Learning with probabilities as inputs, using kernels

Arthur Gretton

Gatsby Computational Neuroscience Unit

NIPS workshop on Probabilistic Numerics, 2015
Motivating example: Expectation Propagation

\[
m_{f \rightarrow V_i}(v_i) = \text{proj} \left[ \int dV \prod_{j=1}^{c} m_{V_j \rightarrow f}(v_j) \right] = q_{f \rightarrow V_i}(v_i)
\]

set of \(c\) variables connected to \(f\)

incoming message from \(V_j\)

projected message

(projection onto exponential family)
Motivating example: Expectation Propagation

A set of \( c \) variables connected to \( f \)

Projected message

\[
\begin{align*}
    m_{f \rightarrow V_i}(v_i) &= \text{proj} \left( \int dV \setminus \{v_i\} f(V) \prod_{j=1}^{c} m_{V_j \rightarrow f}(v_j) \right) \\
    &= \text{proj} \left( \prod_{j=1}^{c} m_{V_j \rightarrow f}(v_j) \right) := q_{f \rightarrow V_i}(v_i)
\end{align*}
\]

Incoming message from \( V_j \)

\[
\text{proj}[r_{f \rightarrow V_i}] := \arg\min_{q \in \text{ExpFam}} \text{KL}[r_{f \rightarrow V_i} \parallel q]
\]

(projection onto exponential family)

**Expensive integral** (besides special cases).

**Goal:** Learn an *uncertainty aware* message operator (regression function)

\[
[m_{V_j \rightarrow f}]_{j=1}^{c} \leftrightarrow q_{f \rightarrow V_i}.
\]

**Challenges:** dealing with *huge sample size*, knowing when to consult expensive oracle.
Overview

- **Introduction to reproducing kernel Hilbert spaces**
  - Kernels and feature spaces
  - Mapping probabilities to feature space

- **Learning with distribution-valued inputs**
  - Learning rates achievable when samples from distributions available
    
    [AISTATS15, JMLR in revision]
  
  - Approximate, uncertainty-aware regression with application to EP
    
    [UAI15]
  
  - Learning to predict direction of causality
    
    [Lopez-Paz et al., 2015]

- **Learning with distribution-valued outputs** (not this talk)
Kernels: similarity between features

- We have two objects $x$ and $x'$ from a set $\mathcal{X}$ (documents, images, ...). How similar are they?
Kernels: similarity between features

- We have two objects $x$ and $x'$ from a set $\mathcal{X}$ (documents, images, ...).
  How similar are they?
- Define **features** of objects:
  - $\varphi_x \in \mathcal{F}$ are features of $x$,
  - $\varphi_{x'} \in \mathcal{F}$ are features of $x'$
- A **kernel** is the dot product between these features:
  \[
k(x, x') := \langle \varphi_x, \varphi_{x'} \rangle_{\mathcal{F}} = \sum_{j \in J} \varphi_x^{(j)} \varphi_{x'}^{(j)}
  \]
- A **function** in the RKHS $\mathcal{F}$ is a linear combination of features,
  \[
f(x) = \langle f, \varphi_x \rangle_{\mathcal{F}} = \sum_{j \in J} f_j \varphi_x^{(j)} \quad f \in \ell_2(J)
  \]
Infinite dimensional feature space

Squared exponential kernel: \( k(x, x') = \exp \left( - \frac{\|x - x'\|^2}{2\sigma^2} \right) \)
Infinite dimensional feature space

Squared exponential kernel: \( k(x, x') = \exp \left( -\frac{||x - x'||^2}{2\sigma^2} \right) \)

\[ \lambda_j \propto b^j \quad b < 1 \]

\[ e_j(x) \propto \exp(-(c - a)x^2)H_j(x\sqrt{2c}), \]

\( a, b, c \) are functions of \( \sigma \), and \( H_j \) is \( j \)th order Hermite polynomial.
The kernel trick

Example RKHS function, squared exponential kernel:

\[ f(x) := \sum_{j=1}^{\infty} f_j \varphi_j(x) \]
The kernel trick

Example RKHS function, squared exponential kernel:

\[ f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x) \]
The kernel trick

Example RKHS function, squared exponential kernel:

\[ f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x) = \sum_{i=1}^{m} \alpha_i \left[ \sum_{j=1}^{\infty} \varphi^{(j)}(x_i) \varphi^{(j)}(x) \right] = \sum_{j=1}^{\infty} f_j \varphi^{(j)} \]

where \( f_j = \sum_{i=1}^{m} \alpha_i \varphi^{(j)}_{x_i} \)
Probabilities in feature space: the mean trick

The kernel trick

- Given $x \in \mathcal{X}$ for some set $\mathcal{X}$, define feature map $\varphi_x \in \mathcal{F}$,
  
  \[ \varphi_x = \left[ \ldots \varphi^{(j)}_x \ldots \right] \in \ell_2 \]

- For positive definite $k(x, x')$,
  
  \[ k(x, x') = \langle \varphi_x, \varphi_{x'} \rangle_{\mathcal{F}} \]

- Function in the RKHS:
  \[ \forall f \in \mathcal{F}, \]
  \[ f(x) = \langle f, \varphi_x \rangle_{\mathcal{F}} \]
Probabilities in feature space: the mean trick

The kernel trick

- Given $x \in \mathcal{X}$ for some set $\mathcal{X}$, define feature map $\varphi_x \in \mathcal{F}$,
  
  $$\varphi_x = \left[ \ldots \varphi_x^{(j)} \ldots \right] \in \ell_2$$

- For positive definite $k(x, x')$,
  
  $$k(x, x') = \langle \varphi_x, \varphi_{x'} \rangle_{\mathcal{F}}$$

- Function in the RKHS:
  
  $$\forall f \in \mathcal{F},$$
  
  $$f(x) = \langle f, \varphi_x \rangle_{\mathcal{F}}$$

The mean trick

- Given $\mathbf{P}$ a Borel probability measure on $\mathcal{X}$, define mean embedding $\mu_\mathbf{P} \in \mathcal{F}$

  $$\mu_\mathbf{P} = \left[ \ldots \mathbb{E}_\mathbf{P} \left[ \varphi_X^{(j)} \right] \ldots \right] \in \ell_2(J)$$

- For positive definite $k(x, x')$,

  $$\mathbb{E}_{\mathbf{P}, \mathbf{Q}} k(X, Y) = \langle \mu_\mathbf{P}, \mu_\mathbf{Q} \rangle_{\mathcal{F}}$$

  for $X \sim \mathbf{P}$ and $Y \sim \mathbf{Q}$.

Need to ensure Bochner integrability of $\varphi_x$ for $x \sim \mathbf{P}$

- $\mathbb{E}_\mathbf{P}(f(X)) =: \langle \mu_\mathbf{P}, f \rangle_{\mathcal{F}}$
Kernels on distributions in supervised learning

- Kernels have been very widely used in supervised learning
  - Support vector classification/regression, kernel ridge regression . . .
Kernels on distributions in supervised learning

- Kernels have been very widely used in supervised learning
- Simple kernel on distributions (population counterpart of set kernel)
  \[ K(P, Q) = \langle \mu_P, \mu_Q \rangle_F \]
- Squared distance between distribution embeddings (MMD)
  \[ \text{MMD}^2(\mu_P, \mu_Q) := \| \mu_P - \mu_Q \|_F^2 = E_P k(x, x') + E_Q k(y, y') - 2 E_{P, Q} k(x, y) \]
Kernels on distributions in supervised learning

- Kernels have been very widely used in supervised learning
- Simple kernel on distributions (population counterpart of set kernel) 
  \[ K(P, Q) = \langle \mu_P, \mu_Q \rangle_F \]
- Can define kernels on mean embedding features

<table>
<thead>
<tr>
<th>$K_G$</th>
<th>$K_e$</th>
<th>$K_C$</th>
<th>$K_t$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e^{-\frac{|\mu_P - \mu_Q|^2}{2\theta^2}}$</td>
<td>$e^{-\frac{|\mu_P - \mu_Q|^2}{2\theta^2}}$</td>
<td>$(1 + |\mu_P - \mu_Q|^2_F / \theta^2)^{-1}$</td>
<td>$(1 + |\mu_P - \mu_Q|^\theta_F)^{-1}$, $\theta \leq 2$</td>
<td>...</td>
</tr>
</tbody>
</table>

$\|\mu_P - \mu_Q\|^2_F = \mathbf{E}_P k(x, x') + \mathbf{E}_Q k(y, y') - 2\mathbf{E}_{P,Q} k(x, y)$
Expectation Propagation

- **Expensive integral** (besides special cases)
- **Goal:** Learn an *uncertainty aware* message operator (regression function)
  \[
  [m_{V_j \rightarrow f}]^c_{j=1} \mapsto q_{f \rightarrow V_i}.
  \]
- **Challenges:** dealing with *huge sample size*, knowing when to consult *expensive oracle*. 

**Diagram:**
- Set of \( c \) variables connected to \( f \)
- Projected message
  \[
  \begin{align*}
  m_{f \rightarrow V_i}(v_i) &= \frac{\text{proj} \left[ \int dV \setminus \{v_i\} f(V) \prod_{j=1}^c m_{V_j \rightarrow f}(v_j) \right]}{m_{V_i \rightarrow f}(v_i)} \\
  q_{f \rightarrow V_i}(v_i) &= \text{arg min}_{q \in \text{ExpFam}} KL [r_{f \rightarrow V_i} \| q]
  \end{align*}
  \]
- Incoming message from \( V_j \)
  \[ \text{proj}[r_{f \rightarrow V_i}] := \text{arg min}_{q \in \text{ExpFam}} KL [r_{f \rightarrow V_i} \| q] \] (projection onto exponential family)
Distribution regression using random Fourier features

Kernel representation by random Fourier features \cite{Rahimi and Recht, 2008}

- **Bochner’s theorem:** Continuous, translation-invariant kernel
  
  $k(a, b) = k(a - b)$ on $\mathbb{R}^m$ positive definite iff \exists prob. meas. $\mathcal{K}(\omega)$

  
  $$k(a - b) = \mathbb{E}_{\omega \sim \mathcal{K}} \mathbb{E}_{c \sim U[0, 2\pi]} \left[ 2 \cos(\omega^\top a + c) \cos(\omega^\top b + c) \right]$$
Kernel representation by random Fourier features \cite{Rahimi and Recht, 2008}

- **Bochner’s theorem:** Continuous, translation-invariant kernel $k(a, b) = k(a - b)$ on $\mathbb{R}^m$ positive definite iff $\exists$ prob. meas. $\mathcal{K}(\omega)$

$$k(a - b) = \mathbf{E}_{\omega \sim \mathcal{K}} \mathbf{E}_{c \sim U[0, 2\pi]} \left[ 2 \cos(\omega^\top a + c) \cos(\omega^\top b + c) \right]$$

- **Random features:** $\varphi_d(a) \in \mathbb{R}^d$ such that

$$k(a - b) \approx \varphi_d(a)^\top \varphi_d(b)$$

1. Draw i.i.d. $\{\omega_i\}_{i=1}^d \sim \mathcal{K}(\omega)$.
2. Draw i.i.d. $\{c_i\}_{i=1}^d \sim U[0, 2\pi]$
3. $\varphi_d(a) = \sqrt{\frac{2}{d}} \left[ \cos \left( \omega_1^\top a + c_1 \right), \ldots, \cos \left( \omega_d^\top a + c_d \right) \right]^\top \in \mathbb{R}^d$
Distribution regression using random Fourier features

- Given incoming messages $P := m_{V_i \rightarrow f}$ and $Q := m_{V_j \rightarrow f}$
- Approximate random Fourier mean embeddings:

$$\mu_{P,d} := \mathbb{E}_{x \sim P} [\varphi_d(x)]$$
Distribution regression using random Fourier features

- Given incoming messages $P := m_{V ightarrow f}$ and $Q := m_{V_j ightarrow f}$
- Approximate random Fourier mean embeddings:
  $$\mu_{P,d} := \mathbb{E}_{x \sim P}[\varphi_d(x)]$$
- Approximate embeddings for kernel $K$ on $\mu_P \in \mathbb{R}^{d'}$:
  $$K_G(\mu_P, \mu_Q) \approx \exp\left(-\frac{\|\mu_{P,d} - \mu_{Q,d}\|_2^2}{2\gamma^2}\right) \approx \psi_{d'}(P) \top \psi_{d'}(Q).$$

- Gaussian process regression directly on features $\psi_{d'}(P) \in \mathbb{R}^{d'}$ [UAI15]
  - Bayesian uncertainty estimates tell us when to consult oracle
  - Efficient rank-1 updates, solution size constant as number of samples increases
Expectation Propagation for Classification

- Sequentially present 4 real datasets to the operator to learn.
- If predictive variance $> \text{threshold}$, ask oracle.

- **Left:** Binary classification error with learned posterior $\mathbf{w}$,
  **Right:** EP runtime.

\[ w \quad \text{Binary Logistic Regression} \]

\[ x_i \xrightarrow{\text{dot}} z_i \xrightarrow{\text{logistic}(f)} p_i \xrightarrow{\text{Bernoulli}} y_i \quad \text{for } i = 1, \ldots, N \]
• Initial silent period = parameter selection + mini-batch training.

• * = start of a new problem.

• Sharp rises after * indicates ability to detect distribution (problem) change.

\[
m_{z \rightarrow f} = \text{Gaussian}(z)\]
Regression using *population* mean embeddings

- **Samples** $z := \{(\mu_{P_i}, y_i)\}_{i=1}^{\ell} \sim \text{i.i.d.} \rho(\mu_P, y) = \rho(y|\mu_P)\rho(\mu_P)$,

  $$\mu_{P_i} = \mathbb{E}_{P_i}[\varphi_x]$$

- **Regression function**

  $$f_\rho(\mu_P) = \int_{\mathbb{R}} y d\rho(y|\mu_P),$$
Regression using *population* mean embeddings

- **Samples** \( z := \{(\mu_{p_i}, y_i)\}_{i=1}^{\ell} \) \( \sim \) \( \rho(\mu_{p}, y) = \rho(y|\mu_{p})\rho(\mu_{p}) \),

\[ \mu_{p_i} = \mathbb{E}_{p_i}[\varphi_x] \]

- **Regression function**

\[ f_{\rho}(\mu_{p}) = \int_{\mathbb{R}} y d\rho(y|\mu_{p}) \]

- **Ridge regression** for labelled distributions

\[ f_{z}^{\lambda} = \arg\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} (f(\mu_{p_i}) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2 , \quad (\lambda > 0) \]

- **Define RKHS** \( \mathcal{H} \) with kernel \( K(\mu_{p}, \mu_{q}) := \langle \psi_{\mu_{p}}, \psi_{\mu_{q}} \rangle_{\mathcal{H}} \):

functions from \( F \subset \mathcal{F} \) to \( \mathbb{R} \), where

\( F := \{\mu_{p} : p \in \mathcal{P}\} \) \quad \( \mathcal{P} \) set of prob. meas. on \( \mathcal{X} \)
Regression using *population* mean embeddings

- Expected risk, Excess risk

\[ \mathcal{R}[f] = \mathbb{E}_{\rho(\mu_P,y)} (f(\mu_P) - y)^2 \quad \mathcal{E}(f^\lambda_z, f_\rho) = \mathcal{R}[f^\lambda_z] - \mathcal{R}[f_\rho]. \]

- Minimax rate \[ \text{Caponnetto and Vito, 2007} \]

\[ \mathcal{E}(f^\lambda_z, f_\rho) = \mathcal{O}_p \left( \ell^{- \frac{bc}{bc+1}} \right) \quad (1 < b, c \in (1, 2]). \]

- \( b \) size of input space, \( c \) smoothness of \( f_\rho \)
Regression using *population* mean embeddings

- Expected risk, Excess risk

\[
\mathcal{R}[f] = \mathbb{E}_{\rho(\mu_P,y)} (f(\mu_P) - y)^2 \quad \mathcal{E}(f^\lambda_z, f_{\rho}) = \mathcal{R}[f^\lambda_z] - \mathcal{R}[f_{\rho}].
\]

- Minimax rate [Caponnetto and Vito, 2007]

\[
\mathcal{E}(f^\lambda_z, f_{\rho}) = \mathcal{O}_p \left( \ell^{-\frac{bc}{bc+1}} \right) \quad (1 < b, c \in (1, 2]).
\]

  - \( b \) size of input space, \( c \) smoothness of \( f_{\rho} \)

- Replace \( \mu_P_i \) with \( \hat{\mu}_P_i = N^{-1} \sum_{j=1}^{N} \varphi x_j \), \( x_j \) i.i.d. \( P_i \)

- Given \( N = \ell^a \log(\ell) \) and \( a = 2 \), (and Hölder condition on \( \psi : F \to \mathcal{H} \))

\[
\mathcal{E}(f^\lambda_{\hat{z}}, f_{\rho}) = \mathcal{O}_p \left( \ell^{-\frac{bc}{bc+1}} \right) \quad (1 < b, c \in (1, 2]).
\]

Same rate as for population \( \mu_P \) embeddings! [AISTATS15, JMLR in revision]
Learning causal direction with mean embeddings

Additive noise model to direct an edge between random variables $x$ and $y$ [Hoyer et al., 2009]

Figure: D. Lopez-Paz
Learning causal direction with mean embeddings

Classification of cause-effect relations [Lopez-Paz et al., 2015]

- **Tuebingen cause-effect pairs**: 82 scalar real-world examples where causes and effects known [Zscheischler, J., 2014]
- **Training data**: artificial, random nonlinear functions with additive gaussian noise.
- **Features**: \( \hat{\mu}_P, \hat{\mu}_P, \hat{\mu}_{P_{xy}} \)
  with labels for \( x \rightarrow y \) and \( y \rightarrow x \)
- **Performance**: 81% correct

Figure: Mooij et al. (2015)
Overview

- **Introduction to reproducing kernel Hilbert spaces**
  - Kernels and feature spaces
  - Mapping *probabilities* to feature space

- **Learning with distribution-valued inputs**
  - *Learning rates* achievable when samples from distributions available
    - [AISTATS15, JMLR in revision]
  - Approximate, uncertainty-aware regression with application to EP
    - [UAI15]
  - Learning to predict *direction of causality* [Lopez-Paz et al., 2015]
Co-authors

- **From UCL:**
  - Steffen Grunewalder
  - Wittawat Jitkrittum
  - Guy Lever
  - Zoltan Szabo

- **External:**
  - Ali Eslami, Deepmind
  - Kenji Fukumizu, ISM
  - Nicolas Heess, Deepmind
  - Barnabas Poczos, CMU
  - Bernhard Schoelkopf, MPI
  - Dino Sejdinovic, Oxford
  - Alex Smola, Google/CMU
  - Le Song, Georgia Tech
  - Bharath Sriperumbudur, Penn. State
Learning when the outputs are distributions
Motivating example: Bayesian inference without a model

Challenges:

- No parametric model of camera dynamics (only samples)
- No parametric model of map from camera angle to image (only samples)
- Want to do filtering: Bayesian inference
Conditional distribution embedding

Bayes rule:

\[ P(y|x) = \frac{P(x|y)\pi(y)}{\int P(x|y)\pi(y)dy} \]

- \( P(x|y) \) is likelihood
- \( \pi \) is prior

How would this look with kernel embeddings?
Bayes rule:

\[
P(y|x) = \frac{P(x|y)\pi(y)}{\int P(x|y)\pi(y)dy}
\]

- \(P(x|y)\) is likelihood
- \(\pi\) is prior

How would this look with kernel embeddings?

Define RKHS \(\mathcal{G}\) on \(\mathcal{Y}\) with feature map \(\psi_y\) and kernel \(l(y, \cdot)\)

We need a **conditional mean embedding**: for all \(g \in \mathcal{G}\),

\[
E_{Y|x^*}g(Y) = \langle g, \mu_{P(y|x^*)}\rangle_{\mathcal{G}}
\]

This will be obtained by **RKHS-valued ridge regression**
Ridge regression from $\mathcal{X} := \mathbb{R}^d$ to a finite vector output $\mathcal{Y} := \mathbb{R}^{d'}$ (these could be $d'$ nonlinear features of $y$):
Define training data

$$X = \begin{bmatrix} x_1 & \ldots & x_m \end{bmatrix} \in \mathbb{R}^{d \times m} \quad Y = \begin{bmatrix} y_1 & \ldots & y_m \end{bmatrix} \in \mathbb{R}^{d' \times m}$$
Ridge regression and the conditional feature mean

Ridge regression from \( \mathcal{X} := \mathbb{R}^d \) to a finite vector output \( \mathcal{Y} := \mathbb{R}^{d'} \) (these could be \( d' \) nonlinear features of \( y \)):

Define training data

\[
X = \begin{bmatrix} x_1 & \ldots & x_m \end{bmatrix} \in \mathbb{R}^{d \times m} \quad Y = \begin{bmatrix} y_1 & \ldots & y_m \end{bmatrix} \in \mathbb{R}^{d' \times m}
\]

Solve

\[
\tilde{A} = \arg \min_{A \in \mathbb{R}^{d' \times d}} \left( \|Y - AX\|^2 + \lambda \|A\|_{HS}^2 \right),
\]

where

\[
\|A\|_{HS}^2 = \text{tr}(A^\top A) = \min\{d, d'\} \sum_{i=1} \gamma_{A,i}^2
\]
Ridge regression from $\mathcal{X} := \mathbb{R}^d$ to a finite vector output $\mathcal{Y} := \mathbb{R}^{d'}$ (these could be $d'$ nonlinear features of $y$):

Define training data

$$X = \begin{bmatrix} x_1 & \ldots & x_m \end{bmatrix} \in \mathbb{R}^{d \times m} \quad Y = \begin{bmatrix} y_1 & \ldots & y_m \end{bmatrix} \in \mathbb{R}^{d' \times m}$$

Solve

$$\tilde{A} = \arg \min_{A \in \mathbb{R}^{d' \times d}} \left( \|Y - AX\|^2 + \lambda \|A\|^2_{HS} \right),$$

where

$$\|A\|^2_{HS} = \text{tr}(A^\top A) = \min\{d,d'\} \sum_{i=1} \gamma_{A,i}^2$$

**Solution:** $\tilde{A} = C_{YX} (C_{XX} + m\lambda I)^{-1}$
Prediction at new point $x$:

$$y^* = \bar{A}x$$

$$= C_{YX} (C_{XX} + m\lambda I)^{-1} x$$

$$= \sum_{i=1}^{m} \beta_i(x)y_i$$

where

$$\beta_i(x) = (K + \lambda mI)^{-1} \begin{bmatrix} k(x_1, x) & \cdots & k(x_m, x) \end{bmatrix}^\top$$

and

$$K := X^\top X \quad k(x_1, x) = x_1^\top x$$
Ridge regression and the conditional feature mean

Prediction at new point $x$:

\[
y^* = \tilde{A} x \\
= C_{YX} (C_{XX} + m\lambda I)^{-1} x \\
= \sum_{i=1}^{m} \beta_i(x) y_i
\]

where

\[
\beta_i(x) = (K + \lambda m I)^{-1} \begin{bmatrix} k(x_1, x) & \ldots & k(x_m, x) \end{bmatrix}^\top
\]

and

\[
K := X^\top X \qquad k(x_1, x) = x_1^\top x
\]

What if we do everything in kernel space?
Ridge regression and the conditional feature mean

Recall our setup:

- Given training \textit{pairs}:
  \[(x_i, y_i) \sim \mathbb{P}_{XY}\]

- \(\mathcal{F}\) on \(\mathcal{X}\) with feature map \(\varphi_x\) and kernel \(k(x, \cdot)\)

- \(\mathcal{G}\) on \(\mathcal{Y}\) with feature map \(\psi_y\) and kernel \(l(y, \cdot)\)

We define the \textbf{covariance between feature maps}:

\[C_{XX} = \mathbb{E}_X (\varphi_X \otimes \varphi_X) \quad C_{XY} = \mathbb{E}_{XY} (\varphi_X \otimes \psi_Y)\]

and matrices of \textbf{feature mapped training data}

\[
X = \begin{bmatrix}
    \varphi_{x_1} & \cdots & \varphi_{x_m}
\end{bmatrix} \quad Y := \begin{bmatrix}
    \psi_{y_1} & \cdots & \psi_{y_m}
\end{bmatrix}
\]
Objective: [Weston et al. (2003), Micchelli and Pontil (2005), Caponnetto and De Vito (2007), ICML12, ICML13 ]

\[
\hat{A} = \arg \min_{A \in \text{HS}(\mathcal{F}, \mathcal{G})} \left( E_{XY} \| Y - AX \|^2_G + \lambda \| A \|^2_{HS} \right), \quad \| A \|^2_{HS} = \sum_{i=1}^{\infty} \gamma_{A,i}^2
\]

Solution same as vector case:

\[
\hat{A} = C_{YY} \left( C_{XX} + m\lambda I \right)^{-1},
\]

Prediction at new \( x \) using kernels:

\[
\hat{A} \varphi_x = \begin{bmatrix} \psi_{y_1} & \cdots & \psi_{y_m} \end{bmatrix} \left( K + \lambda m I \right)^{-1} \begin{bmatrix} k(x_1, x) & \cdots & k(x_m, x) \end{bmatrix} \\
= \sum_{i=1}^{m} \beta_i(x) \psi_{y_i}
\]

where \( K_{ij} = k(x_i, x_j) \)
How is loss $\|Y - AX\|_G^2$ relevant to conditional expectation of some $E_{Y|x}g(Y)$? Define: [Song et al. (2009), Grunewalder et al. (2013)]

$$\mu_{Y|x} := A\varphi_x$$
Ridge regression and the conditional feature mean

How is loss $\|Y - AX\|_G^2$ relevant to conditional expectation of some $E_{Y|x}g(Y)$? Define: [Song et al. (2009), Grunewalder et al. (2013)]

$$\mu_{Y|x} := A\varphi_x$$

We need $A$ to have the property

$$E_{Y|x}g(Y) \approx \langle g, \mu_{Y|x} \rangle_g$$

$$= \langle g, A\varphi_x \rangle_g$$
Ridge regression and the conditional feature mean

How is loss $\|Y - AX\|_G^2$ relevant to conditional expectation of some $E_{Y|x}g(Y)$? Define: [Song et al. (2009), Grunewalder et al. (2013)]

$$\mu_{Y|x} := A\varphi_x$$

We need $A$ to have the property

$$E_{Y|x}g(Y) \approx \langle g, \mu_{Y|x} \rangle_G$$

$$= \langle g, A\varphi_x \rangle_G$$

Natural risk function for conditional mean

$$R(A, P_{XY}) := \sup_{\|g\| \leq 1} E_X \left[ \left( E_{Y|x}g(Y) \right) - \langle g, A\varphi_x \rangle_G \right]_G^2.$$
The squared loss risk provides an upper bound on the natural risk.

\[
\mathcal{R}(A, P_{XY}) \leq E_{XY} \| \psi_Y - A \varphi_X \|^2_G
\]
Ridge regression and the conditional feature mean

The squared loss risk provides an upper bound on the natural risk.

\[ \mathcal{R}(A, \mathbf{P}_{XY}) \leq \mathbf{E}_{XY} \| \psi_Y - A\varphi_X \|_G^2 \]

Proof: Jensen

\[
\mathcal{R}(A, \mathbf{P}_{XY}) := \sup_{\|g\| \leq 1} \mathbf{E}_X \left[ (\mathbf{E}_{Y|X} g(Y)) - \langle g, A\varphi_X \rangle g \right]^2 ,
\]

\[
\leq \mathbf{E}_{XY} \sup_{\|g\| \leq 1} \left[ g(Y) - \langle g, A\varphi_X \rangle g \right]^2
\]
Ridge regression and the conditional feature mean

The squared loss risk provides an upper bound on the natural risk.

\[ \mathcal{R}(A, P_{XY}) \leq E_{XY} \| \psi_Y - A \varphi X \|_G^2 \]

Proof: Jensen

\[
\begin{align*}
\mathcal{R}(A, P_{XY}) &:= \sup_{\|g\| \leq 1} E_X \left[ (E_{X|g(Y)} - \langle g, A \varphi X \rangle G)^2 \right], \\
&\leq E_{XY} \sup_{\|g\| \leq 1} [g(Y) - \langle g, A \varphi X \rangle G]^2 \\
&= E_{XY} \sup_{\|g\| \leq 1} [\langle g, \psi_Y \rangle G - \langle g, A \varphi X \rangle G]^2
\end{align*}
\]
Ridge regression and the conditional feature mean

The squared loss risk provides an upper bound on the natural risk.

\[ \mathcal{R}(A, P_{XY}) \leq \mathbb{E}_{XY} \| \psi_Y - A \varphi_X \|^2_G \]

Proof: Jensen

\[
\begin{align*}
\mathcal{R}(A, P_{XY}) &:= \sup_{\|g\| \leq 1} \mathbb{E}_X \left[ \left( \mathbb{E}_{Y|X} g(Y) \right) - \langle g, A \varphi_X \rangle_G \right]^2, \\
&\leq \mathbb{E}_{XY} \sup_{\|g\| \leq 1} \left[ g(Y) - \langle g, A \varphi_X \rangle_G \right]^2 \\
&= \mathbb{E}_{XY} \sup_{\|g\| \leq 1} \langle g, \psi_Y - A \varphi_X \rangle_G^2
\end{align*}
\]
Ridge regression and the conditional feature mean

The squared loss risk provides an upper bound on the natural risk.

$$\mathcal{R}(A, P_{XY}) \leq E_{XY} \|\psi_Y - A\varphi_X\|^2_G$$

Proof: Jensen

$$\begin{align*}
\mathcal{R}(A, P_{XY}) &:= \sup_{\|g\| \leq 1} E_X \left[ (E_{Y|X} g(Y)) - \langle g, A\varphi_X \rangle \right]^2 \\
&\leq E_{XY} \sup_{\|g\| \leq 1} [g(Y) - \langle g, A\varphi_X \rangle]^2 \\
&= E_{XY} \sup_{\|g\| \leq 1} \langle g, \psi_Y - A\varphi_X \rangle^2_G \\
&= E_{XY} \|\psi_Y - A\varphi_X\|^2_G
\end{align*}$$
Ridge regression and the conditional feature mean

The squared loss risk provides an upper bound on the natural risk.

$$\mathcal{R}(A, P_{XY}) \leq E_{XY} \|\psi_Y - A \varphi_X\|^2_G$$

Proof: Jensen

$$\begin{align*}
\mathcal{R}(A, P_{XY}) &:= \sup_{\|g\| \leq 1} E_X \left[ (E_{Y|X}g(Y)) - \langle g, A \varphi_X \rangle \right]^2, \\
&\leq E_{XY} \sup_{\|g\| \leq 1} [g(Y) - \langle g, A \varphi_X \rangle]^2 \\
&= E_{XY} \sup_{\|g\| \leq 1} \langle g, \psi_Y - A \varphi_X \rangle^2_G \\
&= E_{XY} \|\psi_Y - A \varphi_X\|^2_G
\end{align*}$$

If we assume $E_Y[g(Y)|X = x] \in \mathcal{F}$ then upper bound tight
Kernel Bayes’ law

- Prior: $Y \sim \pi(y)$
- Likelihood: $(X|y) \sim P(x|y)$ from *training* distrib. $P(x, y)$
- Joint distribution: $Q(x, y) = P(x|y)\pi(y)$

Warning: $Q \neq P$, *change of measure* from $P(y)$ to $\pi(y)$
Kernel Bayes’ law

- **Prior**: $Y \sim \pi(y)$
- **Likelihood**: $(X|y) \sim P(x|y)$ from *training* distrib. $P(x, y)$
- **Joint distribution**: $Q(x, y) = P(x|y)\pi(y)$

**Warning**: $Q \neq P$, *change of measure* from $P(y)$ to $\pi(y)$

- **Bayes’ law**: Want $\mu_{Q(y|x)}$ with law

$$Q(y|x) = \frac{P(x|y)\pi(y)}{Q(x)}$$
Kernel Bayes’ law

- **Posterior embedding** via the usual conditional update,

\[ \mu_Q(y|x) = C_{Q(y,x)}C_{Q(x,x)}^{-1}\phi_x. \]
Kernel Bayes’ law

- **Posterior embedding** via the usual conditional update,

  \[ \mu_Q(y|x) = C_{Q(y,x)}C_{Q(x,x)}^{-1}\phi_x. \]

- Given mean embedding of prior: \( \mu_\pi(y) \)

- Learn **marginal covariance** by regression:

  \[ C_{Q(x,x)} = \int (\varphi_x \otimes \varphi_x) \mathbf{P}(x|y)\pi(y) dx dy = C_{(xx)y}C_{yy}^{-1}\mu_\pi(y) \]
Kernel Bayes’ law

- **Posterior embedding** via the usual conditional update,

\[ \mu_{\mathbf{Q}(y|x)} = C_{\mathbf{Q}(y,x)}C_{\mathbf{Q}(x,x)}^{-1}\phi_x. \]

- Given mean embedding of prior: \( \mu_{\pi}(y) \)

- Learn **marginal covariance** by regression:

\[ C_{\mathbf{Q}(x,x)} = \int (\varphi_x \otimes \varphi_x) \mathbf{P}(x|y)\pi(y)dxdy = C_{xx}yC_{yy}^{-1}\mu_{\pi}(y) \]

- Learn **cross-covariance** by regression:

\[ C_{\mathbf{Q}(y,x)} = \int (\varphi_y \otimes \varphi_x) \mathbf{P}(x|y)\pi(y)dxdy = C_{yx}yC_{yy}^{-1}\mu_{\pi}(y). \]
Kernel Bayes’ law: consistency result

- How to compute posterior expectation from data?
- Given samples: \( \{(x_i, y_i)\}_{i=1}^n \) from \( P_{xy} \), \( \{(u_j)\}_{j=1}^n \) from prior \( \pi \).
- Want to compute \( E[g(Y)|X = x] \) for \( g \) in \( G \)
- For any \( x \in \mathcal{X} \),

\[
|g_y^T R_{Y|X} k_X(x) - E[g(Y)|X = x]| = O_p(n^{-\frac{4}{27}}), \quad (n \to \infty),
\]

where

- \( g_y = (g(y_1), \ldots, g(y_n))^T \in \mathbb{R}^n \).
- \( k_X(x) = (k(x_1, x), \ldots, k(x_n, x))^T \in \mathbb{R}^n \).
- \( R_{Y|X} \) learned from the samples, contains the \( u_j \)

Smoothness assumptions:

- \( \pi/p_Y \in \mathcal{R}(C^{1/2}_{YY}) \), where \( p_Y \) p.d.f. of \( P_Y \),
- \( E[g(Y)|X = \cdot] \in \mathcal{R}(C^2_{Q(xx)}) \).
Experiment: Kernel Bayes’ law vs EKF
Experiment: Kernel Bayes’ law vs EKF

- Compare with extended Kalman filter (EKF) on camera orientation task
- 3600 downsampled frames of $20 \times 20$ RGB pixels ($X_t \in [0, 1]^{1200}$)
- 1800 training frames, remaining for test.
- Gaussian noise added to $X_t$. 
Experiment: Kernel Bayes’ law vs EKF

- Compare with extended Kalman filter (EKF) on camera orientation task
- 3600 downsampled frames of $20 \times 20$ RGB pixels ($X_t \in [0, 1]^{1200}$)
- 1800 training frames, remaining for test.
- Gaussian noise added to $X_t$.

Average MSE and standard errors (10 runs)

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>KBR (Gauss)</th>
<th>KBR (Tr)</th>
<th>Kalman (9 dim.)</th>
<th>Kalman (Quat.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-4}$</td>
<td>0.210 ± 0.015</td>
<td>0.146 ± 0.003</td>
<td>1.980 ± 0.083</td>
<td>0.557 ± 0.023</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>0.222 ± 0.009</td>
<td>0.210 ± 0.008</td>
<td>1.935 ± 0.064</td>
<td>0.541 ± 0.022</td>
</tr>
</tbody>
</table>
Selected references

Characteristic kernels and mean embeddings:


Two-sample, independence, conditional independence tests:

Conditional mean embedding, RKHS-valued regression:


Kernel Bayes rule:

Conditions for ridge regression = conditional mean

Conditional mean obtained by ridge regression when $E_Y[g(Y)|X = x] \in \mathcal{F}$

Given a function $g \in \mathcal{G}$. Assume $E_{Y|X}[g(Y)|X = \cdot] \in \mathcal{F}$. Then

$$C_{XX} E_{Y|X}[g(Y)|X = \cdot] = C_{XY}g.$$

Why this is useful:

$$E_{Y|X}[g(Y)|X = x] = \langle E_{Y|X}[g(Y)|X = \cdot], \varphi_x \rangle_{\mathcal{F}}$$

$$= \langle C_{XX}^{-1}C_{XY}g, \varphi_x \rangle_{\mathcal{F}}$$

$$= \langle g, C_{XX}^{-1}C_{XX}^{\top}\varphi_x \rangle g$$

reduction
Conditions for ridge regression = conditional mean

Conditional mean obtained by ridge regression when \( E_Y[g(Y)|X = x] \in \mathcal{F} \)

Given a function \( g \in \mathcal{G} \). Assume \( E_{Y|X} [g(Y)|X = \cdot] \in \mathcal{F} \). Then

\[
C_{XX} E_{Y|X} [g(Y)|X = \cdot] = C_{XY}g.
\]

Proof: [Fukumizu et al., 2004]

For all \( f \in \mathcal{F} \), by definition of \( C_{XX} \),

\[
\langle f, C_{XX} E_{Y|X} [g(Y)|X = \cdot] \rangle_{\mathcal{F}}
\]

\[
= \text{cov} \left( f, E_{Y|X} [g(Y)|X = \cdot] \right)
\]

\[
= E_X (f(X) E_{Y|X} [g(Y)|X])
\]

\[
= E_{XY}(f(X)g(Y))
\]

\[
= \langle f, C_{XY}g \rangle,
\]

by definition of \( C_{XY} \).
References

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm.


In NIPS, 2009.

D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a large-scale kernel machine.


A. Rahimi and B. Recht. Random features for large-scale kernel machines.

D. Haussler. Convolution kernels on discrete structures.