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

Wittawat Jitkrittum[1], Arthur Gretton[1], Nicolas Heess, S. M. Ali Eslami, Balaji Lakshminarayanan[1], 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):

$$\text{incoming messages} \mapsto \text{outgoing message}.$$

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.

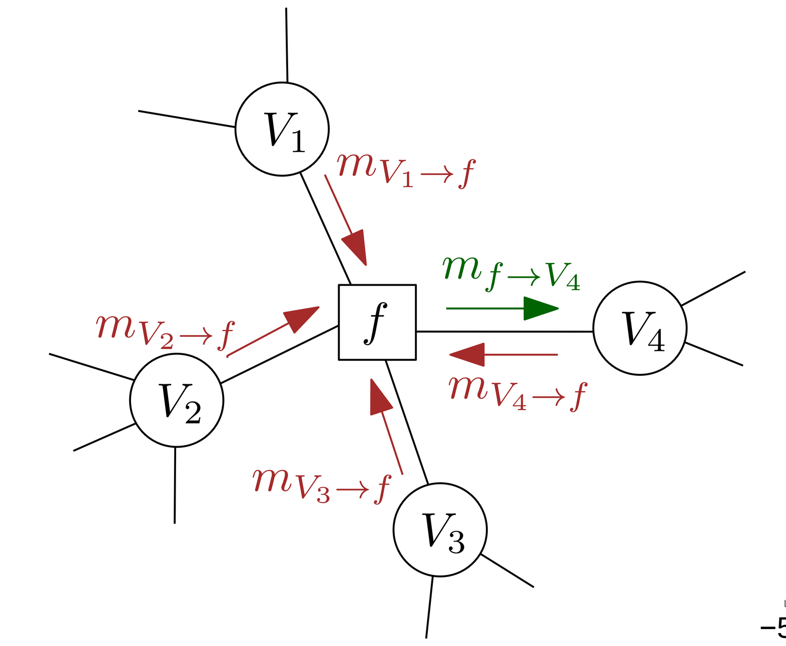
Expectation Propagation (EP)

Under an approximation that each factor fully factorizes, an outgoing EP message $m_{f\to V_i}$ takes the form

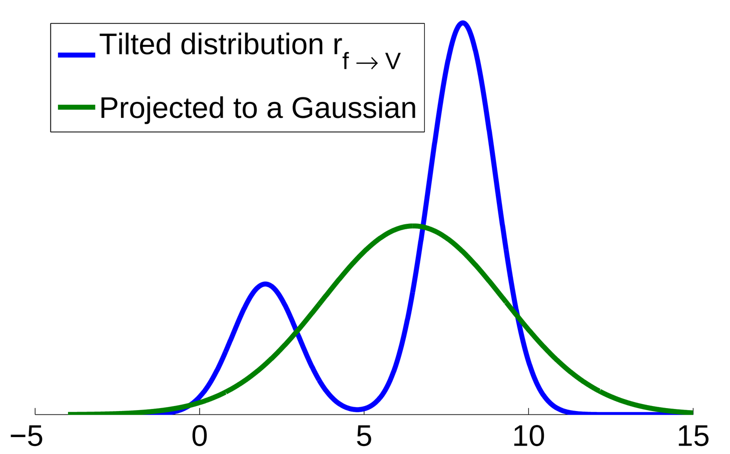
$$m_{f\to V_i}(v_i) = \frac{\text{proj}\left[\int d\mathcal{V}\backslash\{v_i\} f(\mathcal{V}) \prod_{j=1}^{c} m_{V_j\to f}(v_j)\right]}{m_{V_i\to f}(v_i)} := \frac{q_{f\to V_i}(v_i)}{m_{V_i\to f}(v_i)}$$

set of c variables connected to f — projected message — incoming message from V_j

$\text{proj}[r_{f\to V_i}] := \arg\min_{q\in\text{ExpFam}} \text{KL}\,[r_{f\to V_i} \,\|\, q]$ (projection onto exponential family)



(a) Message passing on a factor graph.



(b) Projection of $r_{f\to V}$ to a Gaussian

Projected message:

- $q_{f\to V}(v) = \text{proj}[r_{f\to V}(v)] \in$ ExpFam with sufficient statistic $u(v)$.
- Moment matching: $\mathbb{E}_{q_{f\to V}}[u(v)] = \mathbb{E}_{r_{f\to V}}[u(v)]$.

Kernel on Incoming Messages

Propose to incrementally learn during inference a kernel-based EP message operator (distribution-to-distribution regression)

$$[m_{V_j\to f}]_{j=1}^{c} \mapsto q_{f\to V_i},$$

for any factor f that can be sampled.

- Product distribution of c incoming messages: $r := \times_{l=1}^{c} r_l$, $\quad s := \times_{l=1}^{c} s_l$.
- Mean embedding of r: $\mu_r := \mathbb{E}_{a\sim r} k(\cdot, a)$.
- Gaussian kernel on (product) distributions:

$$\kappa(r,s) = \exp\left(-\frac{\|\mu_r - \mu_s\|_{\mathcal{H}}^2}{2\gamma^2}\right).$$

Two-staged random feature approximation:

$$\kappa(r,s) \overset{1^{\text{st}}}{\approx} \exp\left(-\frac{\|\hat{\phi}(r) - \hat{\phi}(s)\|_{D_{\text{in}}}^2}{2\gamma^2}\right) \overset{2^{\text{nd}}}{\approx} \hat{\psi}(r)^\top \hat{\psi}(s).$$

Message Operator: Bayesian Linear Regression

- **Input:** $X = (x_1|\cdots|x_N)$: N training incoming messages represented as random feature vectors.
- **Output:** $Y = \left(\mathbb{E}_{r^1_{f\to V}} u(v)|\cdots|\mathbb{E}_{r^N_{f\to V}} u(v)\right) \in \mathbb{R}^{D_y\times N}$: sufficient statistics of outgoing messages.
- Inexpensive online update.
- Bayesian regression gives prediction and predictive variance.
- If predictive variance $<$ threshold, query importance sampling oracle.

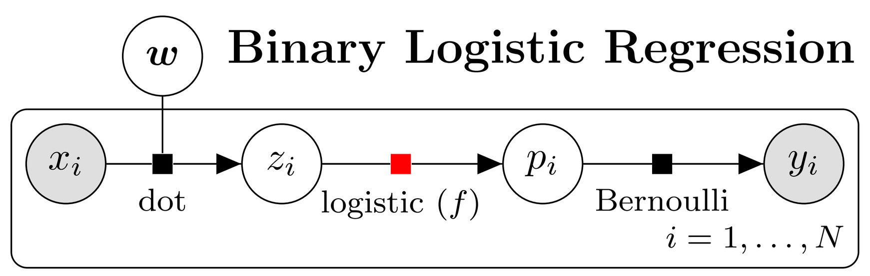
Two-Staged Random Features

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

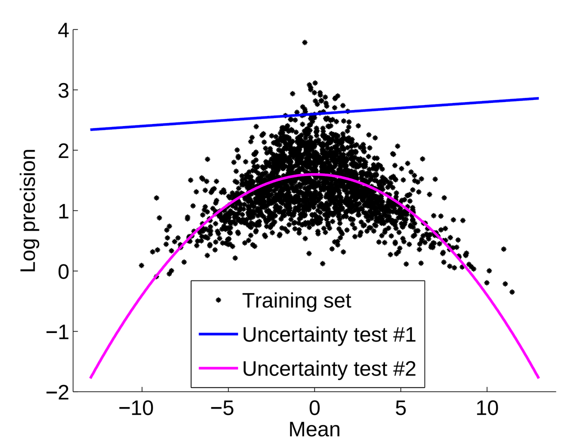
Out: Random features $\hat{\psi}(r) \in \mathbb{R}^{D_{\text{out}}}$

1. Sample $\{\omega_i\}_{i=1}^{D_{\text{in}}} \overset{\text{i.i.d}}{\sim} \mathcal{F}(k)$, $\quad \{b_i\}_{i=1}^{D_{\text{in}}} \overset{\text{i.i.d}}{\sim} U[0,2\pi]$.
2. $\hat{\phi}(r) = \sqrt{\frac{2}{D_{\text{in}}}} \left(\mathbb{E}_{x\sim r} \cos(\omega_i^\top x + b_i)\right)_{i=1}^{D_{\text{in}}} \in \mathbb{R}^{D_{\text{in}}}$
3. Sample $\{\nu_i\}_{i=1}^{D_{\text{out}}} \overset{\text{i.i.d}}{\sim} \mathcal{F}(k_{\text{gauss}}(\gamma^2))$, $\quad \{c_i\}_{i=1}^{D_{\text{out}}} \overset{\text{i.i.d}}{\sim} U[0,2\pi]$.
4. $\hat{\psi}(r) = \sqrt{\frac{2}{D_{\text{out}}}} \left(\cos(\nu_i^\top \hat{\phi}(r) + c_i)\right)_{i=1}^{D_{\text{out}}} \in \mathbb{R}^{D_{\text{out}}}$

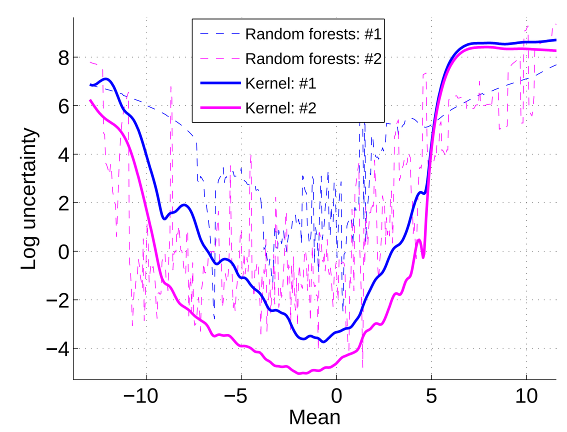
Experiment 1: Uncertainty Estimates



- Approximate the logistic factor: $f(z|x) = \delta\left(z - \frac{1}{1+\exp(-x)}\right)$.
- Incoming messages: $m_{z_i\to f} = \mathcal{N}(z_i; \mu, \sigma^2)$, $\quad m_{p_i\to f} = \text{Beta}(p_i; \alpha, \beta)$.
- Training set = messages collected from 20 EP runs on toy data.



(c) Parameters of $m_{z_i\to f}$

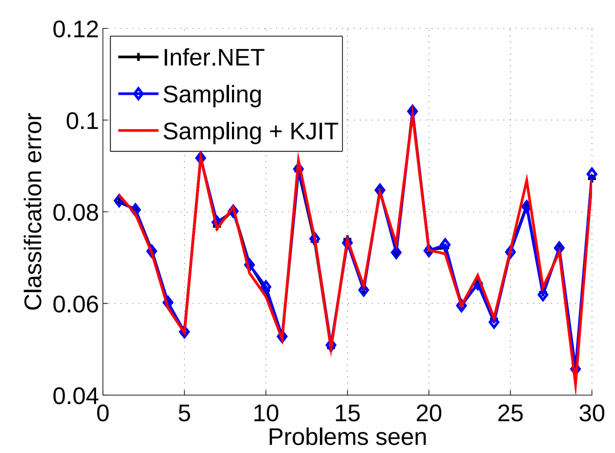


(d) Uncertainty estimates

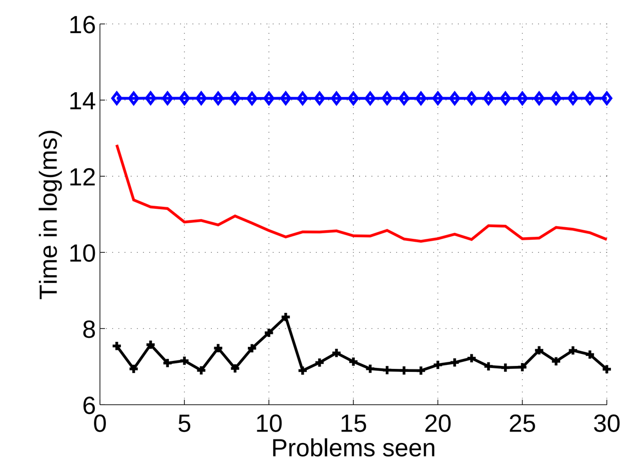
- #Random features: $D_{in} = 300$ and $D_{out} = 500$.

Experiment 2: Classification Errors

Fix true w. Sequentially present 30 problems. Generate $\{(x_i, y_i)\}_{i=1}^{300}$ for each.



(e) Test error



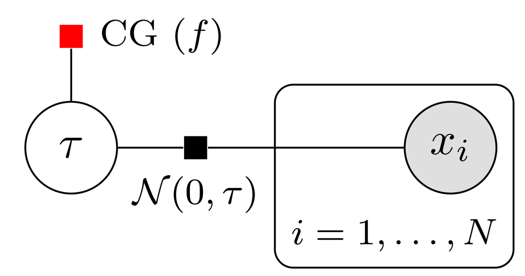
(f) Inference time

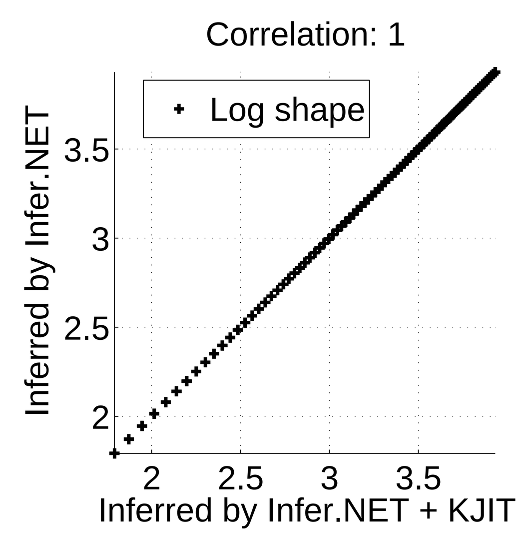
Sampling + KJIT = proposed KJIT with an importance sampling oracle.

Experiment 3: Compound Gamma Factor

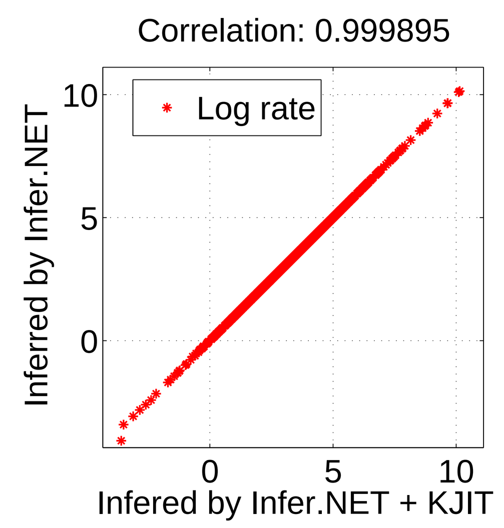
Infer posterior of the precision τ of $x \sim \mathcal{N}(x; 0, \tau)$ from observations $\{x_i\}_{i=1}^{N}$:

$$r_2 \sim \text{Gamma}(r_2; s_1, r_1)$$
$$\tau \sim \text{Gamma}(\tau; s_2, r_2)$$
$$(s_1, r_1, s_2) = (1, 1, 1).$$

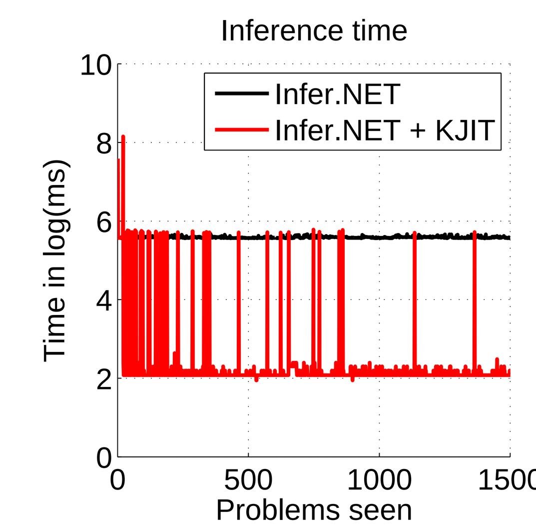




(g) Inferred shape



(h) Inferred rate

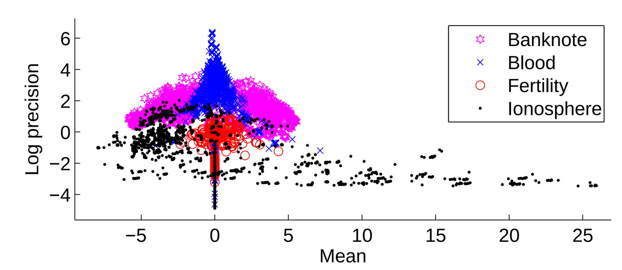


(i) Inference time

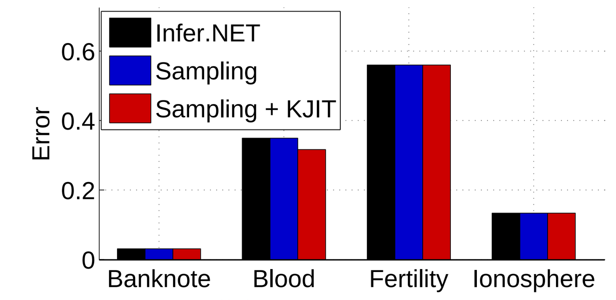
- **Infer.NET + KJIT** = proposed KJIT with a hand-crafted factor as oracle.
- **Inference quality**: as good as hand-crafted factor; much faster.

Experiment 4: Real Data

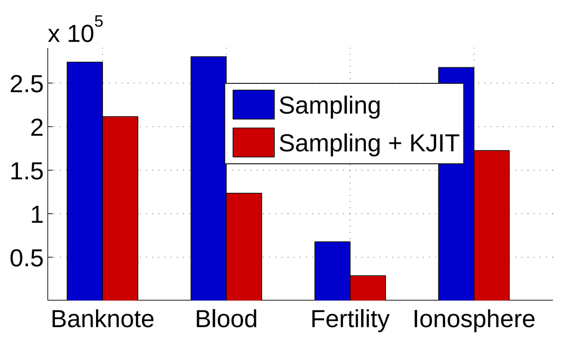
- Binary logistic regression. Sequentially present 4 real datasets to the operator.
- Diverse distributions of incoming messages.



(j) Incoming messages

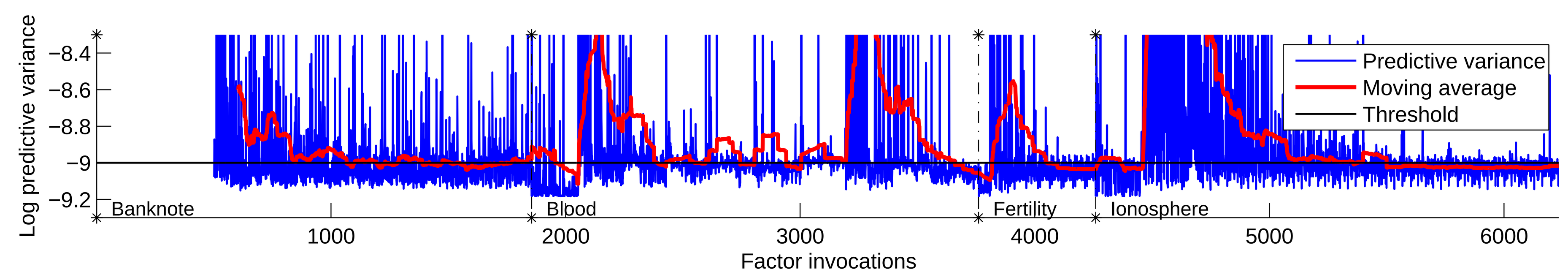


(k) Test error



(l) Inference time

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



We gratefully acknowledge the support of the Gatsby Charitable Foundation.

Email: wittawat@gatsby.ucl.ac.uk Code available at: wittawat.com