propagation, as developed in section VI. Finally, section VII concludes the paper, discussing several prominent avenues for future research.

II. Background

We collect in this section some basic definitions related to Bregman divergences and projections, briefly mentioning some examples from information geometry.

A. Bregman divergence

Consider a convex function $f(\mathbf{q})$ defined over some convex domain $\mathcal{D} \subseteq \mathbb{R}^{\mathcal{N}}$. The graph of any convex function is lower bounded by any tangent hyperplane [38], as depicted in Fig. 1. Assuming $f(\cdot)$ is differentiable at $\mathbf{q}$, this induces the gradient inequality

$$f(\mathbf{r}) \geq f(\mathbf{q}) + \langle \nabla f(\mathbf{q}), \mathbf{r} - \mathbf{q} \rangle, \qquad \text{for all } \mathbf{r} \in \mathcal{D},$$

where $\nabla f(\mathbf{q}) = df(\mathbf{q})/d\mathbf{q}$ is the gradient vector, and $\langle \cdot, \cdot \rangle$ denotes the standard inner product. If $f(\mathbf{q})$ is strictly convex (meaning here that strict inequality holds everywhere in the gradient inequality except $\mathbf{q} = \mathbf{r}$) and differentiable, the *Bregman divergence* [33], [39] induced by f is defined as

$$D_f(\mathbf{r}, \mathbf{q}) \triangleq f(\mathbf{r}) - f(\mathbf{q}) - \langle \nabla f(\mathbf{q}), \mathbf{r} - \mathbf{q} \rangle \geq 0$$

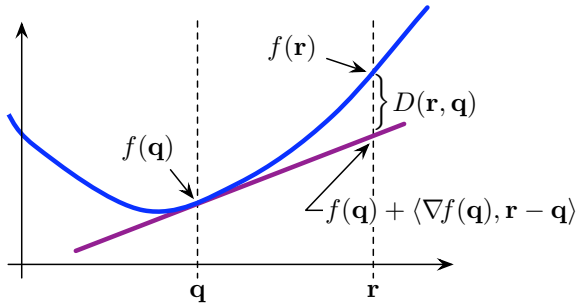

Fig. 1. Illustrating the gradient inequality, applicable to any differentiable convex function.

As f is assumed strictly convex, the Bregman divergence vanishes if and only if $\mathbf{r} = \mathbf{q}$.

Some important examples of Bregman divergences include:

Example 1 (Euclidean distance squared)**:** for which $\mathcal{D} = \mathbb{R}^{\mathcal{N}}$ and $f(\mathbf{q}) = \frac{1}{2}\|\mathbf{q}\|_2^2 = \frac{1}{2}\sum_{i=1}^{N} q_i^2$ yielding $D_f(\mathbf{r}, \mathbf{q}) = \frac{1}{2}\|\mathbf{r} - \mathbf{q}\|_2^2$.

Example 2 (Mahalanobis Distance Squared)**:** for which $\mathcal{D} = \mathbb{R}^{\mathcal{N}}$ and $f(\mathbf{q}) = \mathbf{q}^T\mathbf{A}\mathbf{q}$ for some positive definite symmetric matrix $\mathbf{A}$, yielding $D_f(\mathbf{r}, \mathbf{q}) = (\mathbf{r} - \mathbf{q})^T\mathbf{A}(\mathbf{r} - \mathbf{q})$.

Example 3 (Kullback Leibler Divergence)**:** Let $\mathbf{q} = [q_1, \ldots, q_N]$ together with $q_0 = 1 - \sum_{i=1}^{N} q_i$ collect the evaluations of a probability mass function defined on $N+1$ outcomes. The negative Shannon entropy

$$f(\mathbf{q}) = \sum_{i=0}^{N} q_i \log q_i$$

is convex [40] with domain

$$\mathcal{D} = \left\{ \mathbf{q} \,\middle|\, q_i \geq 0 \; \forall i \in \{1, \ldots, N\}, \;\; \sum_{i=1}^{N} q_i \leq 1 \right\},$$

and its induced Bregman divergence is readily calculated as

$$D_f(\mathbf{r}, \mathbf{q}) = \sum_{i=0}^{N} r_i \log \frac{r_i}{q_i} = I(\mathbf{r} \| \mathbf{q})$$

which is the Kullback-Leibler divergence (a.k.a. relative entropy or information divergence) between $\mathbf{r}$ and $\mathbf{q}$.

To every convex function $f(\mathbf{q})$ is associated a conjugate function $f^*(\boldsymbol{\theta})$ defined through ([38] pp. 104)

$$f^*(\boldsymbol{\theta}) = \sup_{\mathbf{q}} \Big(\langle \mathbf{q}, \boldsymbol{\theta} \rangle - f(\mathbf{q}) \Big).$$

The conjugate function $f^*(\boldsymbol{\theta})$ is likewise convex, and, under some additional regularity conditions (namely, that f be of *Legendre type*, i.e., strictly convex and differentiable everywhere inside an open and convex domain with $|\nabla f(\mathbf{q}^i)| \to \infty$ for any sequence $\mathbf{q}^i$ approaching a boundary, [38] pp. 258), the gradients $\nabla f(\mathbf{q})$ and $\nabla f^*(\boldsymbol{\theta})$ are inverse maps to each other [38], in which case f^* and f are said to form a Legendre transform pair.

Example 4 (Log Partition Function)**:** The conjugate function $f^*(\boldsymbol{\theta})$ to the negative Shannon entropy is [27], [28], [31]

$$f^*(\boldsymbol{\theta}) = \sup_{\mathbf{q}} \Big(\langle \mathbf{q}, \boldsymbol{\theta} \rangle - f(\mathbf{q}) \Big) = \log \left(1 + \sum_{i=1}^{N} \exp(\theta_i) \right)$$

with $\theta_i = \log \frac{q_i}{q_0}$, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N]^T$, and is recognized as the log partition function of thermodynamics. The domain $\mathcal{D}^*$ consists of all vectors in $(\mathbb{R} \cup \{\pm\infty\})^N$ (i.e., all of the extended reals). Note that the gradients of f and f^* form inverses to one another, so that they form a Legendre transform pair.

Under these regularity conditions (that f be of Legendre type), the conjugate function f^* can be rewritten as

$$\begin{aligned} f^*(\boldsymbol{\theta}) &= \sup_{\mathbf{q}} \Big(\langle \mathbf{q}, \boldsymbol{\theta} \rangle - f(\mathbf{q}) \Big) \\ &= \langle (\nabla f)^{-1}(\boldsymbol{\theta}), \boldsymbol{\theta} \rangle - f((\nabla f)^{-1}(\boldsymbol{\theta})) \\ &= \langle \nabla f^*(\boldsymbol{\theta}), \boldsymbol{\theta} \rangle - f(\nabla f^*(\boldsymbol{\theta})) \end{aligned}$$

and hence induces a Bregman divergence

$$\begin{aligned} D_{f^*}(\boldsymbol{\rho}, \boldsymbol{\theta}) &= f^*(\boldsymbol{\rho}) - f^*(\boldsymbol{\theta}) - \langle \nabla f^*(\boldsymbol{\theta}), \boldsymbol{\rho} - \boldsymbol{\theta} \rangle \\ &= \langle \nabla f^*(\boldsymbol{\rho}), \boldsymbol{\rho} \rangle - f(\nabla f^*(\boldsymbol{\rho})) - \langle \nabla f^*(\boldsymbol{\theta}), \boldsymbol{\rho} - \boldsymbol{\theta} \rangle \\ &\quad - (\langle \nabla f^*(\boldsymbol{\theta}), \boldsymbol{\theta} \rangle - f(\nabla f^*(\boldsymbol{\theta}))) \\ &= f(\nabla f^*(\boldsymbol{\theta})) - f(\nabla f^*(\boldsymbol{\rho})) - \langle \boldsymbol{\rho}, \nabla f^*(\boldsymbol{\theta}) - \nabla f^*(\boldsymbol{\rho}) \rangle \\ &= D_f \left(\nabla f^*(\boldsymbol{\theta}), \nabla f^*(\boldsymbol{\rho}) \right) \end{aligned}$$

This effectively swaps the arguments of D_f and uses a reparameterization through the map $\nabla f(\cdot)$ of the domain. In other words, for f of Legendre type, letting $\boldsymbol{\rho} = \nabla f(\mathbf{r})$ and $\boldsymbol{\theta} = \nabla f(\mathbf{q})$, we have

$$D_f(\mathbf{r}, \mathbf{q}) = D_{f^*}(\boldsymbol{\theta}, \boldsymbol{\rho}) \tag{1}$$

for all $\mathbf{r}, \mathbf{q} \in \mathcal{D}$.

Example 5 (Bregman Divergence for the log partition function)**:** The induced Bregman divergence D_{f^*} from the log partition function f^* is again the Kullback Leibler divergence

$$\begin{aligned} D_{f^*}(\boldsymbol{\rho}, \boldsymbol{\theta}) &= \log \frac{1 + \sum_{i=1}^{N} \exp(\rho_i)}{1 + \sum_{i=1}^{N} \exp(\theta_i)} - \sum_{i=0}^{N} \frac{\exp(\theta_i)\,(\rho_i - \theta_i)}{1 + \sum_{j=1}^{N} \exp(\theta_j)} \\ &= I(\mathbf{q} \| \mathbf{r}) \end{aligned}$$

written in terms of the log coordinates, and with the arguments switched with respect to the previous example.

As the previous examples show, the Bregman divergence may or may not be symmetric. In fact, the following lemma shows that the class of symmetric Bregman divergences is rather small:

Lemma 1 (Mahalanobis Divergences are the only Symmetric Bregman Divergences)**:** The only twice differentiable symmetric Bregman divergences are the Mahalanobis squared divergences: $D_f(\mathbf{r}, \mathbf{q}) = (\mathbf{r} - \mathbf{q})^T \mathbf{A} (\mathbf{r} - \mathbf{q})$ with $\mathbf{A}$ a positive definite symmetric matrix.

Proof: Let f be a twice continuously differentiable strictly convex function of Legendre type, and let $\mathcal{H}_{\mathbf{z}}$ denote the Hessian operator (second order derivative matrix operator w.r.t. the variables $\mathbf{z}$). As $D_f(\mathbf{r}, \mathbf{q}) = f(\mathbf{r}) - f(\mathbf{q}) - \langle \nabla f(\mathbf{q}), \mathbf{r} - \mathbf{q} \rangle$, then for any fixed $\mathbf{q}$ we have

$$\mathcal{H}_{\mathbf{r}} D_f(\mathbf{r}, \mathbf{q}) = \mathcal{H}_{\mathbf{r}} f(\mathbf{r}) \tag{2}$$

On the other hand, since for a symmetric Bregman divergence

$$\begin{aligned} 0 &= D_f(\mathbf{r}, \mathbf{q}) - D_f(\mathbf{q}, \mathbf{r}) \\ &= 2f(\mathbf{r}) - 2f(\mathbf{q}) - \langle \nabla f(\mathbf{r}) + \nabla f(\mathbf{q}), \mathbf{r} - \mathbf{q} \rangle \end{aligned} \tag{3}$$

we have, after taking a gradient w.r.t. $\mathbf{q}$

$$0 = \nabla f(\mathbf{r}) - \nabla f(\mathbf{q}) - \mathcal{H}_{\mathbf{q}} f(\mathbf{q})(\mathbf{r} - \mathbf{q}).$$

That is,

$$\nabla f(\mathbf{r}) = \nabla f(\mathbf{q}) + \mathcal{H}_{\mathbf{q}} f(\mathbf{q})(\mathbf{r} - \mathbf{q}).$$

Plugging this into (3) we have

$$0 = 2f(\mathbf{r}) - 2f(\mathbf{q}) - 2\langle \nabla f(\mathbf{q}), \mathbf{r} - \mathbf{q} \rangle - (\mathbf{r} - \mathbf{q})^T \mathcal{H}_{\mathbf{q}} f(\mathbf{q})(\mathbf{r} - \mathbf{q}),$$

which is readily recognized as

$$2D_f(\mathbf{r}, \mathbf{q}) = (\mathbf{r} - \mathbf{q})^T \mathcal{H}_{\mathbf{q}} f(\mathbf{q})(\mathbf{r} - \mathbf{q}).$$

For any fixed $\mathbf{q}$, this is clearly quadratic in $\mathbf{r}$; taking the Hessian of this expression w.r.t. $\mathbf{r}$ then gives

$$\mathcal{H}_{\mathbf{r}} D_f(\mathbf{r}, \mathbf{q}) = \mathcal{H}_{\mathbf{q}} f(\mathbf{q}).$$

When combined with (2), this reveals

$$\mathcal{H}_{\mathbf{q}} f(\mathbf{q}) = \mathcal{H}_{\mathbf{r}} f(\mathbf{r}) \qquad \text{for all } \mathbf{q}, \mathbf{r} \in \mathcal{D}.$$

This says that the Hessian matrix is constant, so that $f(\cdot)$ is quadratic. The requirement that f be strictly convex then yields the positive definite requirement. ■

B. Bregman Projections

Let $f(\mathbf{q})$ be a strictly convex function, $D_f(\mathbf{r}, \mathbf{q})$ its induced Bregman divergence, and $\mathcal{C}$ a convex subset of the domain $\mathcal{D}$. Suppose $\mathbf{q}$ is in $\mathcal{D}$ but not in $\mathcal{C}$. The *Bregman projection* [33], [39], [41] $\overleftarrow{\pi}^{\mathcal{C}}_f(\mathbf{q})$ of $\mathbf{q}$ onto $\mathcal{C}$ is the solution to the best approximation problem

$$\overleftarrow{\pi}^{\mathcal{C}}_f(\mathbf{q}) = \arg\min_{\mathbf{r} \in \mathcal{C}} D_f(\mathbf{r}, \mathbf{q})$$

and is characterized by the inequality

$$D_f(\mathbf{r}, \mathbf{q}) \geq D_f\Big(\mathbf{r}, \overleftarrow{\pi}^{\mathcal{C}}_f(\mathbf{q})\Big) + D_f\Big(\overleftarrow{\pi}^{\mathcal{C}}_f(\mathbf{q}), \mathbf{q}\Big), \qquad \forall \mathbf{r} \in \mathcal{C},$$

or its equivalent form

$$\langle \nabla f(\mathbf{q}) - \nabla f(\overleftarrow{\pi}^{\mathcal{C}}_f(\mathbf{q})), \overleftarrow{\pi}^{\mathcal{C}}_f(\mathbf{q}) - \mathbf{r} \rangle \geq 0 \quad \forall \mathbf{r} \in \mathcal{C}.$$

When f is additionally of Legendre type, then the Bregman projection $\overleftarrow{\pi}^{\mathcal{C}}_{f^*}(\boldsymbol{\theta})$ associated with the conjugate function onto a convex set $\mathcal{C} \subseteq \mathcal{D}^*$, when mapped through the coordinate

change $\nabla f(\cdot)$, may be alternatively interpreted as a certain *right Bregman projection* onto $\hat{\mathcal{C}} \triangleq \nabla f(\mathcal{C})$ defined as

$$\overrightarrow{\pi}_f^{\hat{\mathcal{C}}}(\mathbf{q}) \triangleq \arg\min_{\mathbf{r}\in\hat{\mathcal{C}}} D_f(\mathbf{q},\mathbf{r}).$$

This is because

$$\begin{aligned}\nabla f\left(\overleftarrow{\pi}_{f^*}^{\mathcal{C}}(\boldsymbol{\theta})\right) &= \nabla f\left(\arg\min_{\boldsymbol{\rho}\in\mathcal{C}} D_{f^*}(\boldsymbol{\rho},\boldsymbol{\theta})\right)\\ &= \nabla f\left(\arg\min_{\boldsymbol{\rho}\in\mathcal{C}} D_f(\mathbf{q},\nabla f(\rho))\right)\\ &= \arg\min_{\mathbf{r}\in\hat{\mathcal{C}}} D_f(\mathbf{q},\mathbf{r}) = \overrightarrow{\pi}_f^{\hat{\mathcal{C}}}(\mathbf{q}), \quad \text{by (1).}\end{aligned}$$

Some common examples of Bregman projections include the normal orthogonal projections (in which case $f = f^* = \frac{1}{2}\|\cdot\|_2^2$), and information projections in information geometry [27], [28], [31] (in which case f is the negative Shannon entropy or log partition function). Additional examples of information projections, including marginalization and code book membership enforcement as they relate to coding theory and belief propagation decoding, may be found in the excellent articles [27], [28], [23], [25], [22], [26].

C. Bregman Projections Algorithms

We assemble in this section some standard convex approximation problems and the iterative algorithms that have been devised to solve them. An early instance was proposed by Csiszár [42], but shown subsequently by Dykstra [32] not to converge to the correct solution in all cases; an improved algorithm was developed in [32] to correct this shortcoming. Further developments of interest have been developed by Bauschke and co-workers [43], [36], [41], who extended both the class of algorithms to which such iterative procedures apply and the class of functionals to be used as divergence measures. The first two iterations of the various methods are illustrated in Figure 2 for the special case in which the Bregman divergence is selected to be the squared Euclidean distance.

1) Feasibility and Best Approximation: Suppose the domain $\mathcal{C}$ in the best approximation problem $\overleftarrow{\pi}_f^{\mathcal{C}}(\mathbf{q}) = \arg\min_{\mathbf{r}\in\mathcal{C}} D_f(\mathbf{r},\mathbf{q})$ can be expressed as the intersection of convex sets:

$$\mathcal{C} = \bigcap_{n=1}^{N} \mathcal{C}_n$$

Letting $\overleftarrow{\pi}_f^{(n)}(\cdot)$ denote the Bregman projector onto $\mathcal{C}_{[(n-1) \bmod N]+1}$, a natural way to attempt to solve this problem is the method of cyclic Bregman projections [39], in which the recursion

$$\mathbf{r}_n := \overleftarrow{\pi}_f^{(n)}\mathbf{r}_{n-1}$$

is initialized with $\mathbf{r}_0 \triangleq \mathbf{q}$. An example from [32], however, shows that this need not converge to the correct solution.

Alternatively, under a few regularity conditions, the following iterative algorithm [34], [35], [36], dubbed Dykstra's algorithm with cyclic Bregman projections,

$$\mathbf{r}_n = \overleftarrow{\pi}_f^{(n)}\left(\nabla f^*\left(\nabla f(\mathbf{r}_{n-1}) + \mathbf{s}_{n-N}\right)\right) \quad (4)$$

$$\mathbf{s}_n = \nabla f(\mathbf{r}_{n-1}) + \mathbf{s}_{n-N} - \nabla f(\mathbf{r}_n) \quad (5)$$

yields a sequence $\{\mathbf{r}_n\}$ which is provably convergent to the solution to to the best approximation problem: $\lim_{n\to\infty}\mathbf{r}_n = \overleftarrow{\pi}_f(\mathbf{q})$. The algorithm is initialized according to

$$\mathbf{r}_0 = \mathbf{q}, \qquad \mathbf{s}_{-(N-1)} = \cdots = \mathbf{s}_{-1} = \mathbf{s}_0 = \mathbf{0}.$$

2) Minimum Divergence: Consider two convex sets $\mathcal{C}_1 \subseteq \mathcal{D}$ and $\hat{\mathcal{C}}_2 \subseteq \mathcal{D}^*$ having no intersection in the sense that when $\mathcal{C}_2 \triangleq \nabla f(\hat{\mathcal{C}}_2)$, $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$. A natural desire is to find a pair of vectors $\mathbf{r}_* \in \mathcal{C}_1$ and $\mathbf{q}_* \in \mathcal{C}_2$ which are closest to each other, i.e.,

$$D_f(\mathbf{r}_*,\mathbf{q}_*) = \inf_{\mathbf{r}\in\mathcal{C}_1,\mathbf{q}\in\mathcal{C}_2} D_f(\mathbf{r},\mathbf{q}).$$

One may readily conceive of algorithms which attempt to solve this problem by projecting onto $\mathcal{C}_1$ and $\mathcal{C}_2$ in an alternating manner, yielding the alternating projections algorithm

$$\mathbf{r}_n = \overleftarrow{\pi}_f^{\mathcal{C}_1}(\mathbf{q}_{n-1}) \quad (6)$$

$$\mathbf{q}_n = \overrightarrow{\pi}_f^{\mathcal{C}_2}(\mathbf{r}_n) \quad (7)$$

studied in [41], and in earlier in [30] for information projections.

Although Dykstra's algorithm with cyclic Bregman projections is built solely from left projections, and is thus not suited to such an alternating projections context, in the special case in which $f(\mathbf{q}) = \frac{1}{2}\|\mathbf{q}\|^2$ (using the Euclidean norm), we have $D_f(\mathbf{r},\mathbf{q}) = \frac{1}{2}\|\mathbf{r}-\mathbf{q}\|^2$ and $\overleftarrow{\pi}_f(\cdot) = \overrightarrow{\pi}_f(\cdot)$ reduces to the orthogonal projection operator $\pi(\cdot)$ of Euclidean space. In this case Dykstra's algorithm with cyclic Bregman projections can be applied to the minimum distance problem, yielding the iteration

$$\begin{aligned}\mathbf{r}_n &= \pi_1(\mathbf{q}_{n-1} + \mathbf{v}_{n-1})\\ \mathbf{v}_n &= \mathbf{q}_{n-1} + \mathbf{v}_{n-1} - \mathbf{r}_n\\ \mathbf{q}_n &= \pi_2(\mathbf{r}_n + \mathbf{w}_{n-1})\\ \mathbf{w}_n &= \mathbf{r}_n + \mathbf{w}_{n-1} - \mathbf{q}_n\end{aligned}$$

which is shown in [43] to converge to the minimizing solution, in which $\pi_1(\cdot)$ [resp., $\pi_2(\cdot)$] is the orthogonal projector onto $\mathcal{C}_1$ (resp., $\mathcal{C}_2$).

III. Belief Propagation as an Instance of a Modified Dykstra's Bregman Projections Algorithm

We review here the belief propagation algorithm ([1], [2], [44]), which is an iterative algorithm that attempts to calculate the marginal probabilities of a given joint probability or likelihood function. Our main result rephrases the algorithm as an iterative projection algorithm. We treat the case of binary variables for tractability; the same ideas can be developed for more general data sets and even extended to expectation propagation over continuous probability densities as we show in Section VI.

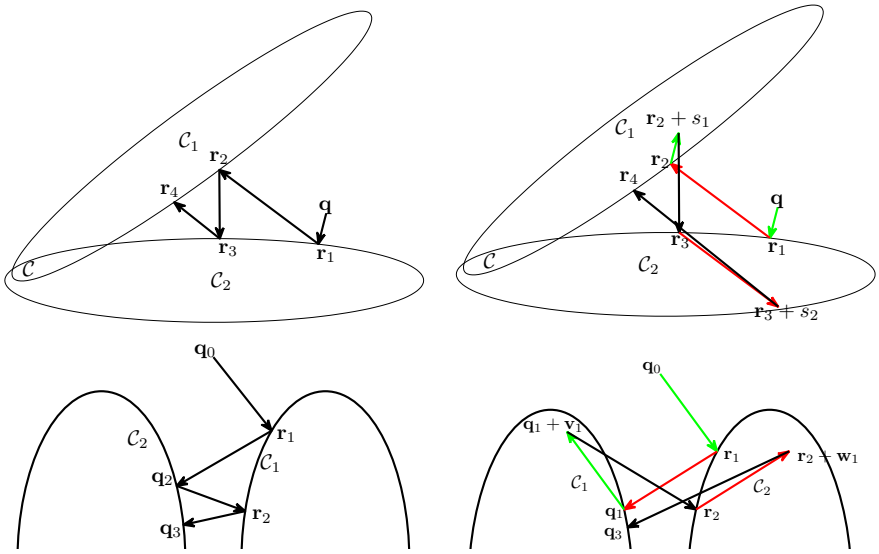

Fig. 2. First two iterations of the alternating projections (left) and Dykstra's algorithm (right) for solving the best approximation (top) and minimum divergence problems (bottom) with the Bregman divergence selected as half the squared Euclidean distance.

For the present treatment, consider again the situation in which we aim to deduce M bits collected as $\mathbf{x} = [x_1, \ldots, x_M]$ based on an observation vector $\mathbf{y}$ and a likelihood function $p(\mathbf{y}|\mathbf{x})$. The belief propagation algorithm is attractive when the likelihood function factors into a product as

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{K} g_k(\mathbf{x})$$

Here the functions $\{g_k\}$ are tacitly parametrized through $\mathbf{y}$. A given g_k will typically depend only on a subset of the variables in $\mathbf{x}$, in the sense that there exists some smallest subset $\mathbf{x}_{\mathcal{M}(k)}$ of the variables $\mathbf{x}$ ($\mathcal{M}(k) \subset \{1, \ldots, M\}$) and a function $\hat{g}_k(\mathbf{x}_{\mathcal{M}(k)})$ such that $g_k(\mathbf{x}) = \hat{g}_k(\mathbf{x}_{\mathcal{M}(k)})$ for all $\mathbf{x} \in \{0,1\}^M$. This observation is key to providing computationally efficient implementations of belief propagation. For ease of exposition, however, we will not burden the notation with further subsets of the variables from $\mathbf{x}$, and hence will use the equivalent functions $g_k(\mathbf{x})$ instead of $\hat{g}_k(\mathbf{x}_{\mathcal{M}(k)})$. It is easily verified that there is no mathematical difference between the belief propagation algorithms that arise from these two forms, and this form is more amenable for understanding the dynamics of belief propagation in the light of convex programming projections algorithms.

A classic example of such a factored form is when the observations in vector $\mathbf{y}$ are the output of a memoryless channel with $\mathbf{x}$ as its input, giving

$$\mathbf{p}(\mathbf{y}|\mathbf{x}) = C(\mathbf{x}) \prod_{i=1}^{M} p(y_i|x_i)$$

in which $C(\mathbf{x})$ is the indicator function for the code which equals one when $\mathbf{x}$ is a code word and zero otherwise, which may itself factor into simpler indicator functions. Other examples where belief propagation may be applied include network diagnostics (e.g., [7], [9], [10]), channel estimation, and self-localization in sensor networks (e.g., [8]), in addition to general inference problems [1].

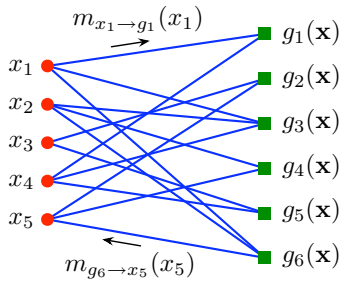

Fig. 3. Example factor graph, illustrating messages passed between variable nodes (on the left) and factor nodes (on the right).

From the factored likelihood function, we may sketch a factor graph [2] as in Figure 3, in which the factor nodes on the right represent the functions $\{g_k(\mathbf{x})\}$ and the variable nodes on the left designate the variables $\{x_i\}$. An edge (or branch) in the graph connects a variable node x_i to factor node $g_k(\mathbf{x})$ if that factor depends on x_i, i.e. if $i \in \mathcal{M}(k)$. The edges provide the paths along which messages are passed between nodes, designated as

$m_{x_i \to g_k}(x_i)$	from variable node x_i to factor node g_k;
$m_{g_k \to x_i}(x_i)$	from factor node g_k to variable node x_i;

and scaled so that the two evaluations sum to one: $m_{x_i \to g_k}(0) + m_{x_i \to g_k}(1) = 1$.

The update equations for the algorithm may be summarized as follows, in which superscript (j) denotes an iteration index:

- *Factor nodes*. Given incoming messages $m^{(j-1)}_{x_n \to g_k}(x_n)$ at the k^{th} factor node, perform:

$$m^{(j)}_{g_k \to x_i}(x_i) = \alpha_i \sum_{x_\ell : \ell \neq i} g_k(\mathbf{x}) \prod_{\substack{n=1 \\ n \neq i}}^{M} m^{(j-1)}_{x_n \to g_k}(x_n), \quad (8)$$

  $i = 1, 2, \ldots, M$; in which the scale factors α_i ensure that evaluations sum to one. For iteration $j = 0$, the initial

incoming messages are

$$m^{(-1)}_{x_n \to g_k}(0) = m^{(-1)}_{x_n \to g_k}(1) = \tfrac{1}{2}\,.$$

- *Variable nodes*. Given incoming messages $m^{(j)}_{g_\ell \to x_i}(x_i)$ at the i^{th} variable node, perform

$$m^{(j)}_{x_i \to g_k}(x_i) = \beta_k \prod_{\substack{\ell=1 \\ \ell \neq k}}^{K} m^{(j)}_{g_\ell \to x_i}(x_i), \tag{9}$$

$k = 1, 2, \ldots, K$; in which the scale factors β_k ensure that evaluations sum to one.

If convergence occurs, the belief quantities

$$r_i(x_i) \propto m_{x_i \to g_k}(x_i) \cdot m_{g_k \to x_i}(x_i), \qquad i = 1, 2, \ldots, M,$$

(properly scaled to sum to one) furnish the bit estimates: $\hat{x}_i = 1$ if $r_i(1) > r_i(0)$, and $\hat{x}_i = 0$ otherwise.

To rephrase this algorithm in terms of information projections, consider K copies of the variable vector $\mathbf{x}$, denoted $\mathbf{x}^1$, ..., $\mathbf{x}^K$, and treated as independent variables; we will associate each copy to one of the factors g_k, giving an extended version of our likelihood function in the form

$$p(\mathbf{x}^1, \ldots, \mathbf{x}^K) = \prod_{k=1}^{K} g_k(\mathbf{x}^k) \tag{10}$$

The true likelihood function is obtained if we constrain $\mathbf{x}^1 = \mathbf{x}^2 = \cdots = \mathbf{x}^K$. Now, the variables $\mathbf{x}^1, \ldots, \mathbf{x}^K$ taken together admit 2^{MK} evaluations. The set of probability mass functions over this set of evaluations is denoted $\mathcal{D}$. Choosing f as the negative Shannon entropy, we recall that for any probability mass function $q(\mathbf{x}) \in \mathcal{D}$, with $\mathbf{q}$ the vector collecting its 2^{MK} evaluations, the gradient $\nabla f(\mathbf{q}) = \boldsymbol{\theta}$ gives its logarithmic form, with i^{th} element $\theta_i = \log(q_i/q_0)$, and inverse map $\nabla f^*(\boldsymbol{\theta}) = \mathbf{q}$. We distinguish two sets. The first set is the set of product distributions, i.e. those distributions which are the product of their bitwise marginals

$$\mathcal{P} = \left\{ \boldsymbol{\theta} : \nabla f^*(\boldsymbol{\theta}) = q(\mathbf{x}^1, \ldots, \mathbf{x}^K) = \prod_{k=1}^{K} \prod_{i=1}^{M} q_{k,i}(x_i^k) \right\},$$
$$\hat{\mathcal{P}} = \nabla f(\mathcal{P}). \tag{11}$$

$\mathcal{P}$ is convex in the logarithmic coordinate domain $\mathcal{D}^*$, and $\hat{\mathcal{P}}$ is the corresponding (non-convex) set in the coordinate domain $\mathcal{D}$. The second set is the convex set of probability mass functions which vanish at evaluations where the K copies of $\mathbf{x}$ differ:

$$\mathcal{Q} = \left\{ r \in \mathcal{D} : r(\mathbf{x}^1, \ldots, \mathbf{x}^K) = 0 \text{ if } \mathbf{x}^i \neq \mathbf{x}^j \text{ for any } i \neq j \right\}. \tag{12}$$

These sets of probability distributions are defined in the same spirit of interpreting variable nodes in a factor graph as equality constraints as introduced in the concept of normal graphs in [45] as depicted in Figure 4. Namely, each edge in the factor graph is viewed as an independent variable, leading to the interest in $\mathcal{P}$ for which all these edge variables are independent. The variable nodes are then interpreted as equality constraints between these edges variable, leading to the interest in $\mathcal{Q}$ which considers only those probability distributions which enforce this equality with probability one. Finally, the factor functions viewed as a function of the free edge variables $\prod_{k=1}^{K} g_k(\mathbf{x}^k)$ (i.e. neglecting the edge equality constraints reflected by variable nodes), listed as a vector of probabilities yield the intialization $\mathbf{z}_{-1}$ to the projection algorithm. That is, $\mathbf{z}_{-1}$ has ith element

$$[\mathbf{z}_{-1}]_i = \prod_{k=1}^{K} g_k(\mathbf{b}_i^k) \tag{13}$$

where $\mathbf{b}_i = [\mathbf{b}_i^1, \ldots, \mathbf{b}_i^K]$ is the KM-bit binary representation of the integer i.

Theorem 1: The belief propagation algorithm is an instance of the following general Bregman projection algorithm

$$\mathbf{k}_n = \overrightarrow{\pi}^{\hat{\mathcal{P}}}_f \left(\nabla f^* \left(\nabla f(\mathbf{z}_{n-1}) + \boldsymbol{\sigma}_{n-1} \right) \right) \tag{14}$$
$$\boldsymbol{\sigma}_n = \nabla f(\mathbf{z}_{n-1}) + \boldsymbol{\sigma}_{n-1} - \nabla f(\mathbf{k}_n) \tag{15}$$
$$\mathbf{r}_n = \overleftarrow{\pi}^{\mathcal{Q}}_f (\nabla f^* (\nabla f(\mathbf{k}_n) + \boldsymbol{\tau}_{n-1})) \tag{16}$$
$$\mathbf{z}_n = \overrightarrow{\pi}^{\hat{\mathcal{P}}}_f (\mathbf{r}_n) \tag{17}$$
$$\boldsymbol{\tau}_n = \nabla f(\mathbf{k}_n) + \boldsymbol{\tau}_{n-1} - \nabla f(\mathbf{z}_n) \tag{18}$$

for integer $n \geq 0$, f the negative entropy, $\mathcal{P}, \mathcal{Q}, \hat{\mathcal{P}}$ given by (11,12), and with initialization $\boldsymbol{\sigma}_{-1} = \boldsymbol{\tau}_{-1} = \mathbf{0}$ and (13).

Before we prove the theorem, it is instructive to compare the projection algorithm generalization of belief propagation introduced in the theorem to the convex programming algorithms presented in Section II-C. We begin by noting that the algorithm bears great resemblence to Dykstra's algorithm with cyclic Bregman projections. Indeed, (15) and (18) matches (5) in Dykstra's algorithm, while the pre-projection processing in (14) and (16) matches the pre-projection processing in (4) in Dykstra's algorithm. However, there are two peculiarities which differentiate the Bregman projection generalization of belief propagation from Dykstra's algorithm with cyclic Bregman projections. First of all, the projection in (14) is a *right* projection (BP) instead of a left projection (Dykstra). Second of all while the projection in (16) is a left projection as usual it is immediately followed by a right projection (17) before (5) is calculated. This left projection followed by a right projection identifies the algorithm as bearing some resemblance with the alternating Bregman projections algorithm (6) and (7). Indeed, we observe that the projection algorithm generalization of BP appears to be a hybrid between alternating Bregman projections and Dykstra's algorithm with cyclic Bregman projections.

Proof: The algorithm presented in Theorem 1 uses notation selected to indicate the greatest affinity with Dykstra's algorithm. In the proof, it will be more notationally convenient to work with the log probability coordinates $\boldsymbol{\xi}_n = \nabla f(\mathbf{k}_n)$ and $\boldsymbol{\zeta}_n = \nabla f(\mathbf{z}_n)$. Following (1), the algorithm (14) – (18) can

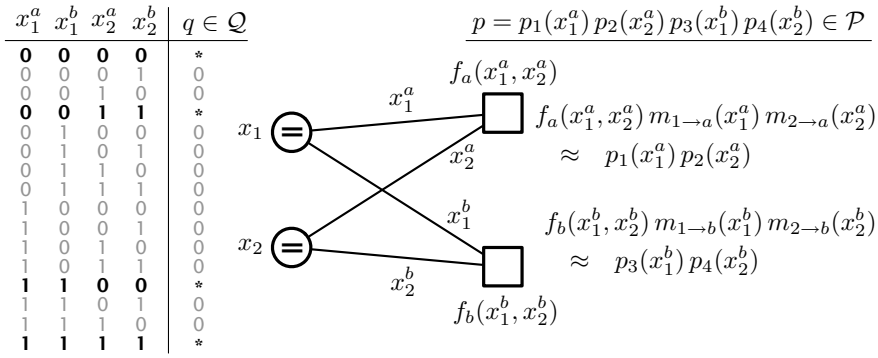

Fig. 4. The probability coordinates of two sets that belief propagation is projecting between shown for the low dimensional special case of two bits.

be rewritten as

$$\begin{aligned}
\boldsymbol{\xi}_n &= \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} D_{f^*}(\boldsymbol{\theta}, \boldsymbol{\zeta}_{n-1} + \boldsymbol{\sigma}_{n-1}) \\
&= \overleftarrow{\pi}^{\mathcal{P}}_{f^*}(\boldsymbol{\zeta}_{n-1} + \boldsymbol{\sigma}_{n-1}) \qquad (19)\\
\boldsymbol{\sigma}_n &= \nabla f(\mathbf{z}_{n-1}) + \boldsymbol{\sigma}_{n-1} - \nabla f(\mathbf{k}_n) \\
\mathbf{r}_n &= \arg\min_{\mathbf{q}\in\mathcal{Q}} D_f\Big(\mathbf{q}, \nabla f^*(\boldsymbol{\xi}_n + \boldsymbol{\tau}_{n-1})\Big) \\
&= \overleftarrow{\pi}^{\mathcal{Q}}_{f}(\nabla f^*(\boldsymbol{\xi}_n + \boldsymbol{\tau}_{n-1})) \qquad (20)\\
\boldsymbol{\zeta}_n &= \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} D_{f^*}\Big(\boldsymbol{\theta}, \nabla f(\mathbf{r}_n)\Big) = \overleftarrow{\pi}^{\mathcal{P}}_{f^*}(\nabla f(\mathbf{r}_n)) \\
\mathbf{z}_n &= \overrightarrow{\pi}^{\hat{\mathcal{P}}}_{f}(\mathbf{r}_n) \\
\boldsymbol{\tau}_n &= \nabla f(\mathbf{k}_n) + \boldsymbol{\tau}_{n-1} - \nabla f(\mathbf{z}_n)
\end{aligned}$$

Here we adopt the convention that Greek letters correspond to the log probability coordinates, while their Roman counterparts designate probability coordinates, with the exception that $\mathbf{k}$ is paired with $\boldsymbol{\xi}$. To verify the theorem, consider first the operation at the factor nodes, and introduce the marginal probability as seen from the k^{th} factor node, obtained by summing away all but the i^{th} variable x_i^k:

$$r_{k,i}^{(j)}(x_i^k) \propto \sum_{x_\ell^k : \ell \neq i} g_k(\mathbf{x}^k) \prod_{n=1}^{M} m_{x_n\to g_k}^{(j-1)}(x_n^k)$$

If we let $\boldsymbol{\xi}_j \in \mathcal{P}$ denote the logarithmic form of the product density generated by the marginals $r_{k,i}^{(j)}(x_i^k)$ (with $i \le i \le M$ and $1 \le k \le K$), then this appears as

$$\boldsymbol{\xi}_j = \arg\min_{\boldsymbol{\theta}\in\mathcal{P}} D_{f^*}(\boldsymbol{\theta}, \boldsymbol{\zeta}_{-1} + \boldsymbol{\rho}_{j-1}) \qquad (21)$$

in which $\boldsymbol{\zeta}_{-1}$ is the initial log likelihood function as in the theorem statement, and $\boldsymbol{\rho}_{j-1} \in \mathcal{P}$ is the logarithmic form of the product density formed from the (right-going) incoming messages $m_{x_i\to g_k}^{(j-1)}(x_i^k)$ at the K factor nodes. As $\nabla f^*(\cdot)$ converts back to the probability domain, this identifies

$$[\nabla f^*(\boldsymbol{\rho}_{j-1})](\mathbf{x}^1, \ldots, \mathbf{x}^K) = \prod_{k=1}^{K}\prod_{i=1}^{M} m_{x_i\to g_k}^{(j-1)}(x_i^k)$$

As $m_{x_i\to g_k}^{(-1)}(x_i^k) = \frac{1}{2}$, we have the initialization $\boldsymbol{\rho}_{-1} = \mathbf{0}$.

Comparing with (8) in which $\mathbf{x}^k$ replaces $\mathbf{x}$, we see $r_{k,i}^{(j)}(x_i^k) \propto m_{g_k\to x_i}^{(j)}(x_i^k) \cdot m_{x_i\to g_k}^{(j-1)}(x_i^k)$, so that the (left-going) return messages $m_{g_k\to x_i}^{(j)}(x_i^k)$ from factor nodes to variable nodes are obtained from the marginals by dividing out the incoming messages $m_{x_i\to g_k}^{(j)}(x_i^k)$. In the logarithmic domain, this is equivalent to the subtraction

$$\boldsymbol{\lambda}_j = \boldsymbol{\xi}_j - \boldsymbol{\rho}_{j-1}$$

in which $\boldsymbol{\lambda}_j \in \mathcal{P}$ is logarithmic form of the product distribution whose marginals are the outgoing messages $m_{g_k\to x_i}^{(j)}(x_i^k)$. As $\nabla f^*(\cdot)$ converts back to the probability domain, this identifies

$$[\nabla f^*(\boldsymbol{\lambda}_j)](\mathbf{x}^1, \ldots, \mathbf{x}^K) = \prod_{k=1}^{K}\prod_{i=1}^{M} m_{g_k\to x_i}^{(j)}(x_i^k).$$

Consider now the variable nodes, and introduce the belief quantities

$$r_i^{(j)}(x_i) \propto \prod_{k=1}^{K} m_{g_k\to x_i}^{(j)}(x_i^k)\Big|_{x_i^1 = \cdots = x_i^K = x_i}, \quad i = 1, 2, \ldots, M.$$

The evaluations on the right-hand side are found in the probability mass function $\nabla_{f^*}(\boldsymbol{\lambda}_j)$ in the M positions for which the bit copies $x_i^1, \ldots, x_i^K$ agree. Thus defining $\mathbf{r}$ as proportional to the vector containing $r_i^{(j)}(x_i)$ in positions for which $x_i^1 = \cdots = x_i^K = x_i$ and zero elsewhere, we have

$$\mathbf{r}^{(j)} = \arg\min_{\mathbf{q}\,\in\,\mathcal{Q}} D_f\Big(\mathbf{q}, \nabla f^*(\boldsymbol{\lambda}_j)\Big) \qquad (22)$$

Projecting this onto $\mathcal{P}$ gives a product distribution whose marginals are these beliefs, resulting in

$$\boldsymbol{\zeta}_j = \arg\min_{\boldsymbol{\theta}\,\in\,\mathcal{P}} D_{f^*}\Big(\boldsymbol{\theta}, \nabla f(\mathbf{r}_j)\Big)$$

Now, from (9) the return messages $m_{x_i\to g_k}^{(j)}(x_i^k)$ are the beliefs with the incoming messages divided out. In the logarithmic domain, this appears as

$$\boldsymbol{\rho}_j = \boldsymbol{\zeta}_j - \boldsymbol{\lambda}_j$$

in which $\boldsymbol{\rho}_j \in \mathcal{P}$ is the product distribution whose marginals are the (right-going) messages $m_{x_i\to g_k}^{(j)}(x_i^k)$.

To complete the proof, it suffices to show that

$$\boldsymbol{\zeta}_{-1} + \boldsymbol{\rho}_{j-1} = \boldsymbol{\zeta}_{j-1} + \boldsymbol{\sigma}_{j-1} \qquad \text{and} \qquad \boldsymbol{\lambda}_j = \boldsymbol{\xi}_j + \boldsymbol{\tau}_{j-1}$$

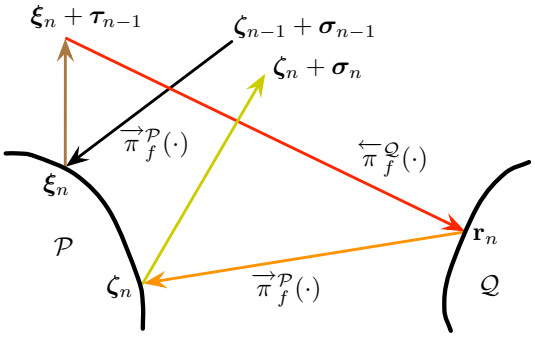

Fig. 5. The Bregman projections interpretation of belief propagation. The set $\mathcal{Q}$ is the set of expectation coordinates of probability distributions on $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k\}$ which have each of the replicas equal with probability one, and the set $\mathcal{P}$ is the set of log coordinates of product distributions. The algorithm is best described as a hybrid between an alternating Bregman projections algorithm and a Dykstra's algorithm with cyclic Bregman projections.

as this will identify (14) with (21) and (16) with (22), respectively. To this end, we can combine the definitions $\boldsymbol{\rho}_j = \boldsymbol{\zeta}_j - \boldsymbol{\lambda}_j$ and $\boldsymbol{\lambda}_j = \boldsymbol{\xi}_j - \boldsymbol{\rho}_{j-1}$ to yield

$$\begin{aligned} \boldsymbol{\rho}_j - \boldsymbol{\rho}_{j-1} &= \boldsymbol{\zeta}_j - \boldsymbol{\xi}_{j-1} \\ \boldsymbol{\lambda}_j - \boldsymbol{\lambda}_{j-1} &= \boldsymbol{\xi}_j - \boldsymbol{\zeta}_{j-1} \end{aligned}$$

Now, using $\boldsymbol{\rho}_{-1} = \mathbf{0}$, we can write the telescoping sum

$$\begin{aligned} (\boldsymbol{\rho}_{j-1} - \boldsymbol{\rho}_{-1}) + \boldsymbol{\zeta}_{-1} &= (\boldsymbol{\rho}_{j-1} - \boldsymbol{\rho}_{j-2}) + (\boldsymbol{\rho}_{j-2} - \boldsymbol{\rho}_{j-3}) \\ &\quad + \cdots + (\boldsymbol{\rho}_0 - \boldsymbol{\rho}_{-1}) + \boldsymbol{\zeta}_{-1} \\ &= \boldsymbol{\zeta}_{j-1} - \boldsymbol{\xi}_{j-2} + \boldsymbol{\zeta}_{j-2} - \boldsymbol{\xi}_{j-3} \\ &\quad + \cdots + \boldsymbol{\zeta}_0 - \boldsymbol{\xi}_{-1} + \boldsymbol{\zeta}_{-1} \\ &= \boldsymbol{\zeta}_{j-1} + \boldsymbol{\sigma}_{j-1} \end{aligned}$$

since $\boldsymbol{\sigma}_{j-1} = -\boldsymbol{\xi}_{j-2} + \boldsymbol{\zeta}_{j-2} + \boldsymbol{\sigma}_{j-2}$ from its defining recursion. This confirms the equivalence of (14) with (21).

In the same vein, since $\boldsymbol{\rho}_{-1} = \mathbf{0}$ we have $\boldsymbol{\lambda}_0 = \boldsymbol{\xi}_0 - \boldsymbol{\rho}_{-1} = \boldsymbol{\xi}_0$. By a second telescoping sum, we may thus write

$$\begin{aligned} \boldsymbol{\lambda}_j &= (\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_0) + \boldsymbol{\xi}_0 \\ &= (\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_{j-1}) + (\boldsymbol{\lambda}_{j-1} - \boldsymbol{\lambda}_{j-2}) + \cdots + (\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_0) + \boldsymbol{\xi}_0 \\ &= \boldsymbol{\xi}_j - \boldsymbol{\zeta}_{j-1} + \boldsymbol{\xi}_{j-1} - \boldsymbol{\zeta}_{j-2} + \cdots + \boldsymbol{\xi}_1 - \boldsymbol{\zeta}_0 + \boldsymbol{\xi}_0 \\ &= \boldsymbol{\xi}_j + \boldsymbol{\tau}_{j-1} \end{aligned}$$

since $\boldsymbol{\tau}_{j-1} = -\boldsymbol{\zeta}_{j-1} + \boldsymbol{\xi}_{j-1} + \boldsymbol{\tau}_{j-2}$ from its defining recursion. This confirms the equivalence between (16) and (22), to complete the proof. ■

The projection iteration shown in Theorem 1 to be equivalent belief propagation is depicted in Figure 5. To be more precise, Theorem 1 introduces a general Bregman projections algorithm, which when utilized with a particular pair of sets and a particular initialization, is equivalent to belief propagation. Since this more general Bregman projections algorithm could be employed with $\mathcal{P}$ and $\mathcal{Q}$ *arbitrary* convex sets, and f an arbitrary strictly convex function of Legendre type, and with an arbitrary initialization, we refer to it henceforth as the belief propagation Bregman projections algorithm in order to differentiate it from its belief propagation special case.

Remark:

Belief propagation is often formulated as a fast algorithm to calculate marginal probabilities from the initial likelihood function [2]. In the present formulation, these marginal probabilities are contained in $\boldsymbol{q}_0$ if obtained from the two-step projection

$$\overrightarrow{\pi}^{\hat{\mathcal{P}}}_f \left(\overleftarrow{\pi}^{\mathcal{Q}}_f (\mathbf{z}_{-1}) \right) \tag{23}$$

in which $\mathbf{z}_{-1}$ contains the initial likelihood function, as in Theorem 1. (The first projection retains only evaluations for which $\mathbf{x}^1 = \cdots = \mathbf{x}^K$, while the second marginalizes these evaluations). Note that this two-step desired projection is not calculated directly because (ironically) it presents a much higher computational burden than the belief propagation iteration, due to the fact that given *any* initial point (i.e., not only those lying in $\mathcal{P}$) it is computationally easier to project first to $\mathcal{P}$ then $\mathcal{Q}$ than to project first onto $\mathcal{Q}$ followed by $\mathcal{P}$. The belief propagation algorithm is known to converge to this desired projection when the factor graph is a tree or forest [2], and thus cycle free.

Having reformulated belief propagation as a more general projections algorithm, we now show how this offers insights into performance and convergence behavior. As remarked earlier, that existing convergence results for belief propagation apply only in an (asymptotically) acyclic setting is not a defect of belief propagation itself, as the algorithm is often observed to converge to a solution acceptably close to a true marginalization of the initial likelihood function. The best explanations at present for this behavior invoke either Bethe free energy approximations [44], or constrained likelihood approximations [46]; see [47] for the relationship between the two interpretations. An alternate approach [48] argues that, asymptotic in the number of variables M to infer, a randomly selected factor graph has arbitrarily long cycles which fail to close themselves prior to convergence. Owing to assumptions on how the factor graph is built, this result is limited to low density parity check codes and some generalizations, and fails to handle the turbo decoder, much less other applications of loopy belief propagation. These approaches still leave many open questions as to an exhaustive characterization of factorizations and models for which belief propagation converges to a vicinity of the true marginals. It is thus of interest to examine whether the convergence results of [43], [36], [41], developed in the convex optimization literature, can extend to belief propagation which inherits certain structural features as shown in Theorem 1. By considering other Bregman projections beyond information divergence in Theorem 1, a family of projection algorithms is obtained, and we show in the next section that other members of this family of algorithms are provably convergent. The framework also enables us to obtain new performance results for belief propagation in the section following it.

IV. Convergence Properties of the Belief Propagation Bregman Projections Algorithm

In this section, we investigate convergence properties of the broader family of projections algorithms suggested by Theorem 1 upon using arbitrary Bregman projections; belief propagation is then a specific instance using information

projections. We note first that by using symmetric Bregman divergences, the distinction between right and left Bregman projections is removed, thus effacing one of the two differences between the conventional belief propagation Bregman projections algorithm and Dykstra's algorithm with cyclic Bregman projections. This special property, together with the convergence literature for Dykstra's algorithm with symmetric Bregman projections, allows us to prove guaranteed convergence of the general projections algorithm when it is utilized with a symmetric Bregman divergence, as the following theorem shows.

Theorem 2 (Belief Propagation Symmetric Bregman Projections Algorithm Converges)**:** Let $\mathbf{z}_{-1}$ be any initialization and $\mathcal{P}$ and $\mathcal{Q}$ be arbitrary closed convex sets. If f induces a twice differentiable symmetric Bregman divergence, then the belief propagation Bregman projections iteration (14) – (18) converges in the sense that $\mathbf{k}_n$ and $\mathbf{z}_{n-1}$ converge to equal limits as $n \to \infty$.

Proof: Begin by recalling that the only symmetric (twice differentiable) Bregman divergences are the Mahalanobis divergences discussed in Example 2. Observe that the Mahalanobis divergences also have the property that they are shift-invariant, so that

$$D_f(\mathbf{k}, \mathbf{y}) = D_f(\mathbf{0}, \mathbf{k} - \mathbf{y}) = (\mathbf{k} - \mathbf{y})^T \mathbf{A} (\mathbf{k} - \mathbf{y})$$

Finally, note that for a Mahalanobis distance generated by $f(\mathbf{k})$ we have $\nabla f^*(\mathbf{k}) = \frac{1}{2}\mathbf{A}^{-1}\mathbf{k}$ and $\nabla f(\mathbf{k}) = 2\mathbf{A}\mathbf{k}$. The algorithm from Theorem 1 with these substitutions becomes

$$\begin{aligned}
\mathbf{k}_n &= \pi_f^{\hat{\mathcal{P}}}(\mathbf{z}_{n-1} + \mathbf{s}_{n-1}) && (24)\\
\mathbf{s}_n &= \mathbf{z}_{n-1} + \mathbf{s}_{n-1} - \mathbf{k}_n && (25)\\
\mathbf{r}_n &= \pi_f^{\mathcal{Q}}(\mathbf{k}_n + \mathbf{t}_{n-1}) && (26)\\
\mathbf{z}_n &= \pi_f^{\hat{\mathcal{P}}}(\mathbf{r}_n) && (27)\\
\mathbf{t}_n &= \mathbf{k}_n + \mathbf{t}_{n-1} - \mathbf{z}_n && (28)
\end{aligned}$$

We show first that $\mathbf{k}_n - \mathbf{z}_{n-1} \to 0$ as $n \to \infty$. As $\mathbf{k}_n$ is the projection of $\mathbf{z}_{n-1} + \mathbf{s}_{n-1}$ onto $\hat{\mathcal{P}}$, we have

$$D_f(\mathbf{z}_{n-1} + \mathbf{s}_{n-1}, \mathbf{r}) \geq D_f(\mathbf{z}_{n-1} + \mathbf{s}_{n-1}, \boldsymbol{x}_n) + D_f(\mathbf{k}_n, \mathbf{r}),$$

for all $\mathbf{r} \in \mathcal{P}$. This inequality thus applies to the particular choice $\mathbf{r} = \mathbf{z}_{n-1} \in \mathcal{P}$, which gives after applying shift invariance

$$D_f(0, \mathbf{s}_{n-1}) \geq D_f(0, \mathbf{s}_n) + D_f(0, \mathbf{k}_n - \mathbf{z}_{n-1}).$$

Iterating this inequality and summing, we thus have

$$D_f(0, \mathbf{s}_1) \geq D_f(0, \mathbf{s}_{N+1}) + \sum_{n=1}^{N} D_f(0, \mathbf{k}_n - \mathbf{z}_{n-1})$$

As $D_f(0, \mathbf{s}_1)$ is bounded, this shows that $D_f(0, \mathbf{k}_n - \mathbf{z}_{n-1}) \to 0$ as $n \to \infty$. Finally, since $D_f(0, \mathbf{k}_n - \mathbf{z}_{n-1}) \leq \lambda_{\text{MAX}}(\mathbf{A}) \|\mathbf{k}_n - \mathbf{z}_{n-1}\|^2$, with $\lambda_{\text{MAX}}(\mathbf{A})$ the maximum eigenvalue of $\mathbf{A}$, we have that $\|\mathbf{k}_n - \mathbf{z}_{n-1}\| \to 0$ as $n \to \infty$.

Next, we note that $\mathbf{k}_n - \mathbf{z}_{n-1} = \mathbf{k}_n + \mathbf{t}_{n-1} - (\mathbf{k}_{n-1} + \mathbf{t}_{n-2})$. As $\mathbf{k}_n - \mathbf{z}_{n-1} \to 0$, we thus have $(\mathbf{k}_n + \mathbf{t}_{n-1}) - (\mathbf{k}_{n-1} + \mathbf{t}_{n-2}) \to 0$ as well. By continuity of the projector $\pi_f^{\mathcal{Q}}(\cdot)$, this implies that $\mathbf{r}_n = \pi_f^{\mathcal{Q}}(\mathbf{k}_n + \mathbf{t}_{n-1})$ converges to a limit. Appealing again to continuity of the projector $\pi_f^{\mathcal{P}}(\cdot)$, this implies that $\mathbf{z}_n = \pi_f^{\mathcal{P}}(\mathbf{r}_n)$ likewise converges to a limit, and therefore that $\mathbf{k}_n$ converges to this same limit. ■

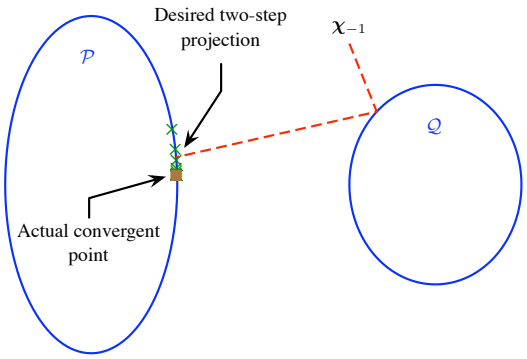

Fig. 6. Illustrating cyclic projection algorithm in the Euclidean case. Dashed line shows two-step projection, while crosses ("×") indicate successive projections $\boldsymbol{\xi}_n$.

Returning our focus to the belief propagation case, for which f is the negative entropy and D_f is not symmetric, we observe the implication of Theorem 2 is that the lack of guaranteed convergence of belief propagation (i.e., its occasional pathological dynamics) stems directly and exclusively from the asymmetry of the divergence it utilizes. Furthermore, this result *suggests* that the frequently observed *good* dynamics behavior of belief propagation stems from its similarity of form to this provably convergent symmetric-divergence variant.

Finally, we note that convergence of an algorithm is only half of the picture. Namely, even when convergence has been proven, it is important to prove performance properties of the convergent points. In this spirit, we observe in Figure 6 that while the belief propagation Bregman projections algorithm converges when utilized with a symmetric divergence, the convergent point is only in the vicinity of (but not generally equal to) the symmetric divergence analog of the "desired" projection $\pi_f^{\mathcal{P}}\pi_f^{\mathcal{Q}}$. Furthermore, even if $\mathcal{P}$ and $\mathcal{Q}$ are taken to be the sets (11)-(12) defined in regular belief propagation the orthogonal (or even Mahalanobis) projections, $\pi_f^{\mathcal{P}}\pi_f^{\mathcal{Q}}$ will have little to do with the marginalization of the initial density. From this fact we observe that while the symmetric divergence variant of the belief propagation Bregman projections algorithm is useful in identifying the root of convergence (mis)behavior of belief propagation, it does not constitute an alternative approximate marginalization algorithm. For this reason, the next section sets about using the new framework to prove when (unmodified) belief propagation is guaranteed to give good approximations of the marginal probabilities.

V. Performance of Belief Propagation for Cyclic Factorizations

In this section we prove that belief propagation yields estimated marginals close to the marginals of the joint distribution for a class of factorizations other than just trees and forests, which constitute acyclic graphs. A popular approach [48] for justifying belief propagation in certain cyclic factor

graphs argues that asymptotic in the number of variables to infer, a randomly selected homogeneous factor graph with a given pair of degree distributions has arbitrarily large girth, such that cycles fail to "close" themselves before a finite number of iterations. Here we consider an alternate route which instead characterizes factorizations that are close to acyclic in a manner independent of factor graph girth. We then show Lipschitz continuity of the projectors involved in belief propagation, to establish that, for all factorizations close to acyclic, the belief propagation algorithm furnishes beliefs that are close to the true marginals. The approach so taken has the virtue of avoiding assumptions of a large factor graph girth or other asymptotic approximations.

The intuition behind the result, together with its power over approaches based only on the girth of the factor graph, is straightforward once the context has been set up. In particular, graph based arguments can only talk about factorizations that are "close to acyclic" in terms of the graph structure of the factor graph, i.e., its girth. This definition is effectively blind to the functions that comprise the factors themselves: it considers only the variables they depend on. On the other hand, common sense dictates that there must be other ways for the factorizations to be close to cycle free: for instance an offending factor node involved in a loop may by extremely weakly dependent on the implicated variable – this ought not to affect the dynamics of BP too much after a bounded number of iterations and thus should also yield answers close to the desired marginalization. An information geometric formulation opens up a mathematical route for this intuition: a new definition for factorizations that are "close to cycle free" in terms of having log coordinates which are close to the set of log coordinates of cycle free factorizations. A summary of the proof that belief propagation run on such a factorization yields estimates closed to the true marginals is diagrammed in Figure 7, and proceeds as follows:

- Consider an initialization (i.e., a factorization) $\boldsymbol{\zeta}_{-1}$ for BP whose log coordinates are close the set of log coordinates of acyclic factorizations. The nearest point in the set of acyclic factorizations (call it $\boldsymbol{\zeta}_{-1}^F$) is an initialization for which BP converges to the marginals of that intialization after a finite number of iterations L.
- Since $\boldsymbol{\zeta}_{-1}$ and $\boldsymbol{\zeta}_{-1}^F$ are close to one another, their marginals must, by continuity of the desired (marginalization) projection, be close to one another.
- Because belief propagation is continuous in its initialization after a finite number of iterations, after an appropriate number of iterations L belief propagation for the cyclic (but close to acyclic) initialization $\boldsymbol{\zeta}_L$ must be close to the result of belief propagation for the nearby acyclic initialization $\boldsymbol{\zeta}_L^F$. But since belief propagation gives the desired projection after L iterations for the acyclic initialization $\boldsymbol{\zeta}_{-1}^F$, $\boldsymbol{\zeta}_L^F$ is just the marginals of $\boldsymbol{\zeta}_{-1}^F$. This then shows that $\boldsymbol{\zeta}_L$ is close to the marginals of $\boldsymbol{\zeta}_{-1}^F$. But putting this together with the previous step says that it must be close to its own marginals (by adding the two distances which are both small via continuity of two different projections operators).

This gives a new way to get bounded convergence from marginals for factor graphs, and proves insensitive to graph topology by exploiting knowledge of the factor functions.

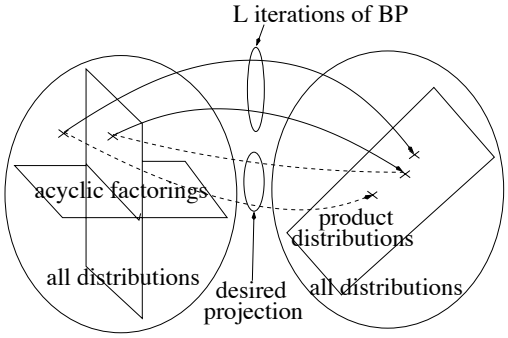

Fig. 7. The technique we use to prove a new region of factorizations for which belief propagation gives estimates near its marginals after a finite number of iterations.

The remainder of this section assembles and proves these ideas formally. In particular sections V-A and V-B determine the set of coordinates factorizations which are effectively acyclic, while section V-C then proves continuity of the projectors and assembles the bound relating iterates of belief propagation to a distribution's marginals.

A. Log Coordinates of Acyclic Factorizations

In order to express our argument in precise mathematical terms, it is first necessary to characterize the set of factorizations, and thus initializations $\boldsymbol{\zeta}_{-1}$, that are derived from factor graphs that are acyclic. Of course, the form of the factorization (10), in which the factors are treated as independent because their arguments are no longer constrained to arise from the same vector, yields log coordinates $\boldsymbol{\theta}$ on the 2^{KM} possible outcomes $\bar{\mathbf{x}} = [\mathbf{x}^1, \ldots, \mathbf{x}^K]$ that are a linear function of the log coordinates $\boldsymbol{\theta}_k$ for the factors g_k:

$$\boldsymbol{\theta} = \sum_{k=1}^{K} \mathbf{P}_k \boldsymbol{\theta}_k$$

Here the matrix $\mathbf{P}_k \in \{0,1\}^{(2^{MK}-1)\times(2^M-1)}$ has i,jth element one if the portion of the ith possible realization of $\bar{\mathbf{x}}$ associated with $\mathbf{x}^k$ is equal to the jth possible realization of $\mathbf{x}^k$, and zero otherwise. Thus, in order to characterize the set of log coordinates of acyclic factorizations, it is sufficient to characterize the log coordinates $\boldsymbol{\theta}_k$ of each of the K factors.

The key feature of a factor graph is the independence of each factor on some subset of the variables. Thus, of primary interest when translating requirements on the factor graph to requirements of the log coordinates of the factors (treated now as functions of all of the variables) is how to check if the log coordinates correspond to independence on a given variable. Now, independence of a given factor g_k on a variable means that the factor must have (when treated as a function of all of the variables) the same value for all realizations of that variable when all of the other variables are held fixed. As such, we can test for independence of factor g_k on the variable x_i by seeing if it lies in the null space of the binary $2^{M-1} \times 2^M - 1$

dimensional matrix $\mathbf{N}_i$ whose $(j,l)th$ element is

$$[\mathbf{N}_i]_{j,l} = \begin{cases} 1 & b_{l,i} = 1, \mathbf{b}_l \setminus b_{l,i} = \mathbf{w}(j) \\ -1 & b_{l,i} = 0, \ \mathbf{b}_l \setminus b_{l,i} = \mathbf{w}(j) \\ 0 & \text{otherwise} \end{cases}$$

for $l \in \{1, \ldots, 2^M - 1\}$ and $j \in \{1, \ldots, M\}$, where $\mathbf{w}(j)$ is the $N-1$ bit binary representation of the integer j. Stacking all such possible matrices $\mathbf{N}_i$ on top of each other to get $\mathbf{N} = [\mathbf{N}_1^T, \mathbf{N}_2^T, \ldots, \mathbf{N}_M^T]^T$, we can then translate from a column vector $\mathbf{a}_k$ whose ith element indicates whether or not factor g_k depends on a variable x_i, into a matrix $\mathbf{G}_k$ which its corresponding log coordinates must lie in (i.e. $\mathbf{G}_k \boldsymbol{\theta}_k = \mathbf{0}$), through the relation

$$\mathbf{G}_k = \left[(\mathbf{1}_{M \times 1} - \mathbf{a}_k) \otimes \mathbf{1}_{2^{M-1} \times (2^M - 1)} \right] \odot \mathbf{N}$$

where $\otimes$ is the Kronecker product and $\odot$ is the entrywise Hadamard product. This then shows that a particular forest factor graph corresponds to a subspace (the null space of the associated $\mathbf{G}_k$ matrices) in which the log coordinates for the factors must lie.

We recall next that the adjacency matrix of a bipartite factor graph assumes the form $\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}$ with $\mathbf{A} \in \{0,1\}^{M \times K}$; the k-th column of $\mathbf{A}$ is the vector $\mathbf{a}_k$ that intervenes in the formulation of $\mathbf{G}_k$. To describe the set of adjacency matrices generated from acyclic graphs, we recall the following basic facts from graph theory [49], [50]:

- A forest (i.e., an acyclic graph) is, by definition, the union of a collection of disjoint trees;
- A tree is, by definition, a connected graph with one more node than edge;
- A graph is connected if and only if the maximum eigenvalue of its adjacency matrix is simple [49, p. 3];
- Half the trace of the square of the adjacency matrix is the number of edges in the graph [50, p. 35].

Thus, we can algebraically describe the set of all adjacency matrices of acyclic factor graphs with M variable nodes and K factor nodes, as the set of matrices $\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}$ with $\mathbf{A} \in \{0,1\}^{M \times K}$ such that

$$\mathbf{A} = \mathbf{J} \, \text{BlockDiag}(\mathbf{A}_1, \ldots, \mathbf{A}_C) \tag{29}$$

where $\mathbf{J}$ is a permutation matrix, $\text{BlockDiag}(\cdot)$ returns a block diagonal matrix whose block diagonal elements are its matrix arguments, and where $\mathbf{A}_c$ is of dimension $M_c \times K_c$ such that

$$\sum_{c=1}^{C} M_c = M, \quad \sum_{c=1}^{C} K_c = K, \lambda_{\max} \begin{bmatrix} \mathbf{0} & \mathbf{A}_c \\ \mathbf{A}_c^T & \mathbf{0} \end{bmatrix} \text{ simple}$$

$$M_c + K_c - 1 = \frac{1}{2} \, \mathsf{tr} \left(\begin{bmatrix} \mathbf{0} & \mathbf{A}_c \\ \mathbf{A}_c^T & \mathbf{0} \end{bmatrix}^2 \right) \tag{30}$$

Here tr and $\lambda_{\max}$ denote the trace and maximum eigenvalue of their matrix argument. To summarize, then, we have shown that the log coordinates $[\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K]$ of all acyclic factorizations may be written as the union over adjacency matrices given by (29) of vector subspaces which lie the null space of the associated matrices $\text{diag}\,[\mathbf{G}_1, \ldots, \mathbf{G}_k]$.

B. Log Coordinates of Effectively Acyclic Factors

Although the previous discussion checks for all acyclic factorizations with K factor nodes and M variable nodes, it cannot detect those factorizations for which a single factor node $g_k(\mathbf{x}^k)$ contains several collections of different variables which are independent under g_k, which thus factors as $g_k(\mathbf{x}^k) = g_{k,1}(\mathbf{x}_1^k) \, g_{k,2}(\mathbf{x}_2^k)$ with $\mathbf{x}_1^k$ and $\mathbf{x}_2^k$ disjoint subsets of $\mathbf{x}^k$. Hence, it is important to describe the log coordinates of *effectively acyclic factorizations* which, via splitting factor nodes into independent parts, may be transformed into an acyclic factor graph. Such effectively acyclic factorizations, too, will benefit from proven convergence of belief propagation to the true marginals. This happens because a factor node $g_k(\mathbf{x}^k)$ which factors into $g_k(\mathbf{x}^k) = g_{k,1}(\mathbf{x}_1^k) g_{k,2}(\mathbf{x}_2^k)$, with $\mathbf{x}_1^k$ and $\mathbf{x}_2^k$ nonintersecting subsets of $\mathbf{x}^k$, calculates messages in the same manner as two separate factor nodes $g_{k,1}(\mathbf{x}_1^k)$ and $g_{k,2}(\mathbf{x}_2^k)$.

We begin with the log coordinates of K factors involving a total of M variables. In determining whether or not the log coordinates lie in the set of log coordinates of effectively acyclic factorizations, we wish to determine if the K factors can be broken up into K' new factors for which the new factor graph is acyclic. Equivalently, we seek K positive integers $J_k, k \in \{1, \ldots, K\}$ such that

$$K' = \sum_{k=1}^{K} J_k$$

with each factor g_k associated with J_k new factors, $g_{k,1}, \ldots, g_{k,J_k}$. The vector indicating which of the M variables $g_{k,j}$ depends on is $\mathbf{a}_{k,j} \in \{0,1\}^{M \times 1}$, and because the factors are to be independent we must have

$$\mathbf{a}_{k,j}^T \, \mathbf{a}_{k,j'} = 0, \quad \forall j \neq j' \in \{1, \ldots, J_k\} \tag{31}$$

Furthermore, because

$$g_k(\mathbf{x}^k) = \prod_{j=1}^{J_k} g_{k,j}(\mathbf{x}^k)$$

the log coordinates of g_k, denoted again by $\boldsymbol{\theta}_k$, must be the sum of the log coordinates of the $g_{k,j}$s, denoted by $\boldsymbol{\theta}_{k,j}$:

$$\boldsymbol{\theta}_k = \sum_{j=1}^{J_k} \boldsymbol{\theta}_{k,j}$$

Also, we know that the independence requirements on $\boldsymbol{\theta}_{k,j}$ dictate that it lies in the null space of the matrix

$$\mathbf{G}_{k,j} = \left[(\mathbf{1}_{M \times 1} - \mathbf{a}_{k,j}) \otimes \mathbf{1}_{2^{M-1} \times (2^M - 1)} \right] \odot \mathbf{N}$$

Denoting a basis for this null space by $\mathbf{F}_{k,j}$, we see that we can detect such a refactorization of a given factor node by checking to see if it lies in the span of the columns of the matrix

$$\mathbf{F}_k := [\mathbf{F}_{k,1}, \mathbf{F}_{k,2}, \ldots, \mathbf{F}_{k,J_k}]$$

Using ideas from the previous section, such a refactorization will correspond to a forest if the new adjacency matrix

$$\mathbf{A} := [\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \ldots, \mathbf{a}_{1,J_1}, \ldots, \mathbf{a}_{K,1}, \ldots, \mathbf{a}_{K,J_K}]$$

with dimensions $M \times K'$ can be represented as

$$\mathbf{A} = \mathbf{J}\,\text{BlockDiag}(\mathbf{A}_1, \ldots, \mathbf{A}_C)$$

where $\mathbf{A}_c$ is of dimension $M_c \times K_c$ such that the conditions in (30) hold. This is the same definition as the set of forest factorizations with dimensions $M \times K'$ from the previous section, with the new requirement, however, that (31) holds. Thus, the set of log coordinates of effectively acyclic factorizations is also a union (over $\mathbf{A}$ matrices) of vector subspaces which we have explicitly described here. We will refer to this set as $\mathcal{F}$ from here on.

C. A New Region of Good Behavior for Belief Propagation

We now use the characterization of the log coordinates of all acyclic factorizations for a particular problem dimension (M, K) to derive a new collection of *cyclic* factorizations for which belief propagation can be proven to exhibit good behavior. To do so, we will need the following basic proposition.

Prop. 1: The vector of messages $\boldsymbol{\zeta}_L$ passed by belief propagation after L iterations is uniformly Lipschitz continuous in the initialization $\boldsymbol{\zeta}_{-1}$ with Lipschitz constant C_L, and the desired marginal projection $\nabla f\left(\overrightarrow{\pi}_f^{\hat{\mathcal{P}}}\left(\overleftarrow{\pi}_f^{\mathcal{Q}}\left(\nabla f^*(\boldsymbol{\zeta}_{-1})\right)\right)\right)$ from (23) is also uniformly Lipschitz continuous in $\boldsymbol{\zeta}_{-1}$ with Lipschitz constant γ.

The proof of this proposition is provided in appendix A, along with bounds on the Lipschitz constants. This proposition allows us to prove the following theorem.

Theorem 3: The set of factorization initializations $\boldsymbol{\zeta}_{-1}$ for which belief propagation provides estimates within Euclidean distance ϵ of their true marginals after a finite number of iterations includes the union over all points $\boldsymbol{\zeta}_{-1}^F$ in the set of log coordinates of effectively acyclic factorizations of balls of initializations $\boldsymbol{\zeta}_{-1}$ no further than $\frac{\epsilon}{C_L+\gamma}$ in Euclidean distance from the set of log coordinates of effectively acyclic factorizations $\mathcal{F}$ described in the previous section, where the tree width of the effectively acyclic factorization $\boldsymbol{\zeta}_{-1}^F$ is denoted by L.

The formal proof is provided in appendix B. This establishes a new region of good behavior for belief propagation, created by "fattening" the set of effectively acyclic factorizations from the union of a finite collection of hyperplanes, which has zero Lebesgue measure, to a set of larger Lebesgue measure that is a collection of balls centered at points in these hyperplanes. The result is expected to have practical utility in applications of belief propagation to graphs of small or moderate girth, such as those associated with turbo codes, but nevertheless have been empirically observed to provide estimates with performance near their exact marginals after a finite number of iterations.

VI. Extension to Expectation Propagation

Although belief propagation has proved widely applicable, its extension to continuous random variables can encounter various difficulties, save for the Gaussian case. Expectation propagation [51], [52], [37] represents an extension of the message-passing iterative behavior of belief propagation to random variables described by exponential families of densities [53], [54], and indeed reduces to belief propagation when applied to Gaussian densities.

The algorithm description shares many affinities with belief propagation, except that the vector $\mathbf{x}$ now contains continuous variables to infer. It begins again with a factored likelihood function

$$g(\mathbf{x}) = \prod_{k=1}^{K} g_k(\mathbf{x})$$

in which each factor on the right-hand side will depend usually only on a subset of the variables contained in $\mathbf{x}$. Expectation propagation attempts to approximate the likelihood function $g(\mathbf{x})$ as the product of exponential family densities, according to

$$g(\mathbf{x}) \approx \prod_{k=1}^{K} h_{\lambda_k}(\mathbf{x})$$

in which each factor $h_{\lambda_k}(\mathbf{x})$ is an exponential density [54], [53] of the form

$$h_{\lambda_k}(\mathbf{x}) = \exp[\boldsymbol{\lambda}_k^T \mathbf{t}_k(\mathbf{x}) - \psi(\boldsymbol{\lambda}_k)], \qquad \text{with}$$
$$\psi(\boldsymbol{\lambda}_k) = \log\left(\int \exp[\boldsymbol{\lambda}_k^T \mathbf{t}_k(\mathbf{x})]\, d\mathbf{x}\right).$$

Here the vector $\mathbf{t}_k(\mathbf{x})$ contains the sufficient statistics, which determine the type of exponential family random variable to be used (Gaussian, beta, chi-squared, Bernoulli, Poisson, etc.), while the the parameters $\boldsymbol{\lambda}_k$ select a particular distribution (taking on the role of the mean and variance, for instance, in the case of Gaussian random variables). The factors h_{λ_k} are each refined according to the minimization problem

$$h_{\lambda_i}(\mathbf{x}) = \arg\min_{\boldsymbol{\lambda}_i} D_f(v_i, q)$$

in which D_f is the continuous analogue of the Kullback-Leibler divergence [40]:

$$D_f(v_i, q) = \int v_i(\mathbf{x}) \log \frac{v_i(\mathbf{x})}{q(\mathbf{x})}\, d\mathbf{x}$$

This is simply the Bregman divergence induced by choosing f as negative differential entropy:

$$f(q) = \int q(\mathbf{x}) \log[q(\mathbf{x})]\, d\mathbf{x}$$

For expectation propagation, the arguments to the Kullback-Leibler divergence assume the functional forms

$$\begin{aligned} v_i(\mathbf{x}) &= \alpha\, g_i(\mathbf{x}) \prod_{k \neq i} h_{\lambda_k}(\mathbf{x}) \\ q(\mathbf{x}) &= \beta \prod_{k} h_{\lambda_k}(\mathbf{x}) \end{aligned}$$

where the constants α and β ensure that the densities integrate to one. The solution for $h_{\lambda_i}(\mathbf{x})$ is again an information projection, and is characterized by the expectation equality

$$\int \mathbf{t}_i(\mathbf{x})\, q(\mathbf{x})\, d\mathbf{x} = \int \mathbf{t}_i(\mathbf{x})\, v_i(\mathbf{x})\, d\mathbf{x}.$$

Repeating this projection in succession for each factor gives rise to a message passing algorithm [52], [51], [37] analogous to that for belief propagation.

As with the belief propagation case, consider K copies of the continuous variables $\mathbf{x}$, denoted as $\mathbf{x}^1, \ldots, \mathbf{x}^K$, with extended likelihood function

$$g(\mathbf{x}^1, \ldots, \mathbf{x}^K) = \prod_{k=1}^{K} g_k(\mathbf{x}^k)$$

so that the original likelihood function is obtained by constraining $\mathbf{x}^1 = \mathbf{x}^2 = \cdots = \mathbf{x}^K$.

Let $\mathcal{D}$ now denote the set of integrable (with respect to Lebesgue measure) probability density functions. We denote by $\mathcal{Q}$ the convex set of probability density functions that vanish almost surely whenever the K copies differ:

$$\mathcal{Q} = \left\{ q \in \mathcal{D} : q(\mathbf{x}^1, \ldots, \mathbf{x}^M) = 0 \text{ if } \mathbf{x}^i \neq \mathbf{x}^j \text{ for any } i \neq j \right\}$$

Similarly, we denote by $\mathcal{P} \in \mathcal{D}^*$ the convex subset dual to the set of exponential product distributions:

$$\mathcal{P} = \left\{ \boldsymbol{\theta} \in \mathcal{D}^* : q(\mathbf{x}^1, \ldots, \mathbf{x}^M) = \exp[\boldsymbol{\lambda}^T \hat{\mathbf{t}}(\mathbf{x}) - \psi(\boldsymbol{\lambda})] \right\}$$

with separable $\hat{\mathbf{t}}(\mathbf{x}) = [\mathbf{t}_1(\mathbf{x}^1), \ldots, \mathbf{t}_K(\mathbf{x}^K)]$.

Theorem 4: With these redefinitions, the expectation propagation algorithm admits the same cyclic projection description as in Theorem 1.

A detailed proof is developed in [16]. The relevance is that the information geometric properties of the projection algorithm of Theorem 1 extend to a much broader family of estimation algorithms, since expectation propagation can handle more complex dependence structures than belief propagation and approximations of continuous random variables through explicit use of exponential families.

VII. Conclusions and Extensions

The information geometric framework pursued here establishes belief propagation as a hybrid between Dykstra's algorithm with cyclic Bregman projections and alternating Bregman projections. The key feature of our framework, when contrasted with previous information geometric developments of belief propagation, is that extrinsic information extraction is accommodated within the context of the projection algorithm itself, thanks to the connection with Dykstra's algorithm. While this avoids the "projection on moving sets" pitfall that has encumbered earlier approaches, it more importantly allows us to show that the projection algorithm underlying belief propagation is convergent whenever symmetric Bregman divergences are substituted; this confirms that the convergence problems with belief propagation are to be attributed to asymmetry of the KL divergence.

In practice, of course, the ubiquitous belief propagation algorithm cannot be algorithmically removed from its information projection origins, and thus conditions under which the standard algorithm is guaranteed to behave well remain of interest. The projection approach pursued here, accordingly, reveals a new region of factorizations, independent of factor graph girth, for which belief propagation provides estimates close to the true marginals after a finite number of iterations. Finally, we showed that the projections algorithm interpretation is easily extended to expectation propagation. Future work should explore providing convergence conditions for unmodified belief propagation by bounding its difference from its "symmetrized" version. Additionally, future work should take popular applications of the belief propagation decoder, such as to turbo codes with intermediate block lengths, and verify if the associated factorization lies with high probability within the set of good behavior exposed here.

VIII. Acknowledgements

The authors would like to thank Pierre Duhamel and Florence Alberge for several helpful discussions.

Appendix A
Proof of Uniform Lipschitz Continuity

A. Lipschitz Continuity of Desired Marginal Projection and Associated Lipschitz Constant

In this section we show that the log coordinates of the desired marginal projection $\nabla f \left(\overrightarrow{\pi}_f^{\hat{\mathcal{P}}} \left(\overleftarrow{\pi}_f^{\mathcal{Q}} (\nabla f^*(\boldsymbol{\zeta}_{-1})) \right) \right)$ are Lipschitz continuous in their argument $\boldsymbol{\zeta}_{-1}$, providing a bound for the Lipschitz constant for the 2-norm $\|\cdot\|_2$. Begin by noting that if we let $\boldsymbol{\theta}_a$ represent a 2^M dimensional vector with 0 in its first position and the remaining $2^M - 1$ positions as the $2^M - 1$ dimensional log coordinates of the factor $g_k(\mathbf{x})$, then the result of the desired projection can be written as

$$\mathbf{P} \left[(\mathbf{B}\lambda)^T, (\mathbf{B}\lambda)^T, \ldots, (\mathbf{B}\lambda)^T \right]^T, \quad \text{where} \tag{32}$$
$$\lambda = \log \left(\mathbf{B}^T \exp(\sum_a \boldsymbol{\theta}_a) \right) - \log \left((\mathbf{1} - \mathbf{B})^T \exp(\sum_a \boldsymbol{\theta}_a) \right)$$

Furthermore, if we represent the result in terms of the K log coordinates of each of the subvectors $\mathbf{x}^k$, then the function is equal to $\left[(\mathbf{B}\lambda)^T, (\mathbf{B}\lambda)^T, , \ldots, (\mathbf{B}\lambda)^T \right]^T$. This shows that it suffices to show Lipschitz continuity of $\mathbf{B}\lambda$ with λ given by (32). since the Lipschitz constant for $\left[(\mathbf{B}\lambda)^T, (\mathbf{B}\lambda)^T, , \ldots, (\mathbf{B}\lambda)^T \right]^T$ is just K times it. The Jacobian matrix of the function in the square brackets in (32) with respect to *any one* of the k factor's log coordinates $\boldsymbol{\theta}_k$ has (i, j)th element which may be interpreted in terms of conditional probabilities from the probability mass function whose log coordinates are $\boldsymbol{\theta} = \sum_a \boldsymbol{\theta}_a$, treated as a probability on a random binary N vector $\mathbf{x}$ with ith element x_i as

$$[\mathbf{H}]_{i,j} := \begin{cases} \Pr[\mathbf{x} = \mathbf{b}_j | x_i = 1] & [\mathbf{b}_j]_1 = 1 \\ -\Pr[\mathbf{x} = \mathbf{b}_j | x_i = 0] & [\mathbf{b}_j]_1 = 0 \end{cases} \tag{33}$$

From this, then, we can write the Jacobian matrix for $\mathbf{B}\lambda$ given by (32) regarded as a function of *all* K factors as $[\mathbf{BH}, \mathbf{BH}, \ldots, \mathbf{BH}]$, where there are exactly K copies of $\mathbf{BH}$. Because the desired projection map forms K copies of this result (one for each copy $\mathbf{x}^k$ of the original vector of variables) the Jacobian matrix for the overall desired projection map is then formed by stacking K copies of the Jacobian matrix for (32). The uniform Lipschitz constant for the 2-norm $\|\cdot\|_2$ is then given by square root of the maximum

spectral radius of this Jacobian matrix over all points in the log coordinates space. Noting that by definition (33) the rows in the matrix $\mathbf{H}$ must have elements in $[-1,1]$ which sum over the entire row to 0, and over the positions for which the ith bit is one to 1, and over the positions for which the ith bit is zero to -1. Letting these be the only constraints (which can only increase the maximum, thus still yielding an appropriate Lipschitz constant), we then choose the Lipschitz constant to be the solution to the quadratic optimization problem

$$\gamma = \sqrt{K^3 \max_{\mathbf{H}\in\mathcal{H}} \rho_{max}\left(\mathbf{H}^T\mathbf{B}^T\mathbf{B}\mathbf{H}\right)}$$

$$\mathcal{H} := \left\{\mathbf{H} | \mathbf{h}_i^T\mathbf{b}^i = 1, \mathbf{h}_i^T(\mathbf{1}-\mathbf{b}^i) = -1, -\mathbf{1} \le \mathbf{h}_i \le \mathbf{1}, \forall i\right\}$$

where ρ_{max} is the maximal eigenvalue of the matrix argument. Here the factor K^3 comes from the fact that the Jacobian matrix was a block matrix with K^2 copies of the same matrix $\mathbf{BH}$ as its block elements. Next, noting that $\mathbf{B}^T\mathbf{B} = 2^{M-2}(\mathbf{I}_M + \mathbf{1}_M\mathbf{1}_M^T)$, we can write the Lipschitz constant as

$$\gamma = \sqrt{K^3 2^{M-2} \max_{\mathbf{H}\in\mathcal{H}} \max_{\mathbf{v}\mid\|\mathbf{v}\|_2=1} \mathbf{v}^T\left(\mathbf{H}^T\mathbf{H} + \mathbf{H}^T\mathbf{1}_M\mathbf{1}_M^T\mathbf{H}\right)\mathbf{v}}$$

If we set $\mathbf{w} = \mathbf{H}\mathbf{v}$, then

$$\mathbf{v}^T\left(\mathbf{H}^T\mathbf{H} + \mathbf{H}^T\mathbf{1}_M\mathbf{1}_M^T\mathbf{H}\right)\mathbf{v} = \sum_{i=1}^M w_i^2 + \left(\sum_{i=1}^M w_i\right)^2.$$

As $\mathbf{H}$ is composed of conditional probabilities, it is easily shown that $-1 \le w_i \le 1$ for each i, and that a "maximizing" configuration of $w_i = 1$ for each i (respectively, $w_i = -1$ for each i) can be attained for certain choices of $\mathbf{H}$. The conditional probability would put all its mass on $\mathbf{b}_{2^M-1}$ or $\mathbf{b}_0$, respectively, and $\mathbf{v}$ would be chosen as the last or first unit vector. For such a configuration,

$$\rho_{\max} = \sum_{i=1}^M w_i^2 + \left(\sum_{i=1}^M w_i\right)^2 = M + M^2$$

As this represents the largest attainable $\rho_{\max}$, we thus have

$$\gamma \le \sqrt{K^3 2^{M-2}(M+M^2)}.$$

B. Lipschitz Constant For Lth BP Iteration

In this section we consider the Lipschitz constant which reflects a bound on the change in ζ_L due to a change in the initialization $\boldsymbol{\zeta}_{-1}$. This is easily related to the calculation in the previous section of a Lipschitz constant for the desired marginalization projection by rewriting the belief propagation iteration in the new variables $\boldsymbol{\alpha}_n = \boldsymbol{\zeta}_n + \boldsymbol{\sigma}_n - \boldsymbol{\zeta}_{-1}$ for $n \ge -1$, and $\boldsymbol{\beta}_n = \boldsymbol{\xi}_n + \boldsymbol{\tau}_{n-1}$ as

$$\begin{aligned}\boldsymbol{\beta}_n &= \overleftarrow{\pi}^{\mathcal{P}}_{f^*}(\boldsymbol{\zeta}_{-1} + \boldsymbol{\alpha}_{n-1}) - \boldsymbol{\alpha}_{n-1}\\ \boldsymbol{\alpha}_n &= \overleftarrow{\pi}^{\mathcal{P}}_{f^*}\left(\nabla f\left(\overleftarrow{\pi}^{\mathcal{Q}}_{f}\left(\nabla f^*(\boldsymbol{\beta}_n)\right)\right)\right) - \boldsymbol{\beta}_n\end{aligned}$$

Recognizing $\boldsymbol{\zeta}_n = \boldsymbol{\alpha}_n + \boldsymbol{\beta}_n$ for $n \ge 0$, and recalling the triangle inequality for norms, it is clear that we may add the Lipschitz constant for $\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_n$ from $\boldsymbol{\zeta}_{-1}$ to get a bound on the Lipschitz constant for $\boldsymbol{\zeta}_n$. Furthermore, the second equation (34) is easily recognized as directly related to the desired projection, for which we have already obtained a bound on the Lipschitz constant in the previous section. In fact, because it only subtracts $\boldsymbol{\beta}$ from the desired projection of $\boldsymbol{\beta}$, treated as a function of $\boldsymbol{\beta}_n$, it has a Lipschitz constant less than or equal to $\gamma + 1$. If the projection $\overleftarrow{\pi}^{\mathcal{P}}_{f^*}$ has a Lipschitz constant of ϕ, then by repeated application of the Lipschitz bounds, we have a Lipschitz constant for $\boldsymbol{\zeta}_L$ in terms of $\boldsymbol{\zeta}_{-1}$ as $(\gamma+1)^{L+1}\phi^{L+1} + (\gamma+1)^L\phi^{L+1}$. It thus remains to obtain (a bound for) the Lipschitz constant ϕ. But this is just a Lipschitz constant for the projection $\overleftarrow{\pi}^{\mathcal{P}}_{f^*}$, which we also obtained in the previous section when we calculated the Lipschitz constant for $\left[(\mathbf{B}\boldsymbol{\lambda})^T, (\mathbf{B}\boldsymbol{\lambda})^T, \ldots, (\mathbf{B}\boldsymbol{\lambda})^T\right]^T$ with respect to *only one* of the K factors, giving $\phi = \sqrt{K^2 2^{M-2}(M+M^2)}$. This gives us a total Lipschitz constant of

$$\begin{aligned}&C_L \le (\gamma+1)^{L+1}\phi^{L+1} + (\gamma+1)^L\phi^{L+1}, \quad \text{with}\\ &\phi = \sqrt{K^2 2^{M-2}(M+M^2)}\end{aligned}$$

APPENDIX B
PROOF OF THEOREM 3

Consider the log coordinates of the factorization (yielding an initialization $\boldsymbol{\zeta}_{-1}$ in our projections interpretation) to which belief propagation is to be applied. Let $\boldsymbol{\zeta}_{-1}^F$ be the Euclidean projection of $\boldsymbol{\zeta}_{-1}$ onto the set of log coordinates associated with effectively acyclic factorizations, and let the associated minimum Euclidean distance be d and the tree width of the factor graph associated with $\boldsymbol{\zeta}_{-1}^F$ be L. Then by the Triangle inequality

$$\begin{aligned}&\|\boldsymbol{\zeta}_L - \overleftarrow{\pi}^{\mathcal{P}}_{f^*}(\nabla f(\overleftarrow{\pi}^{\mathcal{Q}}_{f}(\nabla f^*(\boldsymbol{\zeta}_{-1}))))\|_2 \le\\ &\|\boldsymbol{\zeta}_L - \boldsymbol{\zeta}_L^F\|_2 + \|\boldsymbol{\zeta}_L^F - \overleftarrow{\pi}^{\mathcal{P}}_{f^*}(\nabla f(\overleftarrow{\pi}^{\mathcal{Q}}_{f}(\nabla f^*(\boldsymbol{\zeta}_{-1}))))\|_2\end{aligned}$$

and by Lipshitz continuity this is bounded as

$$\begin{aligned}&\|\boldsymbol{\zeta}_L - \overleftarrow{\pi}^{\mathcal{P}}_{f^*}(\nabla f(\overleftarrow{\pi}^{\mathcal{Q}}_{f}(\nabla f^*(\boldsymbol{\zeta}_{-1}))))\|_2 \le\\ &C_L\|\boldsymbol{\zeta}_{-1} - \boldsymbol{\zeta}_{-1}^F\|_2 + \gamma\|\boldsymbol{\zeta}_{-1} - \boldsymbol{\zeta}_{-1}^F\|_2 = (C_L+\gamma)d\end{aligned}$$

This shows that the ball of points $\boldsymbol{\zeta}_{-1}$ no more than $d = \frac{\epsilon}{C_L+\gamma}$ away from $\boldsymbol{\zeta}_{-1}^F$ in Euclidean norm gives initializations for which belief propagation yields estimates within Euclidean distance ϵ of their true marginals after L iterations.

REFERENCES

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[2] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[3] C. Berrou and A. Glavieux, "Near optimum error correction coding and decoding: Turbo codes," *IEEE Trans. Communications*, vol. 44, no. 10, pp. 1262–1271, Oct. 1996.

[4] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE Trans. Sel. Areas in Communications*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Information Theory*, vol. 45, no. 2, pp. 399–431, 2001, Mar. 1999.

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

[7] Y. Mao, F. R. Kschischang, B. Li, and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks," *IEEE J. Sel. Areas in Communications*, vol. 23, no. 4, pp. 820–829, Apr. 2005.

[8] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas in Communications*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[9] J. M. Walsh and P. A. Regalia, "Expectation propagation for distributed estimation in sensor networks," in *Proc. Int. Workshop Signal Processing Advances in Wireless Communications*, Helsinki, Finland, June 2007.

[10] ——, "Belief propagation distributed estimation in sensor networks: An optimized energy accuracy tradeoff," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Las Vegas, NV, Apr. 2008.

[11] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Communications*, vol. 47, no. 7, pp. 1046–1061, July 1999.

[12] Z. Shi and C. Schlegel, "Iterative multiuser detection and error control code decoding in random CDMA," *IEEE Trans. Signal Processing*, vol. 54, no. 5, pp. 1886–1895, May 2006.

[13] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Information Theory*, vol. 48, no. 7, pp. 1772–1793, July 2002.

[14] M. J. Wainwright, "Sparse graph codes for side information and binning," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 47–57, Sept. 2007.

[15] P. A. Regalia, "Gradient decoding revisited," in *Proc. Asilomar Conf. Circuits, Systems and Computers*, Pacific Grove, CA, Nov. 2007.

[16] J. M. Walsh, "A completed information projection interpretation of expectation propagation," in *Neural Information Processing Systems Workshop on Approximate Bayesian Inference in Continuous/Hybrid Systems*, 2007.

[17] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Selected Areas in Communications*, vol. 19, no. 5, pp. 891–907, May 2001.

[18] H. El Gamal and A. R. Hommons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 671–686, Feb. 2001.

[19] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Communications*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[20] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[21] L. Kocarev, F. Lehmann, G. M. Maggio, B. Scanvino, Z. Tasev, and A. Vardy, "Nonlinear dynamics of iterative decoding systems: Analysis and applications," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1366–1384, Apr. 2006.

[22] M. Moher and T. A. Gulliver, "Cross-entropy and interative decoding," *IEEE Trans. Information Theory*, vol. 44, no. 7, pp. 3097–3104, Nov. 1998.

[23] A. J. Grant, "Information geometry and iterative decoding," in *IEEE Communications Theory Workhop*, May 1999.

[24] F. Alberge, "Iterative decoding as Dykstra's algorithm with alternate I-projection and reverse I-projection," in *16th European Signal Processing Conference (EUSIPCO)*, 2008.

[25] B. Muquet, P. Duhamel, and M. de Courville, "Geometrical interpretations of iterative 'turbo' decoding," in *Proceedings ISIT*, June 2002.

[26] T. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Information Theory*, vol. 46, no. 1, pp. 9–23, Jan. 2000.

[27] S. Ikeda, T. Tanaka, and S. Amari, "Information geometry of turbo and low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 50, no. 6, pp. 1097–1114, June 2004.

[28] ——, "Stochastic reasoning, free energy and information geometry," *Neural Computation*, pp. 1779–1810, 2004.

[29] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "A refined information geometric interpretation of turbo decoding," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, vol. V, Philadelphia, PA, Mar. 2005, pp. 713–716.

[30] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, pp. 205–237, 1984.

[31] I. Csiszár and F. Matúš, "Information projections revisited," *IEEE Trans. Information Theory*, vol. 49, pp. 1474–1490, June 2003.

[32] R. L. Dykstra, "An iterative procedure for obtaining I-projections onto the intersection of convex sets," *Annals of Probability*, vol. 13, no. 3, pp. 975–984, 1985.

[33] Y. Censor and A. Lent, "An iterative row-action method for interval convex programming," *Journal of Optimization Theory and Applications*, vol. 34, no. 3, pp. 312–353, July 1981.

[34] Y. Censor and S. Reich, "The Dykstra algorithm with Bregman projections," *Communications in Applied Analysis*, vol. 2, pp. 407–419, 1998.

[35] L. M. Bregman, Y. Censor, and S. Reich, "Dykstra's algorithm as the nonlinear extension of Bregman's optimization method," *J. Convex Analysis*, vol. 6, no. 2, pp. 319–333, 1999.

[36] H. H. Bauschke and A. S. Lewis, "Dykstra's algorithm with Bregman projections: A convergence proof," *Optimization*, vol. 48, pp. 409–426, 2000.

[37] J. M. Walsh, "Distributed iterative decoding and estimation via expectation propagation: Performance and convergence," Ph.D. dissertation, Cornell Univ., Ithaca, NY, 2006.

[38] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1970.

[39] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Computational Physics*, vol. 7, pp. 200–217, 1967.

[40] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley, 2007.

[41] H. H. Bauschke, P. L. Combettes, and D. Noll, "Joint minimization with alternating Bregman proximity operators," *Pacific Journal of Optimization*, vol. 2, no. 3, pp. 410–424, Sept. 2006.

[42] I. Csiszár, "I-divergence geometry of probability distributions and minimization problems," *Annals of Probability*, vol. 3, pp. 146–159, 1975.

[43] H. H. Bauschke and J. Borwein, "Dykstra's alternating projection algorithm for two sets," *J. Approximation Theory*, vol. 79, no. 3, pp. 418–443, 1994.

[44] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief approximation algorithms," *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2282–2312, July 2005.

[45] G. David Forney, "Codes on Graphs: Normal Realizations," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[46] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "Turbo decoding as iterative maximum likelihood sequence estimation," *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5426–5437, Dec. 2006.

[47] P. A. Regalia and J. M. Walsh, "Optimality and duality of the turbo decoder," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1362–1377, June 2007.

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

[49] D. Cvetković, P. Rowlinson, and S. Simić, *Eigenspaces of Graphs*, ser. Encyclopedia of Mathematics. Cambridge University Press, 1997, vol. 66.

[50] M. Doob, *Topics in Algebraic Graph Theory*. Cambridge University Press, 2004.

[51] T. P. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, MIT, Cambridge, MA, 2001.

[52] T. Minka, "Divergence measures and message passing," Microsoft Research, Cambridge, UK, Tech. Rep. MSR-TR-2005-173, 2005.

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

[54] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed., ser. Springer Series in Statistics. New York: Springer, 1983.

John MacLaren Walsh (S 2001, M 2007) was born in Carbondale, IL in 1981. He received the B.S. (magna cum laude), M.S., and Ph.D. from Cornell University, Ithaca, NY in 2002, 2004, and 2006 respectively.

In September, 2006 he joined the Department of Electrical and Computer Engineering at Drexel University, Philadelphia, PA where he is currently an assistant professor. He is a member of HKN and TBP. His current research interests include: (a) delay mitigating codes and rate delay tradeoffs in multipath routed and network coded networks, (b) joint source separation and identification, and (c) the performance and convergence of distributed collaborative estimation in wireless sensor networks via expectation propagation.

Phillip A. Regalia Phillip A. Regalia was born in Walnut Creek, CA. He received the PhD in Electrical and Computer Engineering from the University of California at Santa Barbara in 1988, and the *Habilitation à Diriger des Recherches* from the Université Paris-Orsay in 1994. He has served as Editor-in-Chief of the *EURASIP J. Wireless Communications and Networking* and the *EURASIP J. Advances in Signal Processing*, as well as associate editor for the *IEEE Trans. Circuits and Systems II*, the *IEEE Trans. Signal Processing*, and the *IEEE Signal Processing Magazine*. His research interests are focused in signal processing, communications, and adaptive algorithms.