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

Wittawat Jitkrittum\(^1\), Arthur Gretton\(^1\), Nicolas Heess, S. M. Ali Eslami, Balajikrishnan, Dino Sejdinovic\(^2\), and Zoltán Szabó\(^1\)

Gatsby Computational Neuroscience Unit, University College London\(^1\) University of Oxford\(^2\)

### 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 \( m_{v_i} \) takes the form

\[
 m_{v_i}(v_i) = \text{proj} \left[ \sum_{v_j \sim v_i} m_{v_j}(v_j) \right] = \frac{q_{v_i}(v_i)}{m_{v_i}(v_i)}
\]

**Message Operator:** Bayesian Linear Regression

**Input:** \( X = [x_1, \cdots, x_N] \) training incoming messages represented as random feature vectors.

**Output:** \( Y = [E_{x_1} u(v_1), \cdots, E_{x_N} u(v_N)] \in \mathbb{R}^k \times N \): 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

In: \( F(k) \): Fourier transform of \( k \), \( D_{\text{in}} \): inner features, \( D_{\text{out}} \): outer features, \( \kappa_{\text{gauss}} \): Gaussian kernel on \( \mathbb{R}^D \)

Out: Random features \( \psi(r) \in \mathbb{R}^D \)

1. Sample \( \xi_i \sim U[0, 2\pi] \)
2. \( \psi(r) = \sqrt{\frac{\pi}{\kappa_{\text{gauss}}}} \cos(r \; \xi_i) \in \mathbb{R}^D \)
3. \( \phi(v) = F(\kappa_{\text{gauss}}(\gamma)) = \sum_{r \in \Omega} \psi(r) \phi(v) \)
4. \( \hat{\phi}(v) = \frac{1}{D_{\text{out}}} \sum_{r \in \Omega} \phi(v) \)

### Experiment 1: Uncertainty Estimates

- Approximate the logistic factor: \( f(z|x) = \frac{1}{1 + \exp(-z)} \).
- Incoming messages: \( m_{v_i} = N(z_i; \mu_i, \sigma_i^2) \)
- Training set = messages collected from 20 EP runs on toy data.

### Experiment 2: Classification Errors

Fix true \( w \). Sequentially present 30 problems. Generate \( \{x_i, y_i\}_{i=1}^{30} \) for each.

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

### Experiment 3: Compound Gamma Factor

Infer posterior of the precision \( r \) of \( x \sim N(x; 0, r) \) from observations \( \{x_i, y_i\}_{i=1}^{30} \):

\[
 r_2 \sim \text{Gamma}(r_2; s_1, s_2) \quad \text{and} \quad T \sim \text{Gamma}(T; s_1, s_2) \\
 (s_1, r_2) = (1, 1). 
\]

**Inferred shape** =
- \( \text{as good as hand-crafted factor; much faster.} \)
- \( \text{as good as hand-crafted factor; much faster.} \)

### 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.