Limitations of MFT → doesn't deal well with pairwise responses

→ change free energy to

\[ \mathcal{F}_{\text{bethe}} = \sum_{ij} b_{ij}(x_i, x_j) \log b_{ij}(x_i, x_j) + \sum_i b_c(x_i) \phi_i(x_i) \]

+ Entropy Term!

Here we explicitly have pairwise terms \( b_{ij}(x_i, x_j) \).

The pseudomarginals are \( \{ b_{ij}(x_i, x_j), b_c(x_i) \} \).

Problems: (i) how do we combine the pseudomarginals together to form a consistent distribution?

(ii) how do we define the entropy?

There are two (related) solutions. Both require defining an entropy term of form:

\[ \sum_{ij} b_{ij}(x_i, x_j) \log b_{ij}(x_i, x_j) + \sum_i b_c(x_i) \log b_c(x_i) \]

Solution (1): \( \rho_{ij} = 1, \forall ij \rho_c = -(n_i - 1), \forall i \) non-sparse nodes.

This gives the Bethe Free Energy \( \Rightarrow \) non-convex

\( \Rightarrow \) Belief Propagation (BP) algorithm.

Solution (2): \( \rho_{ij} = 1, \forall ij, \& \rho_c \leq 1 \).

chosen so that the Free Energy is convex

\( \Rightarrow \) TREW Tree-Reweighted Algorithm

Note: in both cases the "entropy" is an approximation.

BP is more famous than TREW (+ variants).

But TREW-variants are arguably better – can give guaranteed convergence to global optima.
Bethe Free Energy and Belief Propagation

BP works by "message passing"

- messages \( m_{ij}(x_j) \) from node \( i \) to node \( j \) states

\[
\begin{align*}
    b_i(x_i) &= \frac{1}{Z_i} e^{-\Psi_i(x_i)} \prod_{k \neq i} m_{ki}(x_k) & \text{z.i. normalization} \\
    b_{ij}(x_i, x_j) &= \frac{1}{Z_{ij}} e^{-\Psi_{ij}(x_i, x_j)} \prod_{k \neq i, j} m_{ki}(x_k) \prod_{l \neq i, j} m_{lj}(x_l)
\end{align*}
\]

Update equation:

\[
m_{ij}^{t+1}(x_j) = \sum_{x_i} e^{-\Psi_{ij}(x_i, x_j)} \prod_{k \neq i, j} m_{ki}^{t}(x_k)
\]

Properties:

(i) Algorithm will always converge if the graph has no closed loops (tree) - like dynamic program.

(ii) The algorithm will often converge to a good approximate solution if the no. of closed loops is small.

(iii) The algorithm can converge to bad results or even fail to converge in some cases.

The algorithm looks mysterious - why the messages?

- but here is some insight into it.

First: the algorithm re-parametrizes the distribution. At any time step \( \prod_{i} b_{ij}(x_i, x_j) \propto e^{-\Psi_{ij}(x_i, x_j)} \prod_{i} m_i(x_i) \) the \( \theta \)-constant is admissibly.

Second: at a fixed point of the algorithm it can be shown that \( \prod_{i} b_{ij}(x_i, x_j) = b_i(x_i) \) \( \forall i, x_i \) the \( \theta \)-constant consisting.

The Bethe Free Energy must be supplemented by consistency constraints:

\[\sum_{x_i, x_j} \lambda_{ij}(x_j) \left( \sum_{x_i} b_{ij}(x_i, x_j) - b_i(x_j) \right)\]
It can be shown that the extrema of the Bethe free energy obey the consistency and the admissibility constraints (and vice versa).

BP obeys the admissibility constraint at all times and converges (if it does) to a solution which satisfies the consistency constraint.

By contrast, applying CCCP to Bethe gives an algorithm that maintains the consistency constraint and converges to a state that satisfies the admissibility constraint. But for Bethe, the update rule for CCCP cannot be computed analytically and requires minimizing a convex energy. This is a double loop algorithm which is slower than BP (and needs the inner loop to converge).

Where do the messages come from:

\[
\tilde{\mathbb{E}}[b;\mathbf{x}] = \mathbb{E}[b] + \sum_{j} \mathbb{E}[\lambda_{ij}(x_j) \mid x_j] \mathbb{E}[b(x_j) \mid x_j]
\]

Specify marginals in the constraints.

Go to the dual free energy

\[
\tilde{\mathcal{F}}(\mathbf{\lambda})
\]

by solving \( \frac{\partial \tilde{\mathbb{E}}[b;\mathbf{x}]}{\partial b} = 0 \) to obtain \( \tilde{\mathbb{E}}[b] \).

(can be done)

Perform dynamics in the dual space variable \( \mathbf{\lambda} \)

\( \tilde{\mathbb{E}}[b(x_j) \mid x_j] \) looks similar to messages \( \rho_{ij}(x_j) \)

Can express \( \rho_{ij}(x_j) = -\sum_{x_i} \log m_{ij}(x_i) \)

Big Problem:

If \( \tilde{\mathbb{E}}[b;\mathbf{x}] \) is convex, then \( \tilde{\mathcal{F}}(\mathbf{\lambda}) \) is concave and minimizing \( \tilde{\mathcal{F}}(\mathbf{\lambda}) \) corresponds to minimizing \( \tilde{\mathbb{E}}[b;\mathbf{x}] \).

Not true if \( \tilde{\mathbb{E}}[b;\mathbf{x}] \) is non-convex. Explain why BP can fail to converge — if graph is a tree, Bethe is convex.
The convex free energies are better behaved.

\[ \tilde{\mathcal{F}}_{b, \mathcal{J}} = 7 \{ b \} + \sum_{x_j} x_j \left( \sum_{i \neq x_j} b_{y_j}(x_j) - b_{i}(x_j) \right) \]

Again, we can solve for \( b(x) \) by setting \( \frac{\partial \mathcal{F}}{\partial b} \) this gives a concise dual

\[ \tilde{\mathcal{F}}_{\mathcal{J}} = 7 \{ b(x) \} \]

Algorithm that increases the dual free energy.

Coordinates descent will give the maximum of the dual

will correspond to the minimum of the free energy.

A growing number of algorithms—similar to BP.

But this is only a bound, the solution may not be a good estimate of the marginals except for a tree.

TCBD: Express distribution as

\[ p(x) = \frac{1}{Z} e^{\sum_{i \neq x_j} b_{y_j}(x_j)} \]

TRW: Iterate over \( \alpha \) and \( i \to j \)

1. \( m_{i \to j}(x_j) \propto \max_{x_i} \Psi_i(x_i) \Psi_j(x_j) \)

2. \( m_{i \to j}(x_j) \propto \Psi_i(x_i) \prod_{j \neq i} m_{j \to i}(x_j) \)

3. \( m_{i \to j}(x_j) \propto \Psi_j(x_j) \prod_{i \neq j} m_{i \to j}(x_i) \)

\[ \prod_{k \neq i} m_{k \to i}(x_i) \]

\[ \prod_{k \neq j} m_{k \to j}(x_j) \]

\[ \prod_{k \neq i} m_{k \to i}(x_i) \]

\[ \prod_{k \neq j} m_{k \to j}(x_j) \]

\[ \prod_{k \neq i} m_{k \to i}(x_i) \]

\[ \prod_{k \neq j} m_{k \to j}(x_j) \]
Where does the convex free energy come from?

Wainwright et al.

Express: \( P(x) = e^{T \phi(x) - T \psi(x)} \)

Define a distribution \( p(t) \) over the spanning trees of the graph.

On each tree define potential \( \Theta_t \) s.t. \( \sum_t p(t) \Theta_t = \Theta \).

Log problem:

\[
\Phi(t) = \frac{1}{T} \log \left( \frac{\sum_t p(t) \Theta_t}{T} \right) \leq \frac{1}{T} \log \left( \frac{\sum_t p(t) \Phi(t)}{T} \right)
\]

\( \Phi(.) \) convex, Jensen's inequality.

Obtain free energy by minimizing

\[
\sum_t p(t) \Phi(t) \text{ subject to constraint } \sum_t p(t) = 1.
\]

After some manipulation this yields the convex free energy

with \( \Theta = \{ \phi(x), \psi(y) \} \). Convex since each term \( p(t) \) in the summation is convex (since it is a tree).

Comment: exploits the idea that inference is easy over trees — so break down the graph into the set of all spanning trees.
Low-Temperature $\rightarrow$ MAP

\[
P(x) \rightarrow \frac{1}{Z(T)} \left( \frac{1}{P(x)} \right)^{1/2} \quad P(x) = e^{-E(x)} \quad \frac{1}{Z(T)} = e^{-E(x)}
\]

As $T \to 0$, $p(x; T)$ gets peaked about the lowest energy.

As $T \to 0$, $p(x; T)$ becomes uniform.

Calculate free energies

\[
F(B; T) = \frac{1}{T} \sum_{x} E(x) B(x) + T \text{Ent}(B(x))
\]

If $\text{Ent}(B(x))$ is convex, then $F(B; T)$ becomes convex for large $T$.

For better/convergent free energy

$F(B; T)$ relates to a linear programming problem as $T \to 0$ (current research).

Points to be made:

- For some free energies $\beta$ is a critical temperature $T_c \rightarrow$ above $T_c$ solution is trivial.
- Convexity method - deterministic annealing
- Minimize $F(B; T)$ at large $T$, use this to vitrify algorithm at smaller $T$. This heuristic works well for many problems (but no guarantees).
- Solution as $T \to 0$ can become MAP provided certain conditions apply.