Kernel-Based Just-In-Time Learning for Passing Expectation Propagation Messages

Wittawat Jitkrittum¹, Arthur Gretton¹, Nicolas Heess, S. M. Ali Eslami, Balaji Lakshminarayanan¹, Dino Sejdinovic², and Zoltán Szabó¹
Gatsby Computational Neuroscience Unit, University College London¹ University of Oxford²

Introduction
EP is a widely used message passing based inference algorithm.

Problem: Expensive to compute outgoing from incoming messages.

Goal: Speed up computation by a cheap regression function (message operator):

\[ \text{incoming messages} \rightarrow \text{outgoing message}. \]

Merits:
- Efficient online update of the operator during inference.
- Uncertainty monitored to invoke new training examples when needed.
- Automatic random feature representation of incoming messages.

Expectation Propagation (EP)
Under an approximation that each factor fully factorizes, an outgoing EP message \( m_{\text{out}} \) takes the form

\[
m_{\text{out}}(v_i) = \text{proj} \left( \int d\mathbf{w} \mathcal{L}(v_i | \mathbf{w}) \prod_{j \neq i} m_{\text{in}}(v_j) \right) = \text{proj} \left( f \mathcal{L}(v_i | \mathbf{w}) \prod_{j \neq i} m_{\text{in}}(v_j) \right) = \text{proj} \left( q_{\text{in}}(v_i) \right).
\]

A two-stage calculation, projecting onto the exponential family

\[
\text{proj} \left( q_{\text{in}}(v_i) \right) := \arg \min \mathbb{E}_{q_{\text{in}}(v_i)} \text{KL} \left( p_{\text{in}}(v_i) \| q_{\text{in}}(v_i) \right) \text{(projection onto exponential family)}
\]

Message Operator: Bayesian Linear Regression
Input: \( X = (x_1, \ldots, x_N) \) training incoming messages represented as random feature vectors.

Output: \( Y = (\mathbb{E}_{p_{\text{in}}(v_i)} [v_i] \cdot \mathbb{E}_{p_{\text{in}}(v_i)} [v_i]) \in \mathbb{R}^1 \times \mathbb{N} \): sufficient statistics of outgoing messages.

- Inexpensive online update.
- Bayesian regression gives prediction and predictive variance.
- If predictive variance < threshold, query importance sampling oracle.

Two-Staged Random Features
- \( \mathcal{F} : \mathbb{R}^k \rightarrow \mathbb{R}^r \): Fourier transform of \( k \), \( D_{\text{in}} \): inner features, \( D_{\text{out}} \): outer features, \( k_{\text{gaussian}} \): Gaussian kernel on \( \mathbb{R}^k \).

Out: Random features \( r_i \in \mathbb{R}^r \).
- \( \hat{f}(r_i) = \sum_{j=1}^{D_{\text{out}}} \phi(r_i \cdot s_j) \phi(r_i \cdot s_j) \); \( \phi \) is a Gaussian kernel function.

Sampling + KJIT = proposed KJIT with an importance sampling oracle.

Experiment 3: Compound Gamma Factor
Inter posterior of the precision \( \tau \) of \( x \sim N(\mu, \tau) \) from observations \( \{x_i\}_{i=1}^N \):

\[ \tau_2 \sim \text{Gamma}(\tau_2; t_1, t_1) \quad \tau \sim \text{Gamma}(t_1, t_2) \quad (t_1, t_2) = (1, 1). \]

Experiment 4: Real Data
- Binary logistic regression. Sequentially present 4 real datasets to the operator.
- Diverse distributions of incoming messages.

KJIT operator can adapt to the change of input message distributions.