Kernel Belief Propagation

February 19, 2021

Motivation

- Difficult to derive belief propagation messages for continuous RVs with complex densities, which typically rely on easy to compute conditionals (ie conjugacy or discrete)
- Instead, rewrite messages using **nonparametric** representations of densities, i.e. sums of points in some space with no explicit parameters
- Approach extends to any domain on which kernels can be defined, such as strings and graphs

Learning in Markov Random Fields

- Pairwise MRF (typically parameterize log potentials)

$$\mathbb{P}(X) \propto \prod_{s,t \in \mathcal{E}} \Psi_{st}(X_s, X_t) \prod_{s \in \mathcal{V}} \Psi(X_s).$$

- Estimate gradients wrt log potentials by computing edge and node marginals via inference, ie the beliefs $\mathbb{B}(X_s, X_t)$ and $\mathbb{B}(X_s)$

- Belief propagation is an algorithm for performing inference

Belief Prop (BP)

- BP propagates messages from nodes to neighbours iteratively until convergence
- Messages from t to s

$$m_{ts}(X_s) = \int_{X_t \in \mathcal{X}} \Psi_{st}(X_s, X_t)\Psi_t(X_t) \prod_{u \in \delta(t) \setminus \{s\}} m_{ut}(X_t) dX_t$$

- Belief at s

$$\mathbb{B}(X_s) = \Psi_s(X_s) \prod_{t \in \delta(s)} m^*_{ts}(X_s),$$

with fixed point messages m^*

BP

- The integrals in the messages may be difficult to compute

- Solution: Rewrite messages as an expectation, then approximate conditional

$$
\begin{aligned}
m_{ts}(X_s) &= \int_{\mathcal{X}} \mathbb{P}^*(X_t \mid X_s) \prod_{u \in \delta(t) \setminus \{s\}} m_{ut}(X_t) dX_t \\
&= \mathbb{E}_{X_t \mid X_s}\left[\prod_{u \in \delta(t) \setminus \{s\}} m_{ut}(X_t) \right]
\end{aligned}
$$

- Requires fully observed model, otherwise stuck with original integral

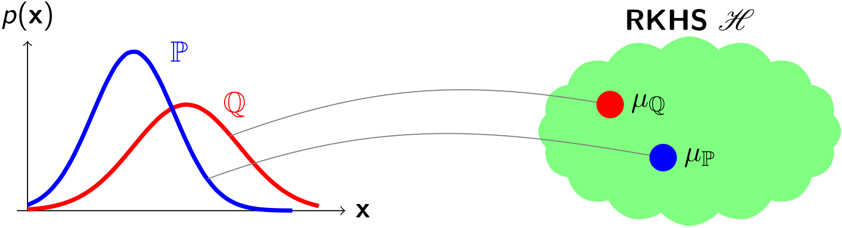
Nonparametric BP Baselines

- Nonparametric BP requires a 2-step process of estimating conditional $\mathbb{P}^*(X_t \mid X_s)$, then computing messages
- NPBP baselines are Gaussian Mixture BP (Sudderth et al, 2003) and Particle BP (Ihler and McAllester, 2009)
- Kernel BP reduces this to a single step of matrix-vector products

High Level Overview of Kernel Belief Propagation

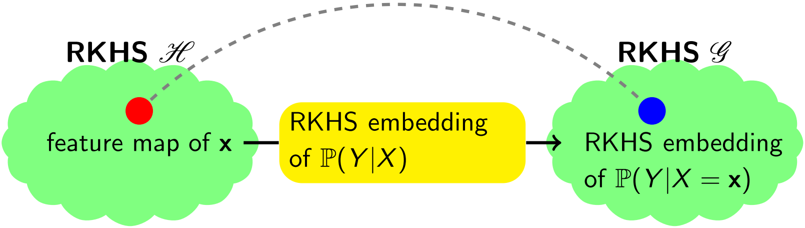
- Embed messages in RKHS
- Approximate expectations via observed samples
- Compute messages with inner products

Kernel Mean Embedding



- Kernel mean embeddings map distributions into Hilbert spaces
- Can approximate embedding in RKHS via sampling

Conditional Distribution Embedding



- Embed conditional probability function as an operator

Why the focus on nonparametric?

- The sell is that this works as an approximation for inference in models where the messages are difficult to derive, i.e. complex distributions

- Kernel mean embeddings also apply to messages that are easy to derive, but we may want to approximate for computational benefits

Definitions

- Domain $\mathcal{X}$
- Hilbert space $\mathscr{H}$ of functions on $\mathcal{X} \mapsto \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle$, kernel K, and feature map ϕ
- The point evaluation property, ie that function evaluation is an inner product,

$$\langle f, K(x, \cdot) \rangle = f(x),$$

implies the reproducing property:

$$\langle K(x, \cdot), K(y, \cdot) \rangle = K(x, y) = \langle \phi(x), \phi(y) \rangle$$

Why is reproducing property needed

- ???

Theorem Notes

- Riesz representation theorem: If operator $\mathcal{A} : \mathscr{H} \to \mathbb{R}$ is bounded, then there exists a representer $g_{\mathcal{A}} \in \mathscr{H}$ st
$$A[f] = \langle f, g_{\mathcal{A}} \rangle, \forall f \in \mathscr{H}.$$
- Point evaluation property: In an RKHS, consider the evaluation functional $\mathcal{F}_{\mathbf{x}}(f) = f(\mathbf{x})$. Riesz representation theorem tells us there exists a representer $k_{\mathbf{x}} : \mathscr{H} \to \mathbb{R}$ st
$$\mathcal{F}_{\mathbf{x}}(f) = \langle f, k_{\mathbf{x}} \rangle = f(\mathbf{x}),$$
referred to as the reproducing kernel for the point $\mathbf{x}$.
- The reproducing property is a special case of the point evaluation property. Consider the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and define $f(\mathbf{x}) = k(\mathbf{y}, \mathbf{x})$ for all $\mathbf{y} \in \mathcal{X}$. Applying the point evaluation property yields
$$f(\mathbf{x}) = \langle k(\mathbf{x}, \cdot), k(\mathbf{y}, \cdot) \rangle,$$
where $k(\mathbf{x}, \cdot)$ is the canonical feature map denoted by $\phi : \mathcal{X} \to \mathscr{H}$.
- Alternatively you can start by assuming the kernel is positive

Refs