Approximation Properties of Reproducing Kernels

M. Eberts (Meister), S. Fischer, N. Schmid, P. Thomann, & I. Steinwart

Institute of Stochastics and Applications
University of Stuttgart

Guangzhou
May 20th, 2017
Overview

- Approximation theoretic questions related to kernel-based learning
- More flexible kernels: spatial decompositions
- More flexible kernels: deeper compositions
Informal Description of Supervised Learning

- $X$ space of input samples
  - $Y$ space of labels, usually $Y \subset \mathbb{R}$.
- Already observed samples

\[ D = ((x_1, y_1), \ldots, (x_n, y_n)) \in (X \times Y)^n \]
Informal Description of Supervised Learning

- $X$ space of input samples
  $Y$ space of labels, usually $Y \subset \mathbb{R}$.
- Already observed samples

$$D = ((x_1, y_1), \ldots, (x_n, y_n)) \in (X \times Y)^n$$

- **Goal:**
  With the help of $D$ find a function $f_D : X \rightarrow \mathbb{R}$ such that $f_D(x)$ is a good prediction of the label $y$ for new, unseen $x$.

- **Learning method:**
  Assigns to every training set $D$ a predictor $f_D : X \rightarrow \mathbb{R}$. 
**Problem:**
The labels $y$ are $\mathbb{R}$-valued.

**Goal:**
Estimate label $y$ for new data $x$ as accurate as possible.

**Example:**
Assumptions

- We have bounded labels $Y = [-1, 1]$.
- $P$ is an unknown probability measure on $X \times [-1, 1]$.
- $D = ((x_1, y_1), \ldots, (x_n, y_n)) \in (X \times Y)^n$ is sampled from $P^n$.
- Future samples $(x, y)$ will also be sampled from $P$.
- (For this talk) we mostly use the least squares loss

$$L(y, t) := (y - t)^2$$

to assess quality of a prediction $t$ for $y$. 
The risk of a predictor $f : X \rightarrow \mathbb{R}$ is the average loss

$$\mathcal{R}_{L,P}(f) := \int_{X \times Y} L(y, f(x)) \, dP(x, y).$$

The Bayes risk is the smallest possible risk

$$\mathcal{R}^*_{L,P} := \inf \left\{ \mathcal{R}_{L,P}(f) \mid f : X \rightarrow \mathbb{R} \text{ (measurable)} \right\}.$$

The Bayes predictor for the least squares loss is $f_{L,P}^*(x) := \mathbb{E}(Y|x)$, i.e.

$$\mathcal{R}_{L,P}(f_{L,P}^*) = \mathcal{R}^*_{L,P}.$$

The excess risk satisfies

$$\mathcal{R}_{L,P}(f) - \mathcal{R}^*_{L,P} = \| f - f_{L,P}^* \|^2_{L_2(P_X)}.$$
Kernel-based learning methods

- Let $H$ be a reproducing kernel Hilbert space, here with bounded kernel.
- Let $L : Y \times \mathbb{R} \to [0, \infty)$ be a convex loss.
Kernel-based learning methods

- Let $H$ be a reproducing kernel Hilbert space, here with bounded kernel
- Let $L : Y \times \mathbb{R} \rightarrow [0, \infty)$ be a convex loss
- Kernel-based learning methods (e.g. SVMs) solve the problem

$$f_{D,\lambda} = \arg \min_{f \in H} \lambda \|f\|_H^2 + \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) ,$$  \hspace{1cm} (1)

where $\lambda > 0$ is a free regularization parameter. Solution is of the form

$$f_{D,\lambda} = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) .$$

**Historical Notes**
- G. Wahba (1971 –): Least squares loss
- V. Vapnik et al. (1992 –): Hinge loss
- Other losses in the last decade or so.
A Typical Oracle Inequality

- Consider the approximation (regularization) error
  \[ A(\lambda) := \inf_{f \in H} \lambda \| f \|_H^2 + R_{L,P}(f) - R^*_L,P \]
- Assume an (dyadic) entropy number behavior
  \[ e_i(I : H \to L_2(P_X)) \leq i^{-1/(2p)} \]

Then with probability \( P^n \) not smaller than \( 1 - e^{-\tau} \) we have

\[ R_{L,P}(f_{D,\lambda}) - R^*_L,P \leq K \left( A(\lambda) + \frac{1}{\lambda^p n} + \frac{\tau A(\lambda)}{\lambda n} \right) \]

Remarks:

- If rate \( A(\lambda) \to 0 \) for \( \lambda \to 0 \) known, we obtain learning rates.
- Entropy behaviour is equivalent to a similar eigenvalue behaviour of
  \[ T_k : L_2(P_X) \to L_2(P_X) \]
  \[ T_k f := \int_X k(x, \cdot) f(x) dP_X(x) \]
For Banach spaces $F \hookrightarrow E$ and $x \in E$, the K-functional is

$$K(x, t) := \inf_{y \in F} \|x - y\|_E + t\|y\|_F, \quad t > 0.$$ 

For $0 < \beta < 1$, $1 \leq r \leq \infty$, the interpolation space $[E, F]_{\beta, r}$ consists of those $x \in E$ with finite $\|x\|_{\beta, r}$, where

$$\|x\|_{\beta, r} := \begin{cases} \left(\int_0^\infty (t^{-\beta} K(x, t))^r t^{-1} dt\right)^{1/r} & \text{if } 1 \leq r < \infty \\ \sup_{t > 0} t^{-\beta} K(x, t) & \text{if } r = \infty. \end{cases}$$

We are interested in the spaces $[L_2(P_X), [H]_\sim]_{\beta, r}$. 
Interpolation Spaces vs. Approximation Properties

Smale & Zhou, 2003
\[ A(\lambda) \preceq \lambda^\beta \] if and only if \( f_{L,P}^* \in [L_2(P_X), [H]_\sim]_{\beta,\infty} \).

Operator techniques (Caponnetto and De Vito, 2007, . . .)
Rates for \( \mathcal{R}_{L,P}(f_{D,\lambda}) \rightarrow \mathcal{R}_{L,P}^* \) are obtained if
\[ f_{L,P}^* \in \text{im } T_{\beta/2}^k \]

Smale & Zhou, 2003
If \( X \) is compact, supp \( P_X = X \) and \( k \) continuous, then
\[ [L_2(P_X), [H]_\sim]_{\beta+\varepsilon,\infty} \subset \text{im } T_{\beta/2}^k \subset [L_2(P_X), [H]_\sim]_{\beta,\infty} \]
Both approximation assumptions are almost the same, since
\[ [L_2(P_X), [H]_\sim]_{\beta+\varepsilon,\infty} \leftrightarrow [L_2(P_X), [H]_\sim]_{\beta,1} \leftrightarrow [L_2(P_X), [H]_\sim]_{\beta,\infty} \]
Let $k$ be a reproducing kernel with compact $I_{k,\nu} : H \to L_2(\nu)$.

Then $T_{k,\nu} = I_{k,\nu} \circ I_{k,\nu}^*$ is selfadjoint, positive, and compact.

Let $(\mu_i)_{i \in I}$ be the family of non-zero eigenvalues of $T_{k,\nu}$ and $([\tilde{e}_i])$ be a corresponding ONS of eigenfunctions in $L_2(\nu)$.

Then $e_i := \mu_i^{-1} I_{k,\nu} [\tilde{e}_i] \in H$ satisfies $[e_i] = [\tilde{e}_i] \in H$ and we have:
Spectral Theorem, Revisited

- Let $k$ be a reproducing kernel with compact $I_{k,\nu} : H \to L_2(\nu)$.
- Then $T_{k,\nu} = I_{k,\nu} \circ I_{k,\nu}^*$ is selfadjoint, positive, and compact.
- Let $(\mu_i)_{i \in I}$ be the family of non-zero eigenvalues of $T_{k,\nu}$ and $([\tilde{e}_i])_\sim$ be a corresponding ONS of eigenfunctions in $L_2(\nu)$.

Then $e_i := \mu_i^{-1} I_{k,\nu}^* [\tilde{e}_i]_\sim \in H$ satisfies $[e_i]_\sim = [\tilde{e}_i]_\sim$ and we have:
- $([e_i]_\sim)$ is an ONS in $L_2(\nu)$.
- $(\sqrt{\mu_i} e_i)$ is an ONS in $H$.

\[
(ker I_{k,\nu})^\perp = \text{im} I_{k,\nu}^* = \text{span}\{\sqrt{\mu_i} e_i : i \in I\}
\]
\[
(ker T_{k,\nu})^\perp = (ker I_{k,\nu}^*)^\perp = \text{im} I_{k,\nu} = \text{span}\{[e_i]_\sim : i \in I\}
\]

Consequence.
$L_2(\nu)$ and $H$ “share a subspace” described by $(e_i)$. 
The Situation so far

Isometric Copy of $H$ in $L_2(\nu)$

$$[H]_\sim = \left\{ \sum_{i \in I} a_i \mu_i^{1/2} [e_i]_\sim : (a_i) \in \ell_2(I) \right\}$$

Closure of $H$ in $L_2(\nu)$

$$\overline{[H]_\sim}^{L_2(\nu)} = \left\{ \sum_{i \in I} a_i [e_i]_\sim : (a_i) \in \ell_2(I) \right\}$$

Question
What is in between?
For $\beta \in [0, 1]$ we can consider the following subspace of $L_2(\nu)$:

$$[H]_{\sim}^\beta := \left\{ \sum_{i \in I} a_i \mu_i^{\beta/2} [e_i]_{\sim} : (a_i) \in \ell_2(I) \right\}$$

$$= \left\{ \sum_{i \in I} b_i [e_i]_{\sim} : (b_i) \in \ell_2(\mu^{-\beta}) \right\},$$

where $\ell_2(\mu^{-\beta})$ is a weighted sequence space with norm:

$$\|(b_i)\|_{\ell_2(\mu^{-\beta})}^2 := \sum_{i \in I} b_i^2 \mu_i^{-\beta}$$
For $\beta \in [0, 1]$ we can consider the following subspace of $L_2(\nu)$:

$$[H]_\sim^\beta := \left\{ \sum_{i \in I} a_i \mu_i^{\beta/2} \langle e_i \rangle_\sim : (a_i) \in \ell_2(I) \right\}$$

$$= \left\{ \sum_{i \in I} b_i \langle e_i \rangle_\sim : (b_i) \in \ell_2(\mu^{-\beta}) \right\},$$

where $\ell_2(\mu^{-\beta})$ is a weighted sequence space with norm:

$$\| (b_i) \|^2_{\ell_2(\mu^{-\beta})} := \sum_{i \in I} b_i^2 \mu_i^{-\beta}$$

By construction, $(\mu_i^{\beta/2} \langle e_i \rangle_\sim)_{i \in I}$ is an ONB of $[H]_\sim^\beta$ and

$$[H]_\sim^0 = [H]_\sim L(\nu)$$

$$[H]_\sim^1 = [H]_\sim$$

$$[H]_\sim^\beta = \text{im} \ T_{k,\nu}^{\beta/2}$$
If \( I_{k,\nu} : H \to L_2(\nu) \) is compact, then, for \( \beta \in (0, 1) \), we have

\[
\text{im } T^{\beta/2}_{k,\nu} = [H]^{\beta} \cong [L_2(\nu), [H]^{\sim}]_{\beta,2}.
\]
If $I_{k,\nu} : H \to L_2(\nu)$ is compact, then, for $\beta \in (0, 1)$, we have

$$\text{im } T_{k,\nu}^{\beta/2} = [H]_\sim^\beta \cong [L_2(\nu), [H]_\sim]_{\beta,2}.$$ 

**Idea of the Proof.**

- Interpolating $L_2(\nu)$ and $[H]_\sim$ is the same as interpolating $\ell_2(I)$ and $\ell_2(\mu^{-1})$.
- We have $[\ell_2(I), \ell_2(\mu^{-1})]_{\beta,2} \cong \ell_2(\mu^{-\beta})$. 
Rates for Fixed Kernel


- Assume $\mu_i \leq i^{-1/p}$
- Assume $f^*_{L,P} \in [L_2(P_X), H]_{\beta,\infty}$ for some $\beta \in (0, 1]$.
- Assume $[L_2(P_X), H]_{s,1} \hookrightarrow L_\infty(P_X)$ for $s = \min\{1, p/(1 - \beta)\}$. This is equivalent to

$$\|f\|_\infty \leq c \|f\|^s_H \|f\|^{1-s}_{L_2(P_X)}, \quad f \in H$$

Then kernel method can learn with the optimal rate $n^{-\frac{\beta}{\beta+p}}$.

Special Case: Sobolev Setting (e.g. Kohler)

- $X$ ball in $\mathbb{R}^d$ and $H := W^m(X)$ Sobolev space with $m > d/2$.
  $\hookrightarrow$ Least squares with splines.
- $P_X$ uniform distribution and $f^*_{L,P} \in B^s_{2,2}(X)$ for some $s \in (d/2, m]$.

The kernel method can learn with the optimal rate $n^{-\frac{2s}{2s+d}}$. 
Improved Convergence

**Fischer & S., 2017**

- Assume $\mu_i \leq i^{-1/p}$
- Assume $f^*_{L,P} \in [L_2(P_X), H]_{\beta,2}$ for some $\beta \in (0, 1]$.
- Assume $[L_2(P_X), H]_{\alpha,2} \hookrightarrow L_\infty(P_X)$ for some $\alpha \in (0, 1)$.

Then, for a suitable sequence $(\lambda_n)$ the decision functions $f_{D,\lambda_n}$ converges to $f^*_{L,P}$ in the norm of $[L_2(P_X), H]_{\gamma,2}$ for $\gamma \in [0, \beta]$ with rate $n^{-r}$, where

$$r = \frac{\beta - \gamma}{\max\{\alpha, \beta\} + p}$$

**Example.**
Let $H = W^m_m(X)$ and $f^*_{L,P} \in B^s_{2,2}(X)$ for some $s \in (d/2, m]$. For $t \in (0, s)$, the rate in $B^t_{2,2}(X)$ is $n^{-r}$, where

$$r = \frac{2s - 2t}{2s + d}$$

Consider Gaussian RKHS $H_\gamma(X)$ with kernel

$$k_\gamma(x, x') := \exp(-\gamma^{-2}\|x - x'\|_2^2), \quad x, x' \in X.$$ 

Then $A_\gamma(\lambda) \leq \lambda^\beta$ for some $\beta \in (0, 1]$ implies $f^*_{L,P} \in C^\infty(X)$.

### Solution

Consider width $\gamma$ as a free parameter.

Theory presented so far does not work anymore.
Rates for Gaussian Kernels

Eberts & S., 2011/3

- $X$ ball in $\mathbb{R}^d$ and $H_\gamma$ is RKHS of Gaussian kernel $k_\gamma$.
- $P_X$ has bounded Lebesgue density.
- Pick $\lambda$ and $\gamma$ by a training/validation approach.

Then, for $s \geq 1$, every $f^{*}_{L, P} \in W^{s}_2(X)$ is learned with the rate $n^{\frac{2s}{2s+d} + \varepsilon}$ without knowing $s$.

The extra factor $n^\varepsilon$ can be replaced by a logarithmic factor.

**Key idea of the proof**

Bound approximation error by convoluting $f^{*}_{L, P}$ with weighted sum of kernels $k_{\gamma_1}, \ldots, k_{\gamma_m}$.
Spatial Decompositions
Optimization Problem

\[ f_{D, \lambda} = \arg \min_{f \in H} \lambda \| f \|^2_H + \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \]

Example: Dual Problem for Hinge Loss

\[ \alpha^* \in \arg \max_{\alpha \in [0, \frac{1}{2\lambda n}]} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \]

Re-substitution

\[ f_{D, \lambda} = \sum_{i=1}^{n} y_i \alpha^*_i k(\cdot, x_i) \]
Problems for Large Data

Computational Requirements

- The size of the optimization problem grows linearly in $n$.
- The kernel matrix $(k(x_i, x_j))$ grows quadratically in $n$.
- Computing the decision functions grows linearly in $n$.
- Solving the optimization problem costs between $O(n^2)$ and $O(n^3)$

Consequences

For 64GB machines, kernel matrices for $n > 100,000$ cannot be stored. Training for such sample sizes, even if only a fixed parameter pair $(\lambda, \sigma)$ is considered, may take up to hours.
Using kernel methods without tricks is impossible for data sizes ranging in the millions.
Problems for Large Data

**Computational Requirements**
- The size of the optimization problem grows linearly in $n$.
- The kernel matrix $(k(x_i, x_j))$ grows quadratically in $n$.
- Computing the decision functions grows linearly in $n$.
- Solving the optimization problem costs between $O(n^2)$ and $O(n^3)$

**Consequences**
- For 64GB machines, kernel matrices for $n > 100,000$ cannot be stored.
- Training for such sample sizes, even if only a fixed parameter pair $(\lambda, \sigma)$ is considered, may take up to hours.
- Using kernel methods without tricks is impossible for data sizes ranging in the millions.


Spatial splits: now
Construction

- Split bounded $X \subset \mathbb{R}^d$ into cells $A_1, \ldots, A_m$ of diameter $\leq r$.
- On each cell $A_j$ train a kernel method with Gaussian kernel and the data in $A_j$, i.e.

$$D_j := \{(x_i, y_i) \in D : x_i \in A_j\}.$$

- The hyper-parameters $\lambda$ and $\sigma$ are found by training/validation on each cell separately.
- To predict $y$ for some test sample $x$, only take the decision function that is constructed on the cell $A_j$ with $x \in A_j$. 
Main Result

Rates for Localized kernel methods (Meister & S., 2016)

- Pick some $\beta > 0$ and $r_n \sim n^{-1/\beta}$.
- Assume that $f^*_{L,P} \in W^s_2(X)$ for some $s < \frac{\beta-d}{2}$.

Then the localized kernel method learns with rate $n^{-\frac{2s}{2s+d}+\varepsilon}$. 

Remarks

- Good adaptivity requires large $\beta$.
- Large $\beta$ leads to large cells.
- Trade-off between statistics and computational complexity.
- Similar results for quantile regression.
Rates for Localized kernel methods (Meister & S., 2016)

- Pick some $\beta > 0$ and $r_n \sim n^{-1/\beta}$.
- Assume that $f^*_{L,P} \in W^s_2(X)$ for some $s < \frac{\beta - d}{2}$.

Then the localized kernel method learns with rate $n^{-2s/d + \varepsilon}$.

Remarks

- Good adaptivity requires large $\beta$.
- Large $\beta$ leads to large cells.
  - Trade-off between statistics and computational complexity.
- Similar results for quantile regression.
The split kernel method can be viewed as an ordinary kernel method using the RKHS:

\[ H = \bigoplus_{j=1}^{m} \sqrt{\lambda_j} H_{A_j,\sigma_j} \]

Investigate how properties of the local RKHS influence properties of the global \( H \) in view of \( P \).

Again we are facing a kernel more complex than usual.
Controlling the Size of the Cells

- **Data:** covertype in binary classification from LIBSVM site, $d = 54$
- **Method:** Hinge loss and number of samples in cells are controlled
Deeper Compositions
At each non-input node, we perform the operation

\[ x \mapsto \sigma(\langle w, x \rangle + b) \]

- Do we need the network structure on the right to classify?
- Can we replace the feature modification on the left by something else?
A simple Network with One Hidden Layer

- Input space $X = [0, 1]$
- One hidden layer with $m$ ReLU-units each performing
  
  $$x \mapsto \Phi_j(x) := \left| w_j x + b_j \right|_+, \quad j = 1, \ldots, m.$$ 

- Output layer creates a function
  
  $$x \mapsto \langle v, \Phi(x) \rangle_{\ell_2^d} = \sum_{j=1}^{m} v_j \left| w_j x + b_j \right|_+$$ 

  Thus it realizes an element in the RKHS with FM $\Phi := (\Phi_1, \ldots, \Phi_m)$. 

For fixed $w_i, b_i \in \mathbb{R}^m$ this RKHS is a set of piecewise linear functions with kinks at $-\frac{b_i}{w_i}, \ldots, -\frac{b_m}{w_m}$. The NN represents all piecewise linear functions with at most $m-1$ kinks and most with $m$ kinks.
A simple Network with One Hidden Layer

- Input space $X = [0, 1]$
- One hidden layer with $m$ ReLU-units each performing
  \[ x \mapsto \Phi_j(x) := \max(w_j x + b_j), \quad j = 1, \ldots, m. \]
- Output layer creates a function
  \[ x \mapsto \langle v, \Phi(x) \rangle_{\ell^d_2} = \sum_{j=1}^m v_j \max(w_j x + b_j). \]

Thus it realizes an element in the RKHS with FM $\Phi := (\Phi_1, \ldots, \Phi_m)$.

- For fixed $w, b \in \mathbb{R}^m$ this RKHS is a set of piecewise linear functions with kinks at
  \[ -\frac{b_1}{w_1}, \ldots, -\frac{b_m}{w_m} \]

- The NN represents all piecewise linear functions with at most $m - 1$ kinks and most with $m$ kinks.

\[ \Rightarrow \text{nonlinear structure, parametric method for each fixed design} \]
Observation
Each layer performs a non-linear transformation

$$\mathbb{R}^{m_i} \rightarrow \mathbb{R}^{m_{i+1}}$$

$$x \mapsto \Phi_{w_i,b_i}(x)$$

Entire feature map is $$\Phi := \Phi_{w_L,b_L} \circ \cdots \circ \Phi_{w_1,b_1}$$

Idea for Rest of the Talk
Replace finite-dimensional spaces by infinite dimensional Hilbert spaces

$$H_i \rightarrow H_{i+1}$$

$$x \mapsto \Phi_{w_i}(x)$$

Use the kernel of the resulting feature map $$\Phi := \Phi_{w_L} \circ \cdots \circ \Phi_{w_1}$$
“Historical Remarks”

Bach, Lanckriet, and Jordan 2004
$L = 2$, linear kernel in second layer $\Rightarrow$ Multiple kernel learning

Cho and Saul, 2009
General setup and some examples

Zhuang, Tsang, and Hoi, 2011
$L = 2$, sum of kernels in composition step, pseudo-dimension bound

Strobl and Visweswaran, 2013
Sums of kernels in each composition step, VC-bounds

Tang, 2013
$\Phi_{L-1} \circ \cdots \circ \Phi_1$ is a neural net with $M$ output nodes, $\Phi_L$ is linear “SVM”.

Wilson, Hu, Salakhutdinov, and Xing, 2016
$\Phi_{L-1} \circ \cdots \circ \Phi_1$ is a neural net with $M$ output nodes, $\Phi_L$ is non-linear.
Observations

Let $H$ be a Hilbert space and $\Phi : X \to H$.

- We obtain a new kernel on $X$ by
  $$k_{\gamma,H,X}(x, x') := \exp\left(-\gamma^{-2} \|\Phi(x) - \Phi(x')\|_H^2\right), \quad x, x' \in X,$$

- If $k(x, x') := \langle \Phi(x), \Phi(x') \rangle$ with $k(x, x) \equiv c$, then
  $$k_{\gamma,H}(x, x') = \exp\left(-2\gamma^{-2}(c - k(x, x'))\right)$$

- If $\Phi_{\gamma,H} : H \to H_{\gamma,H}$ is a feature map of $k_{\gamma,H}$ on $H$, then
  $$\Phi_{\gamma,H} \circ \Phi$$
  is a feature map of $k_{\gamma,H,X}$. 

Idea

- So far we have

\[ k_{\gamma, x, H}(x, x') = \exp(-2\gamma^{-2}(c - k(x, x'))) \]  \hspace{1cm} (2)

- For \( I \subset \{1, \ldots, d\} \) we write \( x_I := (x_i)_{i \in I} \).

- For \( I_1, \ldots, I_m \subset \{1, \ldots, d\} \), let \( k_1, \ldots, k_m \) be kernels on \( \mathbb{R}^{|I_1|}, \ldots, \mathbb{R}^{|I_1|} \).

- Assume that \( k_i(x, x) \equiv 1 \).

For \( I := I_1 \cup \cdots \cup I_m \) consider the kernel

\[ k(x, x') := \sum_{i=1}^{m} \omega_i^2 k_i(x_{I_i}, x'_{I_i}), \quad x, x' \in X_I. \]

in (2). This kernel is denoted by \( k_w \). This can be iterated!
**Definition**

Let $H$ be the RKHS of the kernel

$$k(x, x') := \sum_{i=1}^{m} w_i^2 k_i(x_{l_i}, x'_{l_i}), \quad x, x' \in X_l.$$ 

Then the resulting hierarchical Gaussian kernel $k_{\gamma, X_l, H}$, that is

$$k_{\gamma, X, H}(x, x') = \exp(-2\gamma^{-2}(c - k(x, x')))$$

is said to be:

- of depth 1, if all kernels $k_1, \ldots, k_m$ are linear kernels.
- of depth $L > 1$, if all $k_1, \ldots, k_m$ are hierarchical Gaussian kernels of depth $L - 1$. 
Example 1
Hierarchical Gaussian kernels of depth $L = 1$ are of the form

$$k_w(x, x') := \exp\left(- \sum_{i \in I} w_i^2 (x_i - x'_i)^2 \right), \quad x, x' \in X,$$

ARD kernel

Example 2
Hierarchical Gaussian kernels of depth $L = 2$ are of the form

$$k_{W^{(1)}, w, \gamma}(x, x') = \exp\left(-2\gamma^{-2} \sum_{i=1}^{m} w_i^2 \left(1 - k_w(x_{I_i}, x'_{I_i})\right)\right)$$

$$= \exp\left(-2\gamma^{-2} \sum_{i=1}^{m} w_i^2 \left(1 - \exp\left(- \sum_{j \in I_i} w_{j,i}^2 (x_j - x'_j)^2 \right)\right)\right).$$
Example of a hierarchical Gaussian kernels of depth $L = 3$. 
Definition
A continuous kernel on a compact metric space $X$ is universal, if its RKHS is dense in $C(X)$.

Theorem (Christmann & S., 2010)
A kernel of the form

$$k_{\gamma,H,X}(x,x') := \exp(-\gamma^{-2}\|\Phi(x) - \Phi(x')\|_H^2), \quad x, x' \in X,$$

is universal, if $\Phi$ is continuous and injective.
Theorem (S. & Thomann, 2016)
A hierarchical Gaussian kernel of depth $L \geq 1$ is universal, if it does not ignore coordinates.

Corollary (S. & Thomann, 2016)
Every SVM using a fixed hierarchical Gaussian kernel of depth $L \geq 1$ that does not ignore coordinates is universally consistent.

Remarks
- Learning rates for weights changing with sample size $n$?
- For which distributions do hierarchical Gaussian kernels help?
- Learning the kernel can be, in principle, decoupled from learning a classifier/regressor.
A few words on the proof . . .

- Induction over $L$
- At the highest level we have

$$ k_{\gamma,X,H}(x,x') = \prod_{i=1}^{l} k_{\gamma/w_i,X,H_i}(x_i,x'_i), \quad x, x' \in X_I. $$

- If $k_I$ and $k_J$ are universal kernels on $X_I$ and $X_J$, then $k_I \otimes k_J$ defined by

$$ k_I \otimes k_J(x,x') := k_I(x_I,x'_I) \cdot k_J(x_J,x'_J), \quad x, x' \in X_{I \cup J} $$

is a universal kernel on $X_{I \cup J}$. Use Stone-Weierstraß.

- Universal kernels have injective feature maps.
  $\leadsto k_{\gamma/w_i,X,H_i}$ are universal by induction assumption
## LS Error for Automated Learning Procedures

<table>
<thead>
<tr>
<th>Data Set</th>
<th>SVM</th>
<th>HKL</th>
<th>Ours</th>
<th>RF</th>
<th>DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANK</td>
<td>.2978 ± .0024</td>
<td>.2939 ± .0028</td>
<td>.2596 ± .0039</td>
<td>.2687 ± .0027</td>
<td>.2931 ± .0025</td>
</tr>
<tr>
<td>CADATA</td>
<td>.0538 ± .0016</td>
<td>.0625 ± .0014</td>
<td>.0525 ± .0019</td>
<td>.0509 ± .0015</td>
<td>.0550 ± .0015</td>
</tr>
<tr>
<td>COD</td>
<td>.1574 ± .0023</td>
<td>.1734 ± .0013</td>
<td>.1309 ± .0050</td>
<td>.1725 ± .0020</td>
<td>.1154 ± .0013</td>
</tr>
<tr>
<td>COVTYPE</td>
<td>.5205 ± .0043</td>
<td>.6100 ± .0042</td>
<td>.3995 ± .0148</td>
<td>.4878 ± .0041</td>
<td>.5027 ± .0063</td>
</tr>
<tr>
<td>CPU SMALL</td>
<td>.0036 ± .0002</td>
<td>.0046 ± .0004</td>
<td>.0034 ± .0002</td>
<td>.0032 ± .0002</td>
<td>.0038 ± .0001</td>
</tr>
<tr>
<td>CYCLE</td>
<td>.0105 ± .0003</td>
<td>.0122 ± .0003</td>
<td>.0098 ± .0005</td>
<td>.0084 ± .0003</td>
<td>.0121 ± .0003</td>
</tr>
<tr>
<td>HIGGS</td>
<td>.9021 ± .0017</td>
<td>.8178 ± .0074</td>
<td>.8023 ± .0175</td>
<td>.7770 ± .0024</td>
<td>.9162 ± .0024</td>
</tr>
<tr>
<td>LETTER</td>
<td>.0451 ± .0015</td>
<td>.1151 ± .0018</td>
<td>.0339 ± .0014</td>
<td>.0577 ± .0015</td>
<td>.0448 ± .0018</td>
</tr>
<tr>
<td>MAGIC</td>
<td>.4007 ± .0083</td>
<td>.4282 ± .0082</td>
<td>.3900 ± .0093</td>
<td>.3772 ± .0079</td>
<td>.3783 ± .0085</td>
</tr>
<tr>
<td>PENDIGITS</td>
<td>.0079 ± .0007</td>
<td>.0243 ± .0012</td>
<td>.0070 ± .0007</td>
<td>.0127 ± .0012</td>
<td>.0079 ± .0010</td>
</tr>
<tr>
<td>SAT IMAGE</td>
<td>.0488 ± .0029</td>
<td>.1078 ± .0059</td>
<td>.0467 ± .0030</td>
<td>.0525 ± .0026</td>
<td>.0525 ± .0033</td>
</tr>
<tr>
<td>SEISMIC</td>
<td>.3113 ± .0013</td>
<td>.3189 ± .0022</td>
<td>.2981 ± .0016</td>
<td>.2955 ± .0012</td>
<td>.2975 ± .0014</td>
</tr>
<tr>
<td>SHUTTLE</td>
<td>.0046 ± .0003</td>
<td>.0129 ± .0007</td>
<td>.0042 ± .0004</td>
<td>.0008 ± .0002</td>
<td>.0059 ± .0004</td>
</tr>
<tr>
<td>THYROID</td>
<td>.1750 ± .0081</td>
<td>.1637 ± .0083</td>
<td>.1538 ± .0080</td>
<td>.0251 ± .0031</td>
<td>.1522 ± .0080</td>
</tr>
<tr>
<td>UPDRS</td>
<td>.0537 ± .0052</td>
<td>.1774 ± .0090</td>
<td>.0059 ± .0021</td>
<td>.0305 ± .0016</td>
<td>.0531 ± .0042</td>
</tr>
</tbody>
</table>
Detailed Comparison of 3 Best Methods

- **Relative error compared to SVM**: The graphs show the relative error compared to SVM across different data set numbers. The error is represented on a logarithmic scale, ranging from 0.25 to 1.75.
- **Improvements in % compared to 2nd best method**: This graph illustrates the improvements in percentage compared to the second-best method, ranging from 0 to 8.
- **Worsening in % compared to best error**: This graph shows the percentage worsening compared to the best error, ranging from 0 to 12.

Methods included are SVM, Ours, RF, and DNN.
Resources

**Paper**


**Software**
