

ECE 590.17: Lecture 7 – LP Inference and the Bethe Free Energy

Graphical Models and Inference for Machine Learning

Duke University, Spring 2026

History: Written by Henry Pfister (2026)

Last Modified: 02/23/2026

Outline of lecture:

7.1	Overview and learning goals	2
7.2	Warm-up: inference, lifting, and relaxation	2
7.2.1	MAP (best assignment) and marginal inference	2
7.2.2	Why factorization matters	3
7.3	The marginal polytope and inference via LP	4
7.3.1	Node and factor marginals as a linear mapping	4
7.3.2	The marginal polytope	4
7.3.3	Why this is not yet algorithmic	5
7.4	Consistency and the local polytope	5
7.4.1	Projection onto the variable marginals	6
7.4.2	Binary variables and log-likelihood ratios	6
7.5	Visualizing small binary CSPs	7
7.5.1	A binary parity-check of degree-2: the 2-bit even-parity line	7
7.5.2	A binary parity-check of degree-3: the 3-bit even-parity tetrahedron	8
7.5.3	Small code example	8
7.5.4	The LP relaxation for MAP	10
7.6	Worked example: Sudoku as a simple LP	12
7.6.1	Indicator variables	12
7.6.2	Local constraints (row/column/block counts)	12
7.6.3	ILP and the basic LP relaxation	13
7.6.4	Why the local polytope can be loose	13
7.7	The Gibbs free energy: exact variational principle	14
7.7.1	KL divergence and a key identity	14
7.7.2	Exact variational characterization of Z	14
7.8	The Bethe approximation: from trees to loopy graphs	15
7.8.1	Rewriting the energy term in local marginals	15
7.8.2	Bethe entropy	15
7.8.3	Bethe free energy	15
7.8.4	Bethe is exact on trees	15
7.9	BP and Bethe: fixed points vs. stationary points	16
7.9.1	BP messages and induced beliefs	17
7.9.2	Factor Graphs without Cycles	17
7.9.3	Factor Graphs with Cycles	19
7.9.4	A constrained optimization problem	20
7.9.5	Main theorem	20
7.10	Connecting the three stories: LP, Bethe, and BP	23
7.10.1	A unifying picture	23
7.10.2	Zero temperature limit (optional detail)	23

7.1 Overview and learning goals

1. **LP-based inference:** how MAP inference (best assignment) can be written as an ILP and relaxed to an LP on *pseudo-marginals*.
2. **The marginal polytope:** the convex set of all realizable collections of marginals.
3. **Bethe free energy and BP:** a principled approximation to the Gibbs free energy that is exact on trees, and whose stationary points correspond to loopy BP fixed points.

Throughout, we use the factor graph $G = (V, F)$ with $V = [n]$ and factors $\{f_a\}_{a \in F}$ to define

$$f(x) = \prod_{a \in F} f_a(x_{\partial a}), \quad x = (x_1, \dots, x_n) \in \mathcal{X}^n,$$

and the associated distribution

$$\mu(x) = \frac{1}{Z} f(x), \quad Z = \sum_{x \in \mathcal{X}^n} f(x).$$

Energy notation. It is often convenient to write

$$f(x) = \exp(-E(x)), \quad E(x) = \sum_{a \in F} E_a(x_{\partial a}), \quad E_a(x_{\partial a}) \triangleq -\ln f_a(x_{\partial a}).$$

Then, the MAP assignment is $\arg \min_x E(x)$, and the partition function is $Z = \sum_x e^{-E(x)}$.

(We use the same neighborhood notation as earlier notes: $\partial a \subseteq [n]$ is the set of variables adjacent to factor a , and $\partial i \subseteq F$ is the set of factors adjacent to variable i .)

Some running examples.

- **Sudoku as a factor graph.** Sudoku is a large sparse constraint satisfaction problem; it motivates local consistency, relaxations, and why loopy methods can be useful.
- **Codes and single-parity-check (SPC) constraints.** A factor node a enforces a parity constraint on ∂a (useful pictorially and algebraically).
- **MAX-SAT.** Version of the boolean satisfiability problem where the goal is to maximize the number of satisfied clauses. In this case, there are n literals $(x_1, \dots, x_n) \in \{0, 1\}^n$ and a factor for each clause. If clause $a \in F$ is satisfied, then the factor $f_a(x_{\partial a}) = 2$ and otherwise it equals 1. Thus, if an assignment $x \in \{0, 1\}^n$ satisfies S clauses, then $f(x) = 2^S$. Taking \log_2 gives the number of satisfied clauses.

7.2 Warm-up: inference, lifting, and relaxation

7.2.1 MAP (best assignment) and marginal inference

Given $f : \mathcal{X}^n \rightarrow \mathbb{R}_{\geq 0}$, two fundamental tasks are:

- **MAP / best assignment:**

$$x^* \in \arg \max_{x \in \mathcal{X}^n} f(x) \quad \Leftrightarrow \quad x^* \in \arg \max_x \ln f(x).$$

- **Marginalization:** compute $\mu_i(x_i) = \sum_{x_{[n]\setminus i}} \mu(x)$ (and similarly factor marginals $\mu_a(x_{\partial a})$).

The idea of linear programming (LP) decoding starts from the observation that any discrete optimization problem can be linearized by lifting from a point $x \in \mathcal{X}^n$ to an indicator distribution over \mathcal{X}^n .

For a finite set \mathcal{A} , let $\mathcal{P}(\mathcal{A})$ denote the set of all probability mass functions $q: \mathcal{A} \rightarrow [0, 1]$ on \mathcal{A} satisfying $\sum_{z \in \mathcal{A}} q(z) = 1$. Similarly, let $\mathcal{I}(\mathcal{A}) \subset \mathcal{P}(\mathcal{A})$ denote the subset of all indicator functions $q: \mathcal{A} \rightarrow \{0, 1\}$ on \mathcal{A} satisfying $\sum_{z \in \mathcal{A}} q(z) = 1$.

Indicator variables and linear programming. Let $\mathcal{P}(\mathcal{X}^n)$ denote the set of probability distributions on \mathcal{X}^n and introduce a variable $q \in \mathcal{P}(\mathcal{X}^n)$, i.e.,

$$q(x) \geq 0, \quad \sum_{x \in \mathcal{X}^n} q(x) = 1.$$

Then, we can write

$$\max_{x \in \mathcal{X}^n} \log f(x) = \max_{q \in \mathcal{P}(\mathcal{X}^n)} \sum_{x \in \mathcal{X}^n} q(x) \ln f(x).$$

Equivalently, in energy form, we have

$$\min_{x \in \mathcal{X}^n} E(x) = \min_{q \in \mathcal{P}(\mathcal{X}^n)} \sum_{x \in \mathcal{X}^n} q(x) E(x) = \min_{q \in \mathcal{P}' } \sum_{x \in \mathcal{X}^n} q(x) E(x),$$

where the *reduced simplex* is defined to be

$$\mathcal{P}' \triangleq \{q \in \mathcal{P}(\mathcal{X}^n) \mid q(x) = 0 \text{ if } E(x) = \infty\}$$

and the last step holds because the objective is infinite if $q(x) > 0$ for some x where $E(x) = \infty$.

Remark 1 (Extreme points make the lifting exact). The feasible set $\mathcal{P}(\mathcal{X}^n)$ is a simplex in $\mathbb{R}^{|\mathcal{X}^n|}$. Its extreme points are the delta distributions $\delta_x \in \mathcal{I}(\mathcal{X}^n)$ defined by

$$\delta_x(z) = \begin{cases} 1 & \text{if } z = x \\ 0 & \text{otherwise.} \end{cases}$$

Since the objective is linear in q , an optimizer may be chosen at an extreme point and hence equals $\delta_{x^*}(z)$. Therefore the lifted LP is *exact* but requires $|\mathcal{X}|^n$ variables.

7.2.2 Why factorization matters

Factor graphs help because f is a product of local functions. Exact algorithms (variable elimination, junction trees, BP) are efficient on trees, but can be expensive or intractable on loopy graphs. This motivates *relaxations* and *variational approximations*.

The lifted LP is exact but intractable because it has $|\mathcal{X}|^n$ variables. In this lecture, we:

1. Replace the global distribution q by writing the cost function in terms of its globally consistent *factor marginals* (called the marginal polytope \mathcal{M})
2. Then, relax the marginal polytope \mathcal{M} into a set of locally consistent marginals \mathcal{L} (called the local polytope) satisfying $\mathcal{M} \subseteq \mathcal{L}$.

Definition 2. An n -dimensional *polytope* is a convex set defined by the convex hull of a finite set of points in \mathbb{R}^n . When $n = 2$ (or $n = 3$), a polytope is called a polygon (or a polyhedron).

Definition 3. A linear program (LP) over a polytope $\mathcal{Q} \subset \mathbb{R}^n$ is given, for some $c \in \mathbb{R}^n$, by

$$\min_{x \in \mathcal{Q}} \langle x, c \rangle.$$

An important result in optimization is that a linear program over a polytope has an optimizer at an extreme point of the polytope. This follows from two simple observations. First, any strictly feasible point can be improved by moving in the opposite direction of the cost vector until we hit the boundary of the polytope. Second, if the objective is constant on an affine subset of the boundary, then it has the same constant value on the extreme points of that subset. Thus, we can restrict to optimizing over extreme points of the boundary.

7.3 The marginal polytope and inference via LP

7.3.1 Node and factor marginals as a linear mapping

Given $q \in \mathcal{P}(\mathcal{X}^n)$, define its node marginals $q_i \in \mathcal{P}(\mathcal{X})$ and factor marginals $q_a \in \mathcal{P}(\mathcal{X}^{|\partial a|})$ by

$$q_i(z_i) = \sum_{x: x_i=z_i} q(x), \quad q_a(z_{\partial a}) = \sum_{x: x_{\partial a}=z_{\partial a}} q(x).$$

This is a linear map Γ from $\mathcal{P}(\mathcal{X}^n)$ into the space of local marginals:

$$\Gamma: \mathcal{P}(\mathcal{X}^n) \rightarrow \prod_{i \in [n]} \mathcal{P}(\mathcal{X}) \times \prod_{a \in F} \mathcal{P}(\mathcal{X}^{|\partial a|}), \quad q \mapsto \underline{q} = ((q_i)_{i \in [n]}, (q_a)_{a \in F}).$$

It follows that, if $q = \delta_x \in \mathcal{I}(\mathcal{X}^n)$ is a delta distribution on x , then $\Gamma(\delta_x)$ is the collection of indicator functions (i.e., deterministic marginals) induced by x . Apologies for abusing notation and using the subscripts i and a to distinguish between node and factor marginals.

7.3.2 The marginal polytope

Definition 4. The *marginal polytope* \mathcal{M} is the image of $\mathcal{P}(\mathcal{X}^n)$ under Γ :

$$\mathcal{M} \triangleq \{ \underline{q} = ((q_i)_{i \in [n]}, \{q_a\}_{a \in F}) \mid \exists \mu \in \mathcal{P}(\mathcal{X}^n) \text{ s.t. } \Gamma \mu = \underline{q} \text{ (i.e., } q_i = \mu_i, q_a = \mu_{\partial a}) \}.$$

The *reduced marginal polytope* \mathcal{M}' is the image of the reduced simplex \mathcal{P}' under Γ . It can also be defined by adding constraints to the definition of \mathcal{M} that force $q_a(z_{\partial a}) = 0$ whenever $E_a(z_{\partial a}) = \infty$.

Remark 5. \mathcal{M} and \mathcal{M}' are convex polytopes because each is the linear image of a simplex. Moreover, $\Gamma(\delta_x)$ is an extreme point of \mathcal{M} , corresponding to the deterministic marginals induced by x .

MAP is a linear program over \mathcal{M} . Define the linear energy functional on factor marginals:

$$\langle E, \underline{q} \rangle \triangleq \sum_{a \in F} \sum_{z_{\partial a} \in \mathcal{X}^{|\partial a|}} q_a(z_{\partial a}) E_a(z_{\partial a}).$$

Theorem 6. The MAP assignment problem can be solved as LP over \mathcal{M} (or \mathcal{M}'):

$$\min_{x \in \mathcal{X}^n} E(x) = \min_{\underline{q} \in \mathcal{M}} \langle E, \underline{q} \rangle \Leftrightarrow \max_x \ln f(x) = - \min_{\underline{q} \in \mathcal{M}} \langle E, \underline{q} \rangle.$$

The optimizer may be chosen to be an extreme point of \mathcal{M} corresponding to a MAP assignment.

Proof. Start from the exact lifted LP:

$$\min_x E(x) = \min_{q \in \mathcal{P}(\mathcal{X}^n)} \sum_x q(x) E(x).$$

Using the factorization $E(x) = \sum_{a \in F} E_a(x_{\partial a})$, we expand

$$\sum_x q(x) E(x) = \sum_x q(x) \sum_{a \in F} E_a(x_{\partial a}) = \sum_{a \in F} \sum_x q(x) E_a(x_{\partial a}).$$

For each a , group terms by $z_{\partial a} = x_{\partial a}$:

$$\sum_x q(x) E_a(x_{\partial a}) = \sum_{z_{\partial a}} \left(\sum_{x: x_{\partial a} = z_{\partial a}} q(x) \right) E_a(z_{\partial a}) = \sum_{z_{\partial a}} q_a(z_{\partial a}) E_a(z_{\partial a}).$$

Therefore

$$\sum_x q(x) E(x) = \sum_a \sum_{z_{\partial a}} q_a(z_{\partial a}) E_a(z_{\partial a}) = \langle E, \underline{q} \rangle.$$

Now minimize over $q \in \mathcal{P}(\mathcal{X}^n)$. The set of achievable collections \underline{q} is exactly \mathcal{M} by definition, so the objective can be written as minimizing $\langle E, \underline{q} \rangle$ over $\underline{q} \in \mathcal{M}$. Finally, since \mathcal{M} is a polytope and the objective is linear, an optimizer exists at an extreme point, and extreme points include projections of δ_x corresponding to deterministic assignments. Similar to the simplex LP, we note that the feasible set \mathcal{M} can be reduced to \mathcal{M}' without changing the result because the objective is infinite if $q(x) > 0$ for some x where $E(x) = \infty$. \square

7.3.3 Why this is not yet algorithmic

Theorem 6 is conceptually clean but computationally hard: describing \mathcal{M} exactly typically requires exponentially many constraints. We now build a tractable outer approximation using *local* constraints.

7.4 Consistency and the local polytope

Definition 7. A collection $\underline{q} = (\{q_i\}_{i \in [n]}, \{q_a\}_{a \in F})$ is *locally consistent* if

$$q_i(z_i) \geq 0, \quad \sum_{z_i} q_i(z_i) = 1, \quad (7.1)$$

$$q_a(z_{\partial a}) \geq 0, \quad \sum_{z_{\partial a}} q_a(z_{\partial a}) = 1, \quad (7.2)$$

$$\sum_{z_{\partial a \setminus i}} q_a(z_{\partial a}) = q_i(z_i), \quad \forall a \in F, \forall i \in \partial a, \forall z_i \in \mathcal{X}. \quad (7.3)$$

The set of all locally consistent marginals is called the *local polytope* \mathcal{L} . It contains \mathcal{M} because all globally consistent marginals must satisfy these local consistency constraints. Moreover, all integral points $\Gamma\delta_x$ in \mathcal{L} correspond to deterministic marginals induced by assignments $x \in \mathcal{X}^n$. If some energies are infinite, then we can define the *reduced local polytope* \mathcal{L}' by adding the constraints $q_a(z_{\partial a}) = 0$ for all $a \in F$ and $z_{\partial a}$ such that $E_a(z_{\partial a}) = \infty$.

The local LP relaxation. The *local LP* for MAP is

$$\min_{\underline{q} \in \mathcal{L}} \langle E, \underline{q} \rangle = \min_{\underline{q} \in \mathcal{L}} \sum_{a \in F} \sum_{z \in \partial a} q_a(z) E_a(z). \quad (7.4)$$

This LP can be solved efficiently if the maximum factor degree is bounded. Like the previous LP, the result is unchanged by restricting \mathcal{L} to the reduced local polytope \mathcal{L}' .

7.4.1 Projection onto the variable marginals

Let f be an indicator function for the set S of valid configurations. In that case, a random configuration X is uniform over S and the reduced marginal polytope \mathcal{M}' captures this structure. Suppose Y be a noisy observation of X through a memoryless channel with

$$\mathbb{P}(Y_i = y_i \mid X_i = x_i) = W_i(y_i \mid x_i).$$

Then, the MAP assignment is

$$\arg \max_{x \in S} \mathbb{P}(Y = y \mid X = x) = \arg \max_{x \in S} \prod_{i=1}^n W_i(y_i \mid x_i).$$

Let the space of variable marginals by $\mathcal{V} \triangleq \mathcal{P}(\mathcal{X})^n$ with elements denoted by

$$\underline{v} = \{v_i\}_{i \in [n]} \in \mathcal{V}, \quad v_i: \mathcal{X} \rightarrow [0, 1].$$

Define the projection $\pi: \mathcal{M}' \rightarrow \mathcal{V}$ where

$$\underline{q} = (\{q_i\}_{i \in [n]}, \{q_a\}_{a \in F}) \mapsto \underline{v} = \{v_i\}_{i \in [n]} \quad \text{with} \quad v_i = q_i.$$

Then, with $\overline{\mathcal{M}}' = \pi(\mathcal{M}')$, we can lift the MAP assignment to the LP

$$\begin{aligned} \underline{q}^*(y) &\in \arg \max_{\underline{q} \in \overline{\mathcal{M}}'} \sum_{i=1}^n \sum_{z \in \mathcal{X}} q_i(z) \ln W(y_i \mid z) \\ &= \pi \left(\arg \max_{\underline{q} \in \mathcal{M}'} \sum_{i=1}^n \sum_{z \in \mathcal{X}} q_i(z) \ln W(y_i \mid z) \right). \end{aligned}$$

For $\overline{\mathcal{L}}' = \pi(\mathcal{L}')$, we can approximate the MAP assignment problem by the LP

$$\begin{aligned} \underline{q}_{LP}^*(y) &\in \arg \max_{\underline{q} \in \overline{\mathcal{L}}'} \sum_{i=1}^n \sum_{z \in \mathcal{X}} q_i(z) \ln W(y_i \mid z) \\ &= \pi \left(\arg \max_{\underline{q} \in \mathcal{L}'} \sum_{i=1}^n \sum_{z \in \mathcal{X}} q_i(z) \ln W(y_i \mid z) \right). \end{aligned}$$

7.4.2 Binary variables and log-likelihood ratios

If $|\mathcal{X}| = 2$, then we can identify \mathcal{X} with $\{0, 1\}$ and simplify $q_i(z)$ to the number $q_i = q_i(1)$ (since $q_i(0) = 1 - q_i(1)$). Using this simplification, the convex hull of the set of valid configurations forms a polytope in the unit cube $[0, 1]^n$. Thus, $|\mathcal{X}| = 2$, we treat $\overline{\mathcal{M}}'$ and $\overline{\mathcal{L}}'$ as polytopes in $[0, 1]^n$.

For decoding, we map the observations to log-likelihood ratios (LLRs)

$$\gamma_i \triangleq \ln \frac{p_i(y_i \mid 0)}{p_i(y_i \mid 1)}.$$

Lemma 8 (LLR linearization). For $|\mathcal{X}| = 2$, there is a constant A (independent of x) such that

$$\ln \mathbb{P}(Y = y \mid X = x) = A - \sum_{i=1}^n \gamma_i x_i.$$

Therefore ML decoding is equivalent to

$$\hat{x}_{\text{ML}} \in \arg \min_{x \in \overline{\mathcal{M}}} \sum_{i=1}^n \gamma_i x_i.$$

Proof. We compute

$$\begin{aligned} \ln \mathbb{P}(Y = y \mid X = x) &= \sum_{i=1}^n \ln p_i(y_i \mid x_i) \\ &= \sum_{i=1}^n \left((1 - x_i) \ln p_i(y_i \mid 0) + x_i \ln p_i(y_i \mid 1) \right) \\ &= \sum_{i=1}^n \ln p_i(y_i \mid 0) + \sum_{i=1}^n x_i \ln \frac{p_i(y_i \mid 1)}{p_i(y_i \mid 0)} \\ &= A - \sum_{i=1}^n \gamma_i x_i, \end{aligned}$$

where $A \triangleq \sum_{i=1}^n \ln p_i(y_i \mid 0)$. Since A does not depend on x , maximizing the log-likelihood is equivalent to minimizing $\sum_i \gamma_i x_i$. \square

7.5 Visualizing small binary CSPs

For a set $S \subseteq \mathcal{X}^n$ of valid configurations, let $f(x) = \mathbb{I}\{x \in S\}$. Each configuration x can be represented by a set $\{q_i\}_{i \in [n]}$ of variable indicator functions $q_i \in \mathcal{P}(\mathcal{X})$, where $q_i(z_i) = \mathbb{I}\{z_i = x_i\}$. In this case, the convex hull of the set of valid configurations $\{q_i\}_{i \in [n]}$ defines a polytope in $\mathbb{R}^{n|\mathcal{X}|}$.

If $|\mathcal{X}| = 2$, then we can identify \mathcal{X} with $\{0, 1\}$ and simplify $q_i(z)$ to the number $q_i = q_i(1)$ (since $q_i(0) = 1 - q_i(1)$). In this case, the convex hull of the set of valid configurations forms a convex polytope in the unit cube $[0, 1]^n$. As an example, consider 3 binary variables satisfying an even parity constraint. As discussed in the last section, the implied polytope in $[0, 1]^3$ is the convex hull of the valid configurations: $(0, 0, 0)$, $(1, 1, 0)$, $(0, 1, 1)$, and $(1, 0, 1)$.

This simplification allows one to visualize some small examples in 3D space.

7.5.1 A binary parity-check of degree-2: the 2-bit even-parity line

Consider two binary variables $(x_1, x_2) \in \{0, 1\}^2$ with an *even parity* constraint:

$$x_1 \oplus x_2 = 0.$$

The valid assignments are $(0, 0)$ and $(1, 1)$. Identify node marginals by $q_i \triangleq \mathbb{P}(X_i = 1) \in [0, 1]$. For a deterministic assignment x , the projected point is

$$(q_1, q_2) = (x_1, x_2) \in \{(0, 0), (1, 1)\}.$$

Hence the *projected exact polytope* (convex hull of valid projected points) is the line segment

$$\overline{\mathcal{M}}'_{\text{SPC}(2)} = \text{conv}\{(0, 0), (1, 1)\} = \{(t, t) : t \in [0, 1]\} \subset [0, 1]^2.$$

In this case the local and exact projected polytopes coincide (there is only one factor and no cycles).

7.5.2 A binary parity-check of degree-3: the 3-bit even-parity tetrahedron

Now consider three binary variables $(x_1, x_2, x_3) \in \{0, 1\}^3$ with even parity:

$$x_1 \oplus x_2 \oplus x_3 = 0.$$

Valid assignments are

$$000, \quad 110, \quad 101, \quad 011.$$

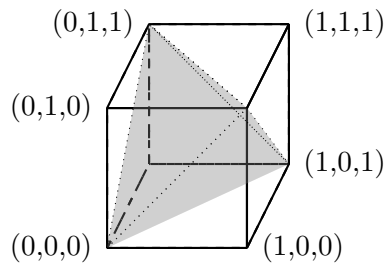
In node-marginal coordinates $q_i = \mathbb{P}(X_i = 1)$, the deterministic projected points are

$$(0, 0, 0), \quad (1, 1, 0), \quad (1, 0, 1), \quad (0, 1, 1).$$

Their convex hull is a tetrahedron inside the unit cube:

$$\overline{\mathcal{M}}'_{\text{SPC}(3)} = \text{conv}\{(0, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1)\} \subset [0, 1]^3.$$

This is exactly the codeword polytope for picture you already draw:



7.5.3 Small code example

Example 9. Consider the binary linear code $\mathcal{C} = \{x \in \{0, 1\}^3 \mid Hx = 0\}$ with parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

This has three parity-check equations:

$$a : x_1 \oplus x_2 = 0, \quad b : x_1 \oplus x_2 \oplus x_3 = 0, \quad c : x_2 \oplus x_3 = 0.$$

As they are linearly independent, the code contains only the all-zero vector. But, the factor graph has a cycle and the local polytope is not equal to the marginal polytope.

To see this, one can verify that the point $\underline{q} = (\{q_i\}, \{q_a\}) \in \mathcal{M}$ defined by

$$q_a(z_1, z_2) = \begin{cases} 2/3 & \text{if } z_1 = z_2 = 1 \\ 1/3 & \text{if } z_1 = z_2 = 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$q_b(z_1, z_2, z_3) = \begin{cases} 1/3 & \text{if } (z_1, z_2, z_3) \in \{110, 101, 011\} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$q_c(z_2, z_3) = \begin{cases} 2/3 & \text{if } z_2 = z_3 = 1 \\ 1/3 & \text{if } z_2 = z_3 = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The variable node marginals are $q_1(1) = q_2(1) = q_3(1) = 2/3$, so this point is a pseudo-codeword.

Geometric description. Let $(x_1, x_2, x_3) \in [0, 1]^3$. Then, the three parity-check polytopes are

$$\begin{aligned} Q_a &= \{(x_1, x_2, x_3) \in [0, 1]^3 : x_1 = x_2\}, \\ Q_b &= \text{conv}\{(0, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1)\}, \\ Q_c &= \{(x_1, x_2, x_3) \in [0, 1]^3 : x_2 = x_3\}. \end{aligned}$$

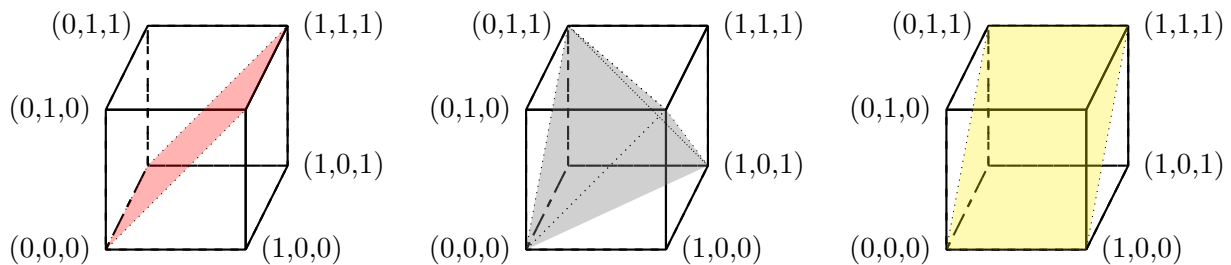
This follows from the earlier visualization examples. Check a enforces $x_1 \oplus x_2 = 0$, so among binary points we must have $(x_1, x_2) \in \{(0, 0), (1, 1)\}$, with x_3 free. Thus, \overline{Q}_a is the convex hull of $\{(0, 0, 0), (0, 0, 1), (1, 1, 0), (1, 1, 1)\}$, i.e., the plane slice $x_1 = x_2$ within the cube. Check c is identical with indices shifted, giving $x_2 = x_3$.

Check b enforces even parity on three bits, so the allowed binary triples are

$$000, 110, 101, 011.$$

The polytope is the convex hull of these four points.

Pictures of $\overline{Q}_a, \overline{Q}_b, \overline{Q}_c$.



Proposition 10 (Computing the projected reduced local polytope). *For this example, let*

$$Q = Q_a \cap Q_b \cap Q_c = \left\{ (t, t, t) : 0 \leq t \leq \frac{2}{3} \right\}.$$

Proof. Intersecting Q_a and Q_c forces $x_1 = x_2 = x_3 = t$ with $t \in [0, 1]$. So $Q_a \cap Q_c$ is the main diagonal segment from $(0, 0, 0)$ to $(1, 1, 1)$.

Now impose membership in $Q_b = \text{conv}\{000, 110, 101, 011\}$. Any point in Q_b can be written as a convex combination

$$x = \alpha 000 + \beta 110 + \gamma 101 + \delta 011, \quad \alpha, \beta, \gamma, \delta \geq 0, \quad \alpha + \beta + \gamma + \delta = 1.$$

The coordinates then satisfy

$$x_1 = \beta + \gamma, \quad x_2 = \beta + \delta, \quad x_3 = \gamma + \delta.$$

If we additionally require $x_1 = x_2 = x_3 = t$, we get the linear system

$$\beta + \gamma = \beta + \delta = \gamma + \delta = t.$$

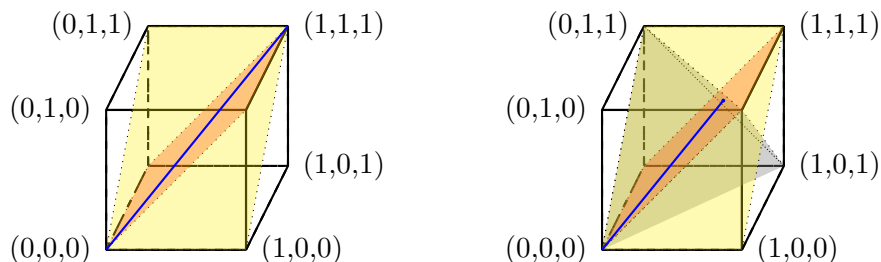
From $\beta + \gamma = \beta + \delta$ we get $\gamma = \delta$. From $\beta + \delta = \gamma + \delta$ we get $\beta = \gamma$. Thus $\beta = \gamma = \delta$. Let $\beta = \gamma = \delta = s \geq 0$. Then $t = \beta + \gamma = 2s$, so $s = t/2$.

Finally the normalization constraint gives

$$\alpha + 3s = 1 \quad \Rightarrow \quad \alpha = 1 - \frac{3t}{2} \geq 0 \quad \Rightarrow \quad t \leq \frac{2}{3}.$$

Therefore the intersection is exactly $\{(t, t, t) : 0 \leq t \leq 2/3\}$. □

Pictures of $Q_a \cap Q_c$ and Q .



Proposition 11 (Extreme points and pseudocodeword). *The polytope $Q = \{(t, t, t) : 0 \leq t \leq 2/3\}$ has exactly two extreme points:*

$$(0, 0, 0) \quad \text{and} \quad \left(\frac{2}{3}, \frac{2}{3}, \frac{2}{3}\right).$$

The point $(0, 0, 0)$ is a codeword, while $(\frac{2}{3}, \frac{2}{3}, \frac{2}{3})$ is a pseudocodeword (a fractional vertex of the relaxation).

Proof. A non-degenerate line segment has the endpoints as its only extreme points. The endpoint $(0, 0, 0)$ is in \mathcal{C} since it satisfies all parity checks. The other endpoint has non-integer coordinates, so it cannot be a convex combination of codewords within $\{0, 1\}^3$ (equivalently, it is not in $\text{conv}(\mathcal{C})$ here), hence it is a fractional vertex of Q . \square

Proposition 12 (When does the pseudocodeword win the LP?). *Consider the LP decoding objective $\min_{x \in \bar{Q}} \sum_{i=1}^3 \gamma_i x_i$. The pseudo codeword acts like a phantom $(1, 1, 1)$ codeword because*

- if $\gamma_1 + \gamma_2 + \gamma_3 > 0$, the unique optimum is $(0, 0, 0)$;
- if $\gamma_1 + \gamma_2 + \gamma_3 < 0$, the unique optimum is $(2/3, 2/3, 2/3)$;
- if $\gamma_1 + \gamma_2 + \gamma_3 = 0$, every point on the segment is optimal.

Proof. Parametrize $x = (t, t, t)$ with $t \in [0, 2/3]$. The objective equals

$$\sum_{i=1}^3 \gamma_i x_i = t(\gamma_1 + \gamma_2 + \gamma_3),$$

which is minimized at $t = 0$ if the sum is positive, at $t = 2/3$ if the sum is negative, and is constant if the sum is zero. \square

What should students take away? This example is small enough that we can see the geometry. Even though each check polytope \bar{Q}_a looks “reasonable,” their intersection can create a fractional extreme point. That fractional extreme point can beat the true codeword for certain LLRs, producing an LP decoding error.

7.5.4 The LP relaxation for MAP

Definition 13 (LP relaxation for MAP). The *LP relaxation* (a.k.a. *local LP*) for MAP is

$$\min_{(\mu_i, \mu_a) \in \mathcal{L}} \langle E, \mu \rangle. \quad (7.5)$$

Lemma 14 (Tightness on trees). *If the factor graph is a tree (acyclic), then $\mathcal{L} = \mathcal{M}$ and the local LP (7.5) returns an exact MAP solution.*

Proof. We give a constructive proof: on a tree, any locally consistent collection of marginals can be “glued” into a global distribution.

Step 1: build a candidate global distribution. Assume the factor graph is connected and acyclic. For each variable node i let $d_i = |\partial i|$. Given $(\mu_i, \mu_a) \in \mathcal{L}$ with full support (i.e., all entries strictly positive), define

$$\tilde{\mu}(x) \triangleq \frac{\prod_{a \in F} \mu_a(x_{\partial a})}{\prod_{i \in [n]} \mu_i(x_i)^{d_i-1}}. \quad (7.6)$$

(If some marginals have zeros, the same proof goes through by restricting to their support and taking limits; see Remark 15.)

Step 2: show $\tilde{\mu}$ is normalized. Since the factor graph is a tree, we can choose an arbitrary variable node r to be the root and orient edges away from the root. In the resulting directed graph, every node (variable or factor) except the root has a unique parent. Now eliminate leaves iteratively:

- If a variable node i is a leaf with parent factor a , then in (7.6) the only factor involving x_i is $\mu_a(x_{\partial a})$ in the numerator and $\mu_i(x_i)^{d_i-1} = \mu_i(x_i)^0 = 1$ in the denominator. Thus, we can remove variable i and factor a while adding a new factor b with $\partial b = \partial a \setminus i$ and defining

$$\mu_b(x_{\partial b}) = \sum_{x_i} \mu_a(x_{\partial a}),$$

i.e., we marginalize x_i out of μ_a .

- If a factor node a is a leaf with parent variable i , then a appears exactly once in the numerator and $\partial a = \{i\}$ implies $\mu_i(x_i)^{d_i-1}$ appears in the denominator. By local consistency (the sum over $x_{\partial a \setminus i}$ is over the empty set), we have $\mu_a(x_i) = \mu_i(x_i)$. Hence

$$\frac{\mu_a(x_i)}{\mu_i(x_i)^{d_i-1}} = \frac{1}{\mu_i(x_i)^{d_i-2}}.$$

After deleting the leaf factor a , the degree of i drops by one, so its denominator exponent in (7.6) is exactly $d_i - 2$. Therefore, we can remove a without changing the value of the expression.

Carrying out this elimination from the leaves up to the root shows that $\sum_x \tilde{\mu}(x) = 1$.

Step 3: show $\tilde{\mu}$ has the desired marginals. A similar elimination argument shows that the marginal of $\tilde{\mu}$ on any factor scope ∂a equals μ_a , and on any variable node equals μ_i . Concretely, to compute $\tilde{\mu}_{\partial a}(x_{\partial a})$, sum (7.6) over all variables not in ∂a . Because the factor graph is a tree, the variables outside ∂a decompose into subtrees attached to ∂a , and each subtree sum collapses to 1 by the same cancellations.

Step 4: conclude $\mathcal{L} \subseteq \mathcal{M}$ on trees. The construction shows that any $(\mu_i, \mu_a) \in \mathcal{L}$ is realizable by a global distribution (namely $\tilde{\mu}$), hence belongs to \mathcal{M} . Since always $\mathcal{M} \subseteq \mathcal{L}$, we obtain $\mathcal{L} = \mathcal{M}$.

Finally, because Theorem 6 gives exact MAP as a linear program over \mathcal{M} and $\mathcal{L} = \mathcal{M}$, the local LP (7.5) is exact on trees. \square

Remark 15. In (7.6), zeros in factor marginals μ_a are harmless because they only appear in the numerator. The only potential singularities come from node marginals μ_i in the denominator. For rigorous handling, one can treat the alphabets for all variables as distinct and remove the impossible values from the correct alphabets.

Example 16. Consider the binary linear code from Example 9. The factors f_a, f_b, f_c take the value 0 whenever their arguments contain an odd number of ones. Thus, the marginal polytope is reduced by the equality constraints

$$\begin{aligned} 0 &= q_a(0, 1) = q_a(1, 0) \\ 0 &= q_b(1, 0, 0) = q_b(0, 1, 0) = q_b(0, 0, 1) = q_b(1, 1, 1) \\ 0 &= q_c(0, 1) = q_c(1, 0). \end{aligned}$$

7.6 Worked example: Sudoku as a simple LP

Sudoku is a constraint satisfaction problem (CSP), so we can view it as MAP inference with zero-one factors. The main point of this example is that, after introducing indicator variables, the *local LP* becomes an extremely simple linear program whose constraints are just “certain sums equal 1.”

7.6.1 Indicator variables

Consider an $N \times N$ Sudoku with symbols $\mathcal{X} = [N]$ (e.g., $N = 9$ with 3×3 blocks). For each cell $(r, c) \in [N] \times [N]$ and value $v \in [N]$, introduce an indicator variable

$$q_{r,c}(v) \in \{0, 1\}, \quad q_{r,c}(v) = 1 \iff x_{r,c} = v.$$

(These are the Sudoku analogue of the q_i variables in the MAP-ILP discussion.)

In a full factor-graph LP, we would also introduce factor marginals q_a : cell factors enforce “one value per cell,” and row/column/block factors enforce “each value appears exactly once” on their scopes. Local consistency would require each $q_{r,c}(v)$ to match the corresponding marginals of every incident factor.

For Sudoku, these consistency equations reduce to simple linear count constraints. For example, a row factor for row r and value v implies $\sum_{c=1}^N q_{r,c}(v) = 1$; similarly, column and block factors imply the analogous column/block equalities, and cell factors imply $\sum_{v=1}^N q_{r,c}(v) = 1$. Hence we can work directly with the $q_{r,c}(v)$ variables: the usual local-consistency requirements are already encoded by these linear constraints.

7.6.2 Local constraints (row/column/block counts)

The Sudoku rules can be written as linear constraints in the indicators.

(i) **Exactly one value per cell.**

$$\sum_{v=1}^N q_{r,c}(v) = 1, \quad \forall r, c \in [N]. \quad (7.7)$$

(ii) **Each value appears exactly once in each row.**

$$\sum_{c=1}^N q_{r,c}(v) = 1, \quad \forall r \in [N], \forall v \in [N]. \quad (7.8)$$

(iii) **Each value appears exactly once in each column.**

$$\sum_{r=1}^N q_{r,c}(v) = 1, \quad \forall c \in [N], \forall v \in [N]. \quad (7.9)$$

(iv) **Each value appears exactly once in each block.** Let \mathcal{B} be the set of blocks, and write $B \in \mathcal{B}$ for a block (a set of cell indices). Then

$$\sum_{(r,c) \in B} q_{r,c}(v) = 1, \quad \forall B \in \mathcal{B}, \forall v \in [N]. \quad (7.10)$$

Clues. A clue “cell (r, c) equals v^* ” is just

$$q_{r,c}(v^*) = 1 \quad (\text{equivalently } q_{r,c}(v) = 0 \text{ for } v \neq v^*). \quad (7.11)$$

7.6.3 ILP and the basic LP relaxation

With integrality, Sudoku is an ILP feasibility problem:

$$q_{r,c}(v) \in \{0, 1\} \quad \text{and} \quad (7.7), (7.8), (7.9), (7.10), (7.11).$$

Dropping integrality yields the *basic Sudoku LP*:

$$\begin{aligned} \text{find } \{q_{r,c}(v)\} \quad \text{s.t.} \quad & (7.7), (7.8), (7.9), (7.10), (7.11), \\ & 0 \leq q_{r,c}(v) \leq 1, \quad \forall r, c, v. \end{aligned} \quad (7.12)$$

(Optionally, add a tiny tie-breaking linear objective $\min \sum_{r,c,v} w_{r,c,v} q_{r,c}(v)$; otherwise it is a pure feasibility LP.)

Remark 17 (How this fits the general framework). This is exactly the “indicator \rightarrow relax” pipeline:

- Integral feasible points correspond to valid Sudoku solutions.
- Fractional feasible points are *pseudo-solutions*: they satisfy all the local counting constraints, but do not correspond to any single grid assignment.
- In the language of graphical models, the variables $q_{r,c}(\cdot)$ are node beliefs, and (7.8)–(7.10) are local marginal-consistency constraints. Thus the feasible set of (7.12) is a concrete instance of the *local polytope* \mathcal{L} .

Remark 18 (One-line pseudo-solution). If there are no clues, the uniform assignment $q_{r,c}(v) = 1/N$ satisfies (7.7)–(7.10) but is not integral. Strictly speaking, this is not a valid Sudoku because the solution is not unique. It is the correct analogue of a pseudocodeword though: the LP enforces the constraints only “on average.” There are other valid Sudokus where the best assignment is unique but there are pseudo-solutions.

7.6.4 Why the local polytope can be loose

On loopy graphs, \mathcal{L} is only enforcing pairwise consistency between each factor and its incident variables. There is no global constraint that guarantees these local marginals arise from a single joint distribution. This gap is exactly the gap between \mathcal{M} and \mathcal{L} .

Example 19. Sudoku factor graphs have many cycles. Local consistency constraints can ensure, e.g., that a cell marginal agrees with the marginal induced by its row constraint, but that does not imply there exists a global Sudoku solution distribution consistent with all these marginals. Thus \mathcal{L} may contain points that look locally consistent but are globally impossible.

7.7 The Gibbs free energy: exact variational principle

7.7.1 KL divergence and a key identity

Definition 20 (KL divergence). For distributions $\nu, \mu \in \mathcal{P}(\mathcal{X}^n)$ with $\text{supp}(\nu) \subseteq \text{supp}(\mu)$, define

$$D(\nu\|\mu) \triangleq \sum_{x \in \mathcal{X}^n} \nu(x) \ln \frac{\nu(x)}{\mu(x)}.$$

Lemma 21 (Nonnegativity of KL). $D(\nu\|\mu) \geq 0$ with equality iff $\nu = \mu$.

Proof. This is a standard consequence of Jensen's inequality applied to the convex function $t \mapsto t \ln t$ (or equivalently, \ln concavity). A short proof: since $-\ln$ is convex,

$$D(\nu\|\mu) = \mathbb{E}_\nu \left[\ln \frac{\nu(X)}{\mu(X)} \right] \geq \ln \left(\mathbb{E}_\nu \left[\frac{\nu(X)}{\mu(X)} \right] \right) = \ln \left(\sum_x \nu(x) \frac{\nu(x)}{\mu(x)} \right),$$

and one can refine this to obtain $D(\nu\|\mu) \geq 0$ (the cleanest route is the log-sum inequality). We omit the standard details. \square

7.7.2 Exact variational characterization of Z

Define the (exact) Gibbs free energy functional

$$G(\nu) \triangleq \mathbb{E}_\nu[E(X)] - H(\nu) = \sum_x \nu(x)E(x) + \sum_x \nu(x) \ln \nu(x), \quad \nu \in \mathcal{P}(\mathcal{X}^n).$$

Theorem 22 (Exact variational principle). Let $\mu(x) = \frac{1}{Z}e^{-E(x)}$. Then

$$-\ln Z = \min_{\nu \in \mathcal{P}(\mathcal{X}^n)} G(\nu),$$

and the unique minimizer is $\nu^* = \mu$.

Proof. Since $\mu(x) = \frac{1}{Z}e^{-E(x)}$, we have

$$E(x) = -\ln \mu(x) - \ln Z.$$

Substitute into $G(\nu)$:

$$\begin{aligned} G(\nu) &= \sum_x \nu(x) (-\ln \mu(x) - \ln Z) + \sum_x \nu(x) \ln \nu(x) \\ &= -\ln Z + \sum_x \nu(x) \ln \frac{\nu(x)}{\mu(x)} \\ &= -\ln Z + D(\nu\|\mu). \end{aligned}$$

By Lemma 21, $D(\nu\|\mu) \geq 0$ with equality iff $\nu = \mu$. Thus the minimum of $G(\nu)$ is $-\ln Z$, uniquely achieved at $\nu = \mu$. \square

Interpretation. The exact inference problem (computing μ and Z) can be viewed as an optimization problem. Unfortunately, the decision variable ν is a distribution over \mathcal{X}^n , which is exponentially large.

The next step is to express (or approximate) this optimization in terms of local marginals.

7.8 The Bethe approximation: from trees to loopy graphs

7.8.1 Rewriting the energy term in local marginals

The energy decomposes as $E(x) = \sum_{a \in F} E_a(x_{\partial a})$. For any distribution ν with factor marginals $\nu_{\partial a}$, we have

$$\mathbb{E}_\nu[E(X)] = \sum_{a \in F} \mathbb{E}_\nu[E_a(X_{\partial a})] = \sum_{a \in F} \sum_{z_{\partial a}} \nu_{\partial a}(z_{\partial a}) E_a(z_{\partial a}).$$

Thus the energy term depends only on factor marginals. The difficulty is the entropy term $H(\nu)$, which generally depends on the full joint.

On trees, there is a miracle: the joint distribution can be reconstructed from consistent local marginals, and the entropy can be written in terms of local entropies. Bethe extends these exact tree identities to loopy graphs as an approximation.

7.8.2 Bethe entropy

For any distribution ρ on a finite alphabet, define its entropy

$$H(\rho) = \sum_z \rho(z) \ln \frac{1}{\rho(z)}.$$

Definition 23 (Bethe entropy). For $(\{\mu_i\}, \{\mu_a\}) \in \mathcal{L}$ define

$$H(\mu)_B \triangleq \sum_{a \in F} H(\mu_a) - \sum_{i=1}^n (d_i - 1) H(\mu_i), \quad d_i \triangleq |\partial i|.$$

Remark 24. On a tree, $H(\mu)_B$ equals the true entropy of the unique joint distribution consistent with (μ_i, μ_a) . On loopy graphs, $H(\mu)_B$ is an approximation and may fail to be concave.

7.8.3 Bethe free energy

Definition 25 (Bethe free energy). For $(\mu_i, \mu_a) \in \mathcal{L}$ define the *Bethe free energy*

$$G_B(\mu) \triangleq \underbrace{\sum_{a \in F} \sum_{z_{\partial a}} \mu_a(z_{\partial a}) E_a(z_{\partial a})}_{\text{average energy}} - \underbrace{H(\mu)_B}_{\text{Bethe entropy}}.$$

What we are doing conceptually. Compare Theorem 22 to G_B :

- We replace the true entropy $H(\nu)$ by $H(\mu)_B$, which depends only on local beliefs.
- We relax the feasible set from the marginal polytope \mathcal{M} to the local polytope \mathcal{L} .

This yields a tractable (but generally non-convex) variational problem.

7.8.4 Bethe is exact on trees

Theorem 26 (Exactness of Bethe on trees). *Assume the factor graph is a tree. Then:*

1. $\mathcal{L} = \mathcal{M}$ (already proved in Lemma 14).

2. For every $(\mu_i, \mu_a) \in \mathcal{L}$, the reconstructed joint $\tilde{\mu}$ from (7.6) satisfies

$$H(\tilde{\mu}) = \sum_{a \in F} H(\mu_a) - \sum_{i=1}^n (d_i - 1) H(\mu_i) = H(\mu)_B.$$

3. Consequently,

$$-\ln Z = \min_{\nu \in \mathcal{P}(\mathcal{X}^n)} G(\nu) = \min_{(\mu_i, \mu_a) \in \mathcal{L}} G_B(\mu),$$

and the minimizer corresponds to the true marginals of μ .

Proof. We already proved $\mathcal{L} = \mathcal{M}$ for trees, so it remains to prove the entropy identity.

Step 1: express $\tilde{\mu}$ in logarithms. From (7.6),

$$\ln \tilde{\mu}(x) = \sum_{a \in F} \ln \mu_a(x_{\partial a}) - \sum_{i=1}^n (d_i - 1) \ln \mu_i(x_i).$$

Step 2: take expectation under $\tilde{\mu}$. Using $H(\tilde{\mu}) = -\mathbb{E}_{\tilde{\mu}}[\ln \tilde{\mu}(X)]$, we get

$$\begin{aligned} H(\tilde{\mu}) &= - \sum_x \tilde{\mu}(x) \left(\sum_{a \in F} \ln \mu_a(x_{\partial a}) - \sum_{i=1}^n (d_i - 1) \ln \mu_i(x_i) \right) \\ &= - \sum_{a \in F} \sum_x \tilde{\mu}(x) \ln \mu_a(x_{\partial a}) + \sum_{i=1}^n (d_i - 1) \sum_x \tilde{\mu}(x) \ln \mu_i(x_i). \end{aligned}$$

Step 3: rewrite the sums using marginals. Because $\tilde{\mu}$ has factor marginals μ_a and node marginals μ_i (from the reconstruction proof),

$$\begin{aligned} \sum_x \tilde{\mu}(x) \ln \mu_a(x_{\partial a}) &= \sum_{z_{\partial a}} \mu_a(z_{\partial a}) \ln \mu_a(z_{\partial a}), \\ \sum_x \tilde{\mu}(x) \ln \mu_i(x_i) &= \sum_{z_i} \mu_i(z_i) \ln \mu_i(z_i). \end{aligned}$$

Therefore,

$$H(\tilde{\mu}) = - \sum_{a \in F} \sum_{z_{\partial a}} \mu_a(z_{\partial a}) \ln \mu_a(z_{\partial a}) + \sum_{i=1}^n (d_i - 1) \sum_{z_i} \mu_i(z_i) \ln \mu_i(z_i),$$

which is exactly

$$H(\tilde{\mu}) = \sum_{a \in F} H(\mu_a) - \sum_{i=1}^n (d_i - 1) H(\mu_i) = H(\mu)_B.$$

Step 4: conclude the variational statements. On a tree, minimizing G_B over $\mathcal{L} = \mathcal{M}$ is the same as minimizing the exact free energy $G(\nu)$ over ν , so the minimum equals $-\ln Z$ by Theorem 22, and the minimizer yields the true marginals. \square

7.9 BP and Bethe: fixed points vs. stationary points

This is the key conceptual result for the second lecture: *loopy BP is not arbitrary; it is attempting to optimize a variational objective.* Specifically, BP fixed points coincide with stationary points of the Bethe free energy (under mild positivity assumptions).

7.9.1 BP messages and induced beliefs

Recall the (sum-product) BP updates on a factor graph:

$$m_{i \rightarrow a}(x_i) \propto \prod_{b \in \partial i \setminus a} \hat{m}_{b \rightarrow i}(x_i), \quad (7.13)$$

$$\hat{m}_{a \rightarrow i}(x_i) \propto \sum_{x_{\partial a \setminus i}} f_a(x_{\partial a}) \prod_{j \in \partial a \setminus i} m_{j \rightarrow a}(x_j). \quad (7.14)$$

At a fixed point of these updates, we define the corresponding *beliefs*

$$\mu_i(x_i) \propto \prod_{a \in \partial i} \hat{m}_{a \rightarrow i}(x_i), \quad (7.15)$$

$$\mu_a(x_{\partial a}) \propto f_a(x_{\partial a}) \prod_{i \in \partial a} m_{i \rightarrow a}(x_i). \quad (7.16)$$

It is a good exercise (and an important sanity check) to verify that these beliefs satisfy local consistency:

$$\sum_{x_{\partial a \setminus i}} \mu_a(x_{\partial a}) = \mu_i(x_i) \quad \text{after normalization.}$$

Thus BP fixed points naturally produce points in \mathcal{L} .

Indeed, if we normalize

$$\mu_i(x_i) = \frac{1}{Z_i} \prod_{b \in \partial i} \hat{m}_{b \rightarrow i}(x_i), \quad \mu_a(x_{\partial a}) = \frac{1}{Z_a} f_a(x_{\partial a}) \prod_{j \in \partial a} m_{j \rightarrow a}(x_j),$$

then μ_i, μ_a are nonnegative and have unit mass by construction. For local consistency,

$$\begin{aligned} \sum_{x_{\partial a \setminus i}} \mu_a(x_{\partial a}) &\propto m_{i \rightarrow a}(x_i) \sum_{x_{\partial a \setminus i}} f_a(x_{\partial a}) \prod_{j \in \partial a \setminus i} m_{j \rightarrow a}(x_j) \\ &\propto m_{i \rightarrow a}(x_i) \hat{m}_{a \rightarrow i}(x_i) \propto \prod_{b \in \partial i} \hat{m}_{b \rightarrow i}(x_i) \propto \mu_i(x_i), \end{aligned}$$

where we used (7.14) and (7.13). After normalization this gives $\sum_{x_{\partial a \setminus i}} \mu_a(x_{\partial a}) = \mu_i(x_i)$, so the BP-induced beliefs lie in \mathcal{L} .

7.9.2 Factor Graphs without Cycles

If the factor graph does not have any cycles, then inference and analysis are both greatly simplified. In particular, the sum-product algorithm (SPA), which is also called belief propagation (BP), can be used to efficiently compute the *factor marginals* $\{\mu_{\partial a}\}_{a \in F}$ and $\{\mu_i\}_{i \in V}$. In this subsection we also allow optional unary factors $f_i(x_i)$; in the earlier notation this is the special case $f_i \equiv 1$.

The message-passing update rules of the SPA are given by

$$\begin{aligned} \mu_{j \rightarrow a}^{(\ell+1)}(x) &\propto f_j(x) \prod_{b \in \partial j \setminus a} \hat{\mu}_{b \rightarrow j}^{(\ell)}(x) \\ \hat{\mu}_{a \rightarrow j}^{(\ell)}(x) &\propto \sum_{\underline{x}_{\partial a}} f_a(\underline{x}_{\partial a}) \delta_{x_j, x} \prod_{i \in \partial a \setminus j} \mu_{i \rightarrow a}^{(\ell)}(x_i), \end{aligned} \quad (7.17)$$

along with the normalization conditions $\sum_{x \in \mathcal{X}} \mu_{j \rightarrow a}^{(\ell)}(x) = 1$ and $\sum_{x \in \mathcal{X}} \hat{\mu}_{a \rightarrow j}^{(\ell)}(x) = 1$. The symbol $\delta_{x_j, x}$ denotes the Kronecker delta function and equals 1 if $x_j = x$ and 0 otherwise. The algorithm is

typically initialized to $\mu_{j \rightarrow a}^{(0)}(x) \propto f_j(x)$. If the factor graph does not have cycles, then this iteration converges to a fixed point after a finite number of steps and we denote the fixed point messages by $\hat{\mu}_{a \rightarrow j}^*(x)$ and $\mu_{j \rightarrow a}^*(x)$. In this case, the factor marginals are given by

$$\begin{aligned}\mu_i(x) &\propto f_i(x) \prod_{b \in \partial i} \hat{\mu}_{b \rightarrow i}^*(x) \\ \mu_{\partial a}(\underline{x}_{\partial a}) &\propto f_a(\underline{x}_{\partial a}) \prod_{i \in \partial a} \mu_{i \rightarrow a}^*(x_i).\end{aligned}\tag{7.18}$$

Another consequence of the factor graph not having cycles is that the joint distribution μ can be written as a function of the factor marginals. This is especially convenient given that these marginals are easily computed with the SPA. The following lemma makes this precise.

Lemma 27. *Consider a factor graph without cycles. Let A be any subset of factor nodes whose induced subgraph is connected and define*

$$\partial A \triangleq \bigcup_{a \in A} \partial a,$$

the set of variable nodes adjacent to A . Then, the marginal $\mu_{\partial A}$ can be written as

$$\mu_{\partial A}(\underline{x}_{\partial A}) = \left(\prod_{a \in A} \frac{\mu_{\partial a}(\underline{x}_{\partial a})}{\prod_{i \in \partial a} \mu_i(x_i)} \right) \left(\prod_{i \in \partial A} \mu_i(x_i) \right).$$

Proof. The proof is by induction on $|A|$. If $|A| = 1$, then let b denote the single factor node in A and observe that the base case

$$\mu_{\partial b}(\underline{x}_{\partial b}) = \left(\frac{\mu_{\partial b}(\underline{x}_{\partial b})}{\prod_{i \in \partial b} \mu_i(x_i)} \right) \left(\prod_{i \in \partial b} \mu_i(x_i) \right) = \mu_{\partial b}(\underline{x}_{\partial b})$$

holds trivially. The subgraph, $S(A)$, induced by A is a tree because it is a connected subgraph of a cycle free graph. If $|A| > 1$, then choose $b \in A$ to be any factor node with $|\partial b| \geq 2$ that is adjacent to a leaf variable node. Such a b exists because $S(A)$ is a tree and $|A| > 1$. Since $S(A)$ is a tree, there is a unique variable node $k \in \partial b$ that is in both ∂b and $\partial(A \setminus b)$. In this case, $S(A \setminus b)$ is connected and $|A \setminus b| = |A| - 1$. Therefore, we can apply the induction hypothesis to get the formula for $\mu_{\partial(A \setminus b)}$. Since x_k separates $\partial(A \setminus b)$ and ∂b in the factor graph, conditional independence implies

$$\begin{aligned}\mu_{\partial A}(\underline{x}_{\partial A}) &= \mu_{\partial(A \setminus b)}(\underline{x}_{\partial(A \setminus b)}) \mu_{\partial b \setminus k | \{k\}}(\underline{x}_{\partial b \setminus k} | x_k) \\ &= \mu_{\partial(A \setminus b)}(\underline{x}_{\partial(A \setminus b)}) \frac{\mu_{\partial b}(\underline{x}_{\partial b})}{\mu_k(x_k)} \\ &= \left(\prod_{a \in A \setminus b} \frac{\mu_{\partial a}(\underline{x}_{\partial a})}{\prod_{i \in \partial a} \mu_i(x_i)} \right) \left(\prod_{i \in \partial(A \setminus b)} \mu_i(x_i) \right) \frac{\mu_{\partial b}(\underline{x}_{\partial b})}{\mu_k(x_k)} \\ &= \left(\prod_{a \in A} \frac{\mu_{\partial a}(\underline{x}_{\partial a})}{\prod_{i \in \partial a} \mu_i(x_i)} \right) \left(\prod_{i \in \partial b} \mu_i(x_i) \right) \left(\prod_{i \in \partial(A \setminus b)} \mu_i(x_i) \right) \frac{1}{\mu_k(x_k)} \\ &= \left(\prod_{a \in A} \frac{\mu_{\partial a}(\underline{x}_{\partial a})}{\prod_{i \in \partial a} \mu_i(x_i)} \right) \left(\prod_{i \in \partial A} \mu_i(x_i) \right),\end{aligned}$$

where the last equality holds because $\partial(A \setminus b) \cap \partial b = \{k\}$ implies that the second and third products have an extra factor of $\mu_k(x_k)$ that cancels the $1/\mu_k(x_k)$ term. \square

Remark 28. Choosing $A = F$ in the above lemma shows that the formula holds for any tree factor graph. Likewise, it is easy to verify that conditional independence implies the formula also holds for any factor graph without cycles (i.e., consisting of disjoint tree components).

Lemma 29. *For a factor graph without cycles, the entropy of μ can be written in terms of the factor marginals as*

$$H(\mu) = \sum_{a \in F} H(\mu_{\partial a}) - \sum_{i \in V} (|\partial i| - 1) H(\mu_i) \quad (7.19)$$

and the free energy can be written as

$$-\ln Z = \sum_{a \in F} \left[\sum_{\underline{x}_{\partial a}} \mu_{\partial a}(\underline{x}_{\partial a}) \ln \frac{1}{f_a(\underline{x}_{\partial a})} \right] + \sum_{i \in V} \left[\sum_{x_i} \mu_i(x_i) \ln \frac{1}{f_i(x_i)} \right] - H(\mu). \quad (7.20)$$

Proof. Using the form of $\mu(\underline{x})$ given by Lemma 27, one can directly compute the entropy with

$$\begin{aligned} H(\mu) &= \sum_{\underline{x} \in \mathcal{X}^n} \mu(\underline{x}) \ln \frac{1}{\mu(\underline{x})} \\ &= - \sum_{\underline{x} \in \mathcal{X}^n} \mu(\underline{x}) \ln \left[\left(\prod_{a \in F} \frac{\mu_{\partial a}(\underline{x}_{\partial a})}{\prod_{i \in \partial a} \mu_i(x_i)} \right) \left(\prod_{i \in V} \mu_i(x_i) \right) \right] \\ &= - \sum_{\underline{x} \in \mathcal{X}^n} \mu(\underline{x}) \left[\sum_{a \in F} \ln \mu_{\partial a}(\underline{x}_{\partial a}) - \sum_{i \in V} (|\partial i| - 1) \ln \mu_i(x_i) \right] \\ &= \sum_{a \in F} \sum_{\underline{x} \in \mathcal{X}^n} \mu(\underline{x}) \ln \frac{1}{\mu_{\partial a}(\underline{x}_{\partial a})} - \sum_{i \in V} (|\partial i| - 1) \sum_{\underline{x} \in \mathcal{X}^n} \mu(\underline{x}) \ln \frac{1}{\mu_i(x_i)} \\ &= \sum_{a \in F} H(\mu_{\partial a}) - \sum_{i \in V} (|\partial i| - 1) H(\mu_i). \end{aligned}$$

The second result follows from Theorem 22, which gives $G(\mu) = -\ln Z$, together with the definition $G(\nu) = \mathbb{E}_\nu[E(X)] - H(\nu)$. \square

7.9.3 Factor Graphs with Cycles

For factor graphs with cycles, there is no guarantee that the SPA will converge or give useful results. Still, one can use equations that are exact for factor graphs without cycles as approximations for arbitrary factor graphs and hope for the best. This approach is sometimes called the *Bethe formalism*. Roughly speaking, the idea is to identify fixed points of the SPA and, at each fixed point, use (7.18) and Lemma 27 to estimate $\mu(\underline{x})$. Under various conditions, this can give good results. For example, if the SPA has a unique fixed point and the factor graph has large girth, then the marginal of any node is essentially determined by its tree-like neighborhood on which the SPA is exact.

Consider a set of variable node beliefs $b \triangleq \{b_i : \mathcal{X} \rightarrow [0, 1]\}_{i \in V}$ and factor node beliefs $\hat{b} \triangleq \{\hat{b}_a : \mathcal{X}^{|\partial a|} \rightarrow [0, 1]\}_{a \in F}$ satisfying the marginal consistency constraints

$$\sum_{\underline{x}_{\partial a \setminus i}} \hat{b}_a(\underline{x}_{\partial a}) = b_i(x_i),$$

for all $(i, a) \in E$ and $x_i \in \mathcal{X}$. The set of all (b, \hat{b}) pairs satisfying these conditions is called the *marginal polytope* associated with the factor graph and denoted by \mathcal{M} . A set of factor node beliefs

\hat{b} is called *consistent* if there is a b such that $(b, \hat{b}) \in \mathcal{M}$ and the set of all such beliefs is denoted by \mathcal{M}' . For any $\hat{b} \in \mathcal{M}'$, the *Bethe entropy* is defined to be

$$H_B(\hat{b}) \triangleq \sum_{a \in F} \sum_{\underline{x}_{\partial a}} \hat{b}_a(\underline{x}_{\partial a}) \ln \frac{1}{\hat{b}_a(\underline{x}_{\partial a})} - \sum_{i \in V} (|\partial i| - 1) \sum_{x_i} b_i(x_i) \ln \frac{1}{b_i(x_i)},$$

where we note that the b_i 's can be computed from the \hat{b}_a 's. Basically, this formula treats the beliefs as marginals and applies the entropy formula (7.19) for a factor graph without cycles. This extends the domain of the entropy formula in (7.19) from factor graphs without cycles to general factor graphs.

Likewise, one can extend the free energy formula in (7.20) from factor graphs without cycles to general factor graphs. This results in the *Bethe free energy* formula

$$F_B(\hat{b}) = \sum_{a \in F} \sum_{\underline{x}_{\partial a}} \hat{b}_a(\underline{x}_{\partial a}) \ln \frac{1}{f_a(\underline{x}_{\partial a})} + \sum_{i \in V} \sum_{x_i} b_i(x_i) \ln \frac{1}{f_i(x_i)} - H_B(\hat{b}). \quad (7.21)$$

Reasoning by analogy from the variational Gibbs free energy, one might hope that minimizing this function over all consistent beliefs will lead to a set of beliefs that approximates well the marginals $\mu_{\partial a}$. Hence, we define

$$\hat{b}^* = \arg \min_{\hat{b} \in \mathcal{M}'} F_B(\hat{b})$$

and note that $F_B(\hat{b}^*)$ is known as the Bethe estimate of $-\ln Z$ because $\min_{\nu \in \mathcal{P}(\mathcal{X}^n)} G(\nu) = -\ln Z$. This leads us to the question, ‘‘How hard is minimizing the Bethe free energy?’’. In the next section, we will see that this question is intimately connected to the SPA.

Remark 30. The negation of the Bethe free energy is called the *Bethe free entropy* and can similarly be maximized. An easy way to remember the difference is to recall that physical processes tend to minimize energy and maximize entropy.

7.9.4 A constrained optimization problem

Consider the Bethe variational problem

$$\min_{(\mu_i, \mu_a) \in \mathcal{L}} G_B(\mu). \quad (7.22)$$

Because \mathcal{L} is a polytope, this is a finite-dimensional constrained optimization problem, but it is typically non-convex (due to the Bethe entropy).

7.9.5 Main theorem

Theorem 31 (Bethe stationary points \iff BP fixed points). *Assume all factors are strictly positive: $f_a(x_{\partial a}) > 0$ for all $a \in F$ and all $x_{\partial a}$. Let $(\mu_i, \mu_a) \in \mathcal{L}$ be a point with full support (all entries positive). Then (μ_i, μ_a) is a stationary point of (7.22) if and only if there exist positive messages $\{m_{i \rightarrow a}\}, \{\hat{m}_{a \rightarrow i}\}$ satisfying the BP fixed-point equations (7.13)–(7.14) such that the beliefs induced by (7.15)–(7.16) equal (μ_i, μ_a) (after normalization).*

Proof. We give a complete Lagrangian derivation.

Step 1: write the constrained Lagrangian. The constraints defining \mathcal{L} are:

$$\sum_{x_i} \mu_i(x_i) = 1, \quad \forall i, \quad (7.23)$$

$$\sum_{x_{\partial a}} \mu_a(x_{\partial a}) = 1, \quad \forall a, \quad (7.24)$$

$$\sum_{x_{\partial a \setminus i}} \mu_a(x_{\partial a}) = \mu_i(x_i), \quad \forall a, \forall i \in \partial a, \forall x_i. \quad (7.25)$$

Introduce multipliers:

$$\eta_i \in \mathbb{R} \text{ for (7.23), } \eta_a \in \mathbb{R} \text{ for (7.24), } \lambda_{a \rightarrow i}(x_i) \in \mathbb{R} \text{ for (7.25).}$$

Define the Lagrangian (dropping nonnegativity since we assume interior point):

$$\begin{aligned} \mathcal{L}(\mu, \eta, \lambda) \triangleq & G_B(\mu) + \sum_i \eta_i \left(\sum_{x_i} \mu_i(x_i) - 1 \right) + \sum_a \eta_a \left(\sum_{x_{\partial a}} \mu_a(x_{\partial a}) - 1 \right) \\ & + \sum_a \sum_{i \in \partial a} \sum_{x_i} \lambda_{a \rightarrow i}(x_i) \left(\sum_{x_{\partial a \setminus i}} \mu_a(x_{\partial a}) - \mu_i(x_i) \right). \end{aligned}$$

Step 2: take derivatives w.r.t. μ_a and μ_i . We need the derivatives of entropies. For a distribution ρ , $H(\rho) = -\sum_z \rho(z) \ln \rho(z)$, hence

$$\frac{\partial}{\partial \rho(z)} (-H(\rho)) = \frac{\partial}{\partial \rho(z)} \left(\sum_{z'} \rho(z') \ln \rho(z') \right) = \ln \rho(z) + 1.$$

Therefore, differentiating \mathcal{L} yields:

Derivative w.r.t. $\mu_a(x_{\partial a})$:

$$0 = \frac{\partial \mathcal{L}}{\partial \mu_a(x_{\partial a})} = E_a(x_{\partial a}) + (\ln \mu_a(x_{\partial a}) + 1) + \eta_a + \sum_{i \in \partial a} \lambda_{a \rightarrow i}(x_i). \quad (7.26)$$

Derivative w.r.t. $\mu_i(x_i)$: Note that G_B contains $-(-(d_i-1)H(\mu_i)) = +(d_i-1) \sum_{x_i} \mu_i(x_i) \ln \mu_i(x_i)$, so

$$0 = \frac{\partial \mathcal{L}}{\partial \mu_i(x_i)} = (d_i - 1)(\ln \mu_i(x_i) + 1) + \eta_i - \sum_{a \in \partial i} \lambda_{a \rightarrow i}(x_i). \quad (7.27)$$

Step 3: solve the stationary equations into a multiplicative form. Rearrange (7.26):

$$\ln \mu_a(x_{\partial a}) = -E_a(x_{\partial a}) - 1 - \eta_a - \sum_{i \in \partial a} \lambda_{a \rightarrow i}(x_i).$$

Exponentiating and absorbing constants into a proportionality constant gives

$$\mu_a(x_{\partial a}) \propto e^{-E_a(x_{\partial a})} \prod_{i \in \partial a} e^{-\lambda_{a \rightarrow i}(x_i)} = f_a(x_{\partial a}) \prod_{i \in \partial a} u_{a \rightarrow i}(x_i), \quad (7.28)$$

where we defined $u_{a \rightarrow i}(x_i) \triangleq e^{-\lambda_{a \rightarrow i}(x_i)}$.

Similarly, rearranging (7.27) yields

$$\ln \mu_i(x_i) = -1 - \frac{\eta_i}{d_i - 1} + \frac{1}{d_i - 1} \sum_{a \in \partial i} \lambda_{a \rightarrow i}(x_i),$$

so (again absorbing constants)

$$\mu_i(x_i) \propto \prod_{a \in \partial i} e^{\lambda_{a \rightarrow i}(x_i)/(d_i-1)}. \quad (7.29)$$

At this point the expressions do not yet look exactly like BP. The next step is to reparameterize the multipliers so that the exponents disappear. This is a standard degree-normalization trick.

Step 4: reparameterize multipliers into BP-style messages. Define new functions $m_{i \rightarrow a}(x_i) > 0$ and $\hat{m}_{a \rightarrow i}(x_i) > 0$ by

$$\hat{m}_{a \rightarrow i}(x_i) \triangleq u_{a \rightarrow i}(x_i) = e^{-\lambda_{a \rightarrow i}(x_i)}, \quad m_{i \rightarrow a}(x_i) \triangleq \frac{\mu_i(x_i)}{\hat{m}_{a \rightarrow i}(x_i)}.$$

(Heuristically: the “incoming product” at node i will be proportional to μ_i , and $m_{i \rightarrow a}$ is what remains after removing the contribution from a .)

With this definition, (7.28) becomes exactly the BP belief form (7.16):

$$\mu_a(x_{\partial a}) \propto f_a(x_{\partial a}) \prod_{i \in \partial a} \hat{m}_{a \rightarrow i}(x_i) \quad \text{and} \quad \hat{m}_{a \rightarrow i}(x_i) \propto \frac{\mu_a \text{ marginalized}}{\mu_i} \quad (\text{see next step}).$$

Also, by construction,

$$\mu_i(x_i) \propto \prod_{a \in \partial i} \hat{m}_{a \rightarrow i}(x_i),$$

which matches (7.15) up to normalization.

Step 5: recover the BP update equation from local consistency. Use the local consistency constraint (7.25):

$$\mu_i(x_i) = \sum_{x_{\partial a \setminus i}} \mu_a(x_{\partial a}).$$

Substitute the multiplicative form (7.16):

$$\begin{aligned} \mu_i(x_i) &\propto \sum_{x_{\partial a \setminus i}} f_a(x_{\partial a}) \prod_{j \in \partial a} m_{j \rightarrow a}(x_j) \\ &= m_{i \rightarrow a}(x_i) \sum_{x_{\partial a \setminus i}} f_a(x_{\partial a}) \prod_{j \in \partial a \setminus i} m_{j \rightarrow a}(x_j). \end{aligned}$$

Rearranging gives

$$\frac{\mu_i(x_i)}{m_{i \rightarrow a}(x_i)} \propto \sum_{x_{\partial a \setminus i}} f_a(x_{\partial a}) \prod_{j \in \partial a \setminus i} m_{j \rightarrow a}(x_j).$$

But the left-hand side is (by definition) proportional to $\hat{m}_{a \rightarrow i}(x_i)$:

$$\hat{m}_{a \rightarrow i}(x_i) = \frac{\mu_i(x_i)}{m_{i \rightarrow a}(x_i)}.$$

Thus we have derived the factor-to-variable BP equation (7.14).

Finally, from the definition $m_{i \rightarrow a}(x_i) = \mu_i(x_i)/\hat{m}_{a \rightarrow i}(x_i)$ and $\mu_i(x_i) \propto \prod_{b \in \partial i} \hat{m}_{b \rightarrow i}(x_i)$, we obtain

$$m_{i \rightarrow a}(x_i) \propto \prod_{b \in \partial i \setminus a} \hat{m}_{b \rightarrow i}(x_i),$$

which is exactly the variable-to-factor BP equation (7.13).

Conclusion (stationary \Rightarrow BP fixed point). Starting from a stationary point of the constrained Bethe problem, we constructed positive messages satisfying the BP fixed-point equations.

Reverse direction (BP fixed point \Rightarrow stationary). Assume we have positive messages satisfying the BP fixed-point equations and define beliefs by (7.15)–(7.16). These beliefs are locally consistent and normalized, so they lie in \mathcal{L} . Now define multipliers $\lambda_{a \rightarrow i}(x_i) = -\ln \hat{m}_{a \rightarrow i}(x_i)$ and choose η_i, η_a so that (7.26)–(7.27) hold (this is always possible because the proportionality constants in (7.15)–(7.16) can be absorbed into η_i, η_a). Thus the KKT (stationarity) conditions are satisfied, so the beliefs correspond to a stationary point. \square

Remark 32. Theorem 31 concerns *stationary points*, not necessarily global minima. Because G_B can be non-convex on loopy graphs, there may be multiple BP fixed points, and the algorithm can depend on initialization and scheduling.

7.10 Connecting the three stories: LP, Bethe, and BP

7.10.1 A unifying picture

- **Exact (intractable) marginal inference:**

$$-\ln Z = \min_{\nu \in \mathcal{P}(\mathcal{X}^n)} \mathbb{E}_\nu[E(X)] - H(\nu).$$

- **Exact on trees (tractable):** rewrite the same objective in local marginals because $\mathcal{L} = \mathcal{M}$ and Bethe entropy is exact.
- **Loopy graphs (approximate):** replace $(\mathcal{M}, H(\nu))$ by $(\mathcal{L}, H(\mu)_B)$. Stationary points are BP fixed points.
- **LP for MAP:** ignore the entropy (or take a zero-temperature limit) and minimize only the average energy over a relaxation.

7.10.2 Zero temperature limit (optional detail)

Introduce temperature $T > 0$ (or inverse temperature $\beta = 1/T$) and define

$$\mu_\beta(x) = \frac{1}{Z_\beta} e^{-\beta E(x)}.$$

Then the exact variational principle becomes

$$-\frac{1}{\beta} \ln Z_\beta = \min_{\nu \in \mathcal{P}(\mathcal{X}^n)} \mathbb{E}_\nu[E(X)] - \frac{1}{\beta} H(\nu).$$

As $\beta \rightarrow \infty$, the entropy coefficient $1/\beta \rightarrow 0$, so the optimization becomes dominated by $\mathbb{E}_\nu[E(X)]$. At the level of relaxations, this is the heuristic bridge:

$$\text{LP-MAP} \approx (\text{Bethe} / \text{variational}) \text{ inference at } T \rightarrow 0.$$

We do not need this limit formally, but it can be useful to see that LP and Bethe are related approximations in the same family.