Defining an equivalent optimisation problem, introduce a new variable by using the KL duality:

Local factor parameterisation returns power EP:

\[ q^*, q \]

Approximate posterior is obtained by solving a constrained minimisation problem with the following energy (with \( \sum_n \frac{1}{\alpha_n} \neq 1 \)):

\[
\min_{\tilde{p}_n} \mathcal{F}(q, \{\tilde{p}_n\}) = \left( 1 - \sum_n \frac{1}{\alpha_n} \right) \text{KL}[q||p_0] - \sum_n \frac{1}{\alpha_n} \mathbb{E}_{p_n} \left[ \log \frac{p_0(\theta)f_n(\theta)^{\alpha_n}}{\tilde{p}_n(\theta)} \right].
\]

**VFE constraints**: \( \tilde{p}_n = q, \forall n \); \( \mathcal{F}(q, \{\tilde{p}_n\}) \) simplified to \( \mathcal{F}_{\text{VFE}}(q) \)

**Power-EP constraints**: \( \mathbb{E}_q[\phi(\theta)] = \mathbb{E}_{\tilde{p}_n}[\phi(\theta)], \forall n \); (moment matching)

**(*NEW*) BB-\( \alpha \) constraints**: \( N \mathbb{E}_q[\phi(\theta)] = \sum_n \mathbb{E}_{p_n}[\phi(\theta)]; \) (moment averaging)

**(*NEW*) Mixing distributed EP and BB-\( \alpha \);**

**(*NEW*) Extensions to latent variable models.**

---

**From VFE to Power EP**

- Target distribution \( p(\theta) \propto p_0(\theta) \prod_i f_n(\theta), \) e.g., \( f_n(\theta) = p(x_n|\theta) \);
- VI minimises the variational free energy (VFE):

\[
\min_q \mathcal{F}_{\text{VFE}}(q) = \mathbb{E}_q \left[ \log \frac{q(\theta)}{p_0(\theta)} - \sum_n \log f_n(\theta) \right] = \text{KL}[q||p] - \text{const}.
\]

- An equivalent optimisation problem, subject to \( \tilde{p}_n = q, \forall n; \)

\[
\min_{\tilde{p}_n} \left( 1 - \sum_n \frac{1}{\alpha_n} \right) \text{KL}[q||p_0] - \sum_n \frac{1}{\alpha_n} \mathbb{E}_{p_n} \left[ \log \frac{p_0(\theta)f_n(\theta)^{\alpha_n}}{\tilde{p}_n(\theta)} \right].
\]

- Constraint relaxation: from \( q = \tilde{p}_n, \forall n \) to moment matching:

\[
\mathbb{E}_q[\phi(\theta)] = \mathbb{E}_{\tilde{p}_n}[\phi(\theta)], \forall n.
\]

- Introduce a new variable by using the KL duality:

\[
-\text{KL}[q||p_0] = \min_{\lambda(\theta)} -\mathbb{E}_q[\lambda(\theta)] + \log \mathbb{E}_p[\exp(\lambda(\theta))].
\]

- Write \( \lambda_{-n} \) as the Lagrange multiplier for moment matching constraints, and solve the Lagrangian:

\[
\tilde{p}_n(\theta) = \frac{1}{p_0(\theta)f_n(\theta)} \exp \left[ \lambda_{-n}^T \phi(\theta) \right],
\]

\[
\left( \sum_n \frac{1}{\alpha_n} - 1 \right) \lambda(\theta) = \sum_n \frac{1}{\alpha_n} \lambda_{-n}^T \phi(\theta) + \text{const}.
\]

- Defining \( \lambda(\theta) = \lambda_{-n}^T \phi(\theta) + \text{const} \) and substituting in the fixed point solutions, we arrive the power-EP (dual) energy:

\[
\min_{\lambda_n} \max_{(\lambda_{-n})} \left( \sum_n \frac{1}{\alpha_n} - 1 \right) \log Z_q - \sum_n \frac{1}{\alpha_n} \log Z_n,
\]

subject to \( \left( \sum_n \frac{1}{\alpha_n} - 1 \right) \lambda_n = \sum_n \frac{1}{\alpha_n} \lambda_{-n}. \)

- Approximation: \( q(\theta) = \frac{1}{p_0(\theta)} \exp \left[ \lambda_{-n}^T \phi(\theta) \right]. \)

- Local factor parameterisation returns power EP: define \( \lambda_n = (\lambda_n - \lambda_{-n})/\alpha_n \), then rewrite \( \lambda_n = \sum_j \lambda_j \) and \( \lambda_{-n} = \lambda_q - \alpha_n \lambda_n. \)

Now \( f_n(\theta) \approx \exp \left[ \lambda_n^T \phi(\theta) \right]. \)

---

**BB-\( \alpha \) & Distributed Algorithms**

**Deriving black-box alpha:**

- Power-EP proposes \( N \) sets of constraints (main reason for memory overhead);
- Idea: reduce to weighted moment averaging:

\[
\mathbb{E}_q[\phi(\theta)] = \sum_n w_n \mathbb{E}_{p_n}[\phi(\theta)], \quad \sum_n w_n = 1;
\]

- Choose \( \alpha_n = \alpha, w_n = 1/N \) and solve the Lagrangian again, we arrive at the BB-\( \alpha \) (dual) energy:

\[
\min_{\lambda_n} \left( \frac{N}{\alpha} - 1 \right) \log Z_q - \frac{1}{\alpha} \sum_n \log \int p_0(\theta)f_n(\theta)^{\alpha} \exp[\lambda_n^T \phi(\theta)] d\theta.
\]

**Distributed Power-EP algorithms:**

- In this case factor indices are divided into subsets \( N_1, N_2, ..., N_K \)
- Rewrite \( p(\theta) \propto p_0(\theta) \prod_i F_{n_i}(\theta), F_{\theta i}(\theta) = \prod_{n_i \in N_i} f_{n_i}(\theta), \)
- ...and repeat the same procedure!

- Alternatively, add extra constraints \( \tilde{p}_i = \tilde{p}_j, \forall i, j \in N_k. \)

**Mixing BB-\( \alpha \) and distributed methods:**

- Distributed BB-\( \alpha \): let \( \mathbb{E}_q[\phi(\theta)] = \frac{1}{N} \sum_k \mathbb{E}_{p_n}[\phi(\theta)]. \)

- Nesting BB-\( \alpha \) in distributed EP: let \( \mathbb{E}_q[\phi(\theta)] = \mathbb{E}_{\tilde{p}_n}[\phi(\theta)]. \)

---

**Extension: Latent Variable Models**

- Assume factorised approximation \( q(\theta, z_n) = q(\theta) \prod_n q(z_n) \):

\[
\mathcal{F}_{\text{VFE}}(q) = \mathbb{E}_q \left[ \log \frac{q(\theta)}{p_0(\theta)} + \sum_n \log \frac{q(z_n)}{p_0(z_n)} - \sum_n \log f_n(\theta, z_n) \right].
\]

- Decouple \( q \) to \( \tilde{p}_n \), similarly, and

- VI considers constraints \( \tilde{p}_n(\theta, z_n) = q(\theta)q(z_n), \forall n \); \( \forall n \)

- Different constraint relaxations return different algorithms!

- Full EP: \( \mathbb{E}_{\tilde{p}_n}[\phi(\theta), \psi(z_n)] = \mathbb{E}_q[\phi(\theta), \psi(z_n)]; \)

- Nesting EP in VI: \( \mathbb{E}_{\tilde{p}_n}[\psi(z_n)] = \mathbb{E}_q[\psi(z_n)]; \) \( \mathbb{E}_{\tilde{p}_n}[\phi(\theta)] = \tilde{p}_n(\theta); \)

- Nesting VI in EP: \( \mathbb{E}_{\tilde{p}_n}[\phi(\theta)] = \mathbb{E}_q[\phi(\theta)]; \)

- “Tilted” VMP: \( \mathbb{E}_{\tilde{p}_n}[\phi(\theta), \psi(z_n)] = \mathbb{E}_q[\phi(\theta), \psi(z_n)]; \)

- Different constraint relaxations return different algorithms!