Rethinking Kernel Learning

Yiping Lu
1500010638
School Of Mathematical Science
Peking University
BeiJing, China
luyiping9712@pku.edu.cn

Abstract

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space.(Wiki) In this review, I'll review kernel learning from a statistical learning view point and its potential future application.

1 Reproducing Kernel Hilbert Space

1.1 Reproducing Kernel Hilber Space

Definition. A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ if it is symmetric and positive semi-definite.

The definition of RKHS have many equivalent ways.

Definition.

- *Definition 1.* $k(\cdot,\cdot)$ is a reproducing kernel of a Hilber space $\mathcal{H}$ if for $\forall f \in \mathcal{H}$, we have $f(x) = \langle k(x,\cdot), f\rangle$
- *Definition 2.* a Hilbert space of functions with all evaluation functinos bounded and linear.

Learning a classifier in the RKHS can be consider as a mapping maps data into a higher dimensional feature space $x \to \Phi(x) = \left[\sqrt{\lambda_1}\phi_1(x), \cdots, \sqrt{\lambda_i}\phi_i(x), \cdots\right]$ where λ_i and ϕ_i are the eigenvalues and eigenfunctions of the reproducing kernel $k(\cdot,\cdot)$. (Moreover, we have $k(x,z) = \sum \lambda_i\phi_i(x)\phi_i(z)$.

Representer Theorem. Give a reproducing kernel k and let $\mathcal{H}$ be the corresponding RKHS. Then for a function $L : \mathbb{R}^n \to \mathbb{R}$ and non-decreasing function $\Omega : \mathbb{R} \to \mathbb{R}$. The solution of the optimization problem

$$\min_{f\in\mathcal{H}} J(f) = \min_{f\in\mathcal{H}} \left\{L(f(x_1), \cdots, f(x_n)) + \Omega(\|f\|^2_{\mathcal{H}})\right\}$$

can be expressed as

$$f^* = \sum_{i=1}^{n} \alpha_i k(x_i,\cdot).$$

Furthermore, if $\Omega(\cdot)$ is strictly increasing, then all solutions have this form.

The representer theorem enables us to solve the optimization problem in a finite dimension subspace.
Example.

- Linear Kernel: $k(x,z) = \langle x, z\rangle$

- Polynomial Kernel $k(x,z) = (\langle x, z\rangle + c)^d$
- RBF Kernel: $k(x,z) = exp(-\gamma\|x-z\|^2)$
- Consider the hilber space $\mathcal{H}^1$ and the inner product $\langle u, v\rangle = \langle u, \Delta v\rangle$, in this setting the reproducing kernel is the Green function.

1.2 Statistical Learning Theory In The Kernel Setting

We next present a result which computes the upper bound of the Rademacher average of a function class which is a ball $f \in \mathcal{H}, \|f\|_{\mathcal{H}} \leq t$ in the RKHS.

Theorem. Let $\mathcal{H}$ be a RKHS with kernel k and let $K = (k(x_i, x_j))) \in \mathbb{R}^{\aleph\times\aleph\times\aleph}$. Define $\mathcal{F}_t = \{f \in \mathcal{H}, \|f\|_{\mathcal{H}} \leq t\}$. Then we have

$$\hat{\mathcal{R}}_n(\mathcal{F}_t) := \mathbb{E}\left[\sup_{f\in\mathcal{F}_t}\frac{1}{n}\sum_{i=1}^{n}\epsilon_i f(x_i)|X_1,\cdots,X_n\right] \leq \frac{t}{n}\sqrt{trace(K)}$$

and

$$\mathcal{R}_n(\mathcal{F}_t) \leq \frac{t}{\sqrt{n}}\sqrt{\sum_{i=1}^{\infty}\lambda_i}$$

where λ_i's are the eigenvalues of the operator $T_k : f \to \int k(\cdot, x)f(x)dP(x)$

Proof.

By the reproducing property we have

$$\begin{aligned}\sup_{f\in\mathcal{F}_t}\frac{1}{n}\sum_{i=1}^{n}\epsilon_i f(x_i) &= \sup_{f\in\mathcal{F}_t}\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\langle k(x_i,\cdot), f\rangle \\ &= t\left\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i k(x_i,\cdot)\right\| \\ &= t\sqrt{\frac{1}{n^2}\sum_{i,j=1}^{n}\epsilon_i\epsilon_j k(x_i,x_j)}\end{aligned}$$

Thus

$$\begin{aligned}\hat{\mathcal{R}}_n((F)_t) &= \mathbb{E}\left[\frac{t}{n}\sqrt{\sum_{i,j=1}^{n}\epsilon_i\epsilon_j k(x_i,x_j)|X_1,\cdots,X_n}\right] \\ &= \frac{t}{n}\sqrt{\mathbb{E}\left[\sum_{i,j=1}^{n}\epsilon_i\epsilon_j k(x_i,x_j)|X_1,\cdots,X_n\right]} \\ &= \frac{t}{n}\sqrt{\sum_{i=1}^{n}k(x_i,x_i)} = \frac{t}{n}\sqrt{trace(K)}\end{aligned}$$

1.3 Bayesian View Point:Gaussian Process Regression

In this section, we following the setting in [2] where first consider a Bayesian view point, *i.e.* introducing a gaussian random field as a prior. Let us consider a centered Gaussian field on Ω with covariance function $\Lambda(x,y) := \mathbb{E}[\xi(x)\xi(y)]$, then consider

$$\begin{aligned}\mathcal{L}u(x) &= \xi(x), && x \in \Omega \\ \mathcal{B}u(x) &= 0, && x \in \partial\Omega\end{aligned}$$

The solution is a centered Gaussian field with covariance $\Gamma(x,y) = \mathbb{E}[u(x)u(y)]$ is $\Gamma(x,y) = \int_{\Omega^2} G(x,z)\Lambda(z,z')G(y,z')dzdz'$

Remark. If $\Lambda(x,y) = \int_\Omega G(x,z)G(y,z)dz$ and $\mathcal{L}^*\mathcal{L}\Gamma(x,y) = \delta(x-y)$

Conditioning of the solution posterior to the observation of N linear functions of $u(x)$:

$$\int_\Omega u(x)\psi_i(x)dx, i \in \{1, \cdots, N\}$$

Θ is defined as $\Theta_{i,j} = \int_{\Omega^2} \psi_i(x)\Gamma(x,y)\psi_j(y)dxdy$ which is the covariance matrix of the observation.

$$l^T\Theta l = \|v\|_\Lambda^2$$

where v is the solution of

$$\begin{aligned} \mathcal{L}^*u(x) &= \sum_{j=1}^N l_j\psi_j(x), && x \in \Omega \\ \mathcal{B}u(x) &= 0, && x \in \partial\Omega \end{aligned}$$

Here $\|f\| := \int_\Omega f(x)\Lambda(x,y)f(y)dxdy$

Using Gaussian Noise's motivation lies in the fact that for Gaussian fields, conditional expected values can be computed via linear projection, *i.e.*

$$\mathbb{E}[u(x)|\Phi] = \sum_{i=1}^N \Psi_i\phi_i(x)$$

here $\phi_i(x) := \sum_{j=1}^N \Theta_{i,j}^{-1}\int_\Omega \Gamma(x,y)\psi_j(y)dy$

Remark. The covariance is

$$\sigma^2(x,x) = \Gamma(x,x) - \sum_{i,j=1}^N \Theta_{i,j}^{-1}\int_\Omega \Gamma(x,y)\psi_j(y)dy\int_\Omega \Gamma(x,y)\psi_i(y)dy$$

Then we define $V := \{\phi \in \mathcal{H}(\Omega) | \mathcal{L}\phi \in L^2(\Omega), \mathcal{B}\phi = 0\}$ and a scalar product on V defined by

$$\langle u, v\rangle := \int_\Omega (\mathcal{L}u(x))(\mathcal{L}v(x))dx$$

Theorem. The space V and the reproducing kernel $\Gamma(x,y)$ forms a reproducing kernel Hilbert space, *i.e.* $\langle v, \Gamma(\cdot,x)\rangle = v(x)$

Remark. $\left\langle v, \int_\Omega \Gamma(\cdot,y)f(y)dy\right\rangle = \int_\Omega v(y)f(y)dy$

Define

$$V_i := \{\phi \in V | \int_\Omega \phi(x)\psi_i(x) = 1, \int_\Omega \phi(x)\psi_j(x) = 1, j \neq i\}$$

Consider the following optimization problem

$$\min \langle \phi, \phi\rangle \; s.t. \phi \in V_i$$

with unique minimizer

$$\phi_i(x) := \sum_{j=1}^N \Theta_{i,j}^- 1\int_\Omega \Gamma(x,y)\psi_i(y)dy$$

Consider $\theta_i(x) := \int_\Omega \Gamma(x,y)\psi_i(y)dy$ then

$$\mathcal{L}\theta_i(x) = \int_\Omega G(y,x)\psi_i(y)dy$$

Noting that $\|\mathcal{L}\theta_i\| = \Theta_{i,i}$

$$\int_\Omega \phi_i(x)\psi_i(x) = (\Theta^{-1}\cdot\Theta)_{i,j} = \delta_{i,j}$$

Remark. Note that ϕ_i is also equal to the expected value of $u(x)$ conditioned on $\int_\Omega u(x)\psi_i(x) = 1$ and $\int_\Omega u(x)\psi_j(x) = 0, j \neq i$, *i.e.*

$$\phi_i(x) = \mathbb{E}\left[u(x)|\int_\Omega u(x)\psi_i(x) = 1, \int_\Omega u(x)\psi_j(x) = 0, j \neq i\right]$$

Let $\mathcal{L}$ and $\mathcal{B}$ be the linear integro-differential operators on Ω and $\partial\Omega$, next we consider u as the solution of the itegro-differential equation:

$$\begin{aligned} \mathcal{L}u(x) &= g(x), && x \in \Omega \\ \mathcal{B}u(x) &= 0, && x \in \partial\Omega \end{aligned}$$

Then we will give the estimation error of the previous estimation

Pointwise Estimate

$$\left\| v(x) - \sum_{i=1}^N \phi_i(x)\left(\int_\Omega v(y)\psi_i(y)dy\right)\right\| \leq \sigma(x)\|x\|_V$$

where $\sigma^2(x)$ is the variance $\Gamma(x,x) - \sum_{i,j=1}^N \Theta_{i,j}^{-1}\int_\Omega \Gamma(x,y)\psi_j(y)dy\int_\Omega \Gamma(x,y)\psi_i(y)dy$

In particular, if u is the solution of the original integro-differential equation, then

$$\left\| u(x) - \sum_{i=1}^n \phi_i(x)\left(\int_\Omega u(y)\psi_i(y)\right)\right\| \leq \sigma(x)\|g\|_{L_2}$$

$\mathcal{H}(\Omega)$**-norm Estimates** Write

$$\rho(V_0) := \sup_{v\in V_0}\frac{\|v\|_{\mathcal{H}(\Omega)}}{\|v\|_V}$$

where $\|\cdot\|_{\mathcal{H}(\Omega)}$ is the natural norm associated with the space on which the operator $\mathcal{L}$ is defined.

$$\left\| v(x) - \sum_{i=1}^N \phi_i(x)\left(\int_\Omega v(y)\psi_i(y)dy\right)\right\|_{\mathcal{H}(\Omega)} \leq \rho(V_0)\|x\|_V$$

Proof. Write $v_\Psi(x) := \sum_{i=1}^N \phi_i(x)(\int_\Omega v(y)\psi_i(y)dy)$, then

$$\langle v, v\rangle \leq \|v - v_\psi\|_{\mathcal{H}(\Omega)} \leq \rho(V_0)\langle v - v_\Psi, v - v_\Psi\rangle^{\frac{1}{2}}$$

In [13, 14] utilizing the gaussian process to formulate this problem and utilizing the variance calculated to do active learning.

1.4 Approximation Or Concentration? Overfitting Or Perfect Fitting?

All the previous discussion are all about kernel ridge regression, *i.e.*

$$\min\sum_{i=1}^n l(f(x_i), y_i) + \|f\|_{\mathcal{H}}$$

The purpose is to constraint the hypothesis set into a bounded ball under the norm $\|f\|_{\mathcal{H}}$, in this section our discussion may seems contradict to the traditional statistical understanding. However, recent works have shown that directly interpolating *i.e.* solving the constrained optimization problem

$$\min \|f\|_{\mathcal{H}} \qquad\qquad s.t. f(x_i) = y_i, i = 1, 2, \cdots, n$$

i.e. in this section, we want to introduce the risk bounds for classification and regression rules that interpolate. In recent work [4], authors tried interpolating schemes in the kernel setting. From a statistical view point, the VC-dimension of the the kernel interpolating scheme is $+\infty$. (Also, the interpolating also contradict to the Rademacher complexity.) By traditional understanding, such kind of learning algorithms will not introduce "generalization property",however, which the core part of machine learning *i.e.* predicting on unseen data. However like [5], the similar generalization property is discovered. Later [6, 7, 8] build theory for the generalization theory of ridgeless regression, nonparametric interpolation. This is a field catching rising attention.

2 Kernel Selection

In previous section, I'm just taking the method of regression into consideration but not considering the selection of the kernel. Unlike the traditional discussion, in this section we are considering learning a kernel function from the data. Then in the next section, I will gives another view point of deep learning.

2.1 Kernel Selection Principle

In this section, I will introduce several learning objectives to decide "what is a good kernel"

- [9] wants the model trained to have a large margin.
- [10] wants the model have high data efficiency.
- [11] wants the model to have high entropy.
- [12] utilize the gaussian process view point and all his objective is to minimize the variance.
- [18] wants the kernel to have a low local Rademacher complexity.

2.2 Kernel Updating Rules

Here we introduce several ways to learn the kernel

In [11,12], they introduce a neural network $\phi(x)$ and the kernel $k(x, y)$ is defined as

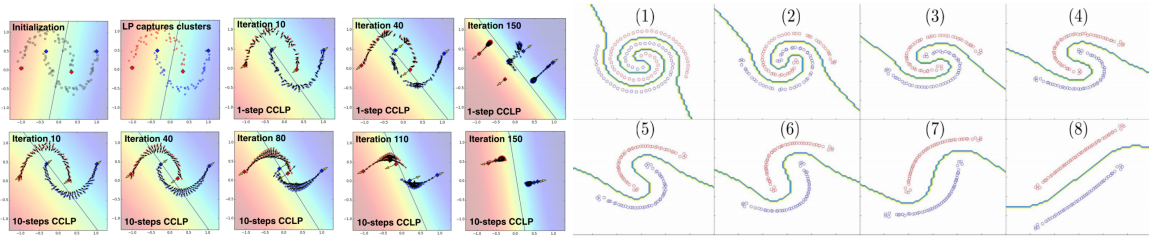
$$k(x, y) = \langle \phi(x), \phi(y) \rangle$$

and directly minimizing the objective functions.

In [9,10] they using a technique to move the data point and they using the previous kernel, *i.e.*

$$k(x, y) = k_G(x + f(x), y + f(y))$$

the difference is that [9] directly get the function form a calculation of curvature and [10] using the kernel regression.



3 The Deep Kernels

[15] first introduced a kernel formulation of the initialization of a deep neural nets. In their setting, the neural network is a concentration of a Computation skeleton. The initialization of the neural network is done by sampling a Guassian random variable.

Definition. (Dual activation and kernel.) The dual activation of an activation σ is the function $\hat{\sigma} : [-1, 1] \to \mathbb{R}$ defined as

$$\hat{\sigma}(\rho) = \mathbb{E}_{(X,Y)\sim N_\rho} \sigma(X)\sigma(Y)$$

Here N_ρ is a multivariate Gaussian distribution on $\mathbb{R}^2$ with mean 0 and covariance matrix $\left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$. The dual kernel w.r.t to a Hilbert space $\mathcal{H}$ is the kernel $\kappa_\sigma : \mathcal{H}^1 \times \mathcal{H}^1 \to \mathbb{R}$ defined as

$$\kappa_\sigma(x, y) = \hat{\sigma}(\langle x, y\rangle_{\mathcal{H}})$$

For a neural network which defined as a directed acyclic graph (DAG). We denote its nodes as $V(\mathcal{N})$ and edges $E(\mathcal{N})$. For all nodes, we define the $h_{v,w}$ recursively as

$$h_{v,w}(x) = \sigma_v(\sum_{u\in in(v)} \omega_{uv} h_{u,w}(x))$$

The we will introduce the dual kernel of the neural network

Definition. (Compositional kernels). Let $\mathcal{S}$ be a computation skeleton with normalized activations (the norm of an activation function is defined as $\|\sigma\| := \sqrt{\mathbb{E}_{X\sim N(0,1)}\sigma^2(X)}$) and (single) output node $\wr$. For every node v, inductively define a kernel $k_v : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as following. For an input node v corresponding to the ith coordinate, define $k_v(x, y) = \left\langle x^i, y^i \right\rangle$ and for a non-input node v, define

$$k_v(x, y) = \hat{\sigma}_v(\frac{\sum_{u\in in(v)} k_u(x, y)}{|in(v)|})$$

The final kernel κ_S is $\kappa_\wr$

In [15] they prove if you initialization the skeleton r times with

$$r \geq \frac{(4C^4)^{depth(\mathcal{S})+1} \log(8|\mathcal{S}|/\delta)}{\epsilon^2}$$

Then for all $x, x' \in \mathcal{X}$ with probability of at least $1 - \delta$ we have

$$|k_w(x, x') - k_{\mathcal{S}}(x, x')| \leq \epsilon$$

[18] analysis the kernel of single layer neural network with Relu activation in a analytic form

$$g(x, y) = (\frac{1}{2} - \frac{\arccos\langle x, y\rangle}{2\pi})\langle x, y\rangle$$

In fact, it is a dot-product kernel and its spectrum can be obtained through spherical harmonic decomposition

$$g(x, y) = \sum_{u=1}^{\infty} \gamma_u \phi_u(x)\phi_u(y)$$

and in the paper they show the $\Omega(m^{-1})$ eignvalue decay speed.

Later [16, 20] using this kernel perspective to prove the **global convergence** of the gradient descent algorithm. The basic idea of the proof is that **the kernel doesn't changes too large during training** thus the training loss is expotentially decay.

[19] tends to prove that the evolution of an ANN during training can also be described by a kernel: during gradient descent on the parameters of an ANN, the network function f_θ (which maps input vectors to output vectors) follows the kernel gradient of the functional cost (which is convex, in contrast to the parameter cost) w.r.t. a new kernel: the Neural Tangent Kernel (NTK). The *kernel gradient* is defined as

$$\nabla_K C|_{f_0}(x) = \frac{1}{N}\sum_{j=1}^{N} K(x, x_j) d|_{f_0}(x_j)$$

The NTK can be seen in `https://www.youtube.com/watch?v=raT2ECrvbag`.

4 Kernel View Of Semi-supervised Learning

From the previous discussion, It's easy to see that semi-supervised learning works because it learns a kernel fits the data distribution, the profit of the kernel needs to be discovered (like generalization, asymptotic of labeled and unlabeled data).

References

[1] Tuo ZHao, Lecture Note STAT 598/CSE8803 Advanced Machine Learning. `https://www2.isye.gatech.edu/~tzhao80/Lectures/8803.pdf`

[2] Owhadi H. Bayesian numerical homogenization[J]. Multiscale Modeling & Simulation, 2015, 13(3): 812-828.

[3] Bartlett P L, Mendelson S. Rademacher and Gaussian complexities: Risk bounds and structural results[J]. Journal of Machine Learning Research, 2002, 3(Nov): 463-482.

[4] Belkin M, Ma S, Mandal S. To understand deep learning we need to understand kernel learning ICML. 2018.

[5] Zhang C, Bengio S, Hardt M, et al. Understanding deep learning requires rethinking generalization ICLR2017.

[6] Belkin M, Hsu D, Mitra P. Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate. NIPS2018.

[7] Belkin M, Rakhlin A, Tsybakov A B. Does data interpolation contradict statistical optimality?[J]. arXiv preprint arXiv:1806.09471, 2018.

[8] Liang T, Rakhlin A. Just Interpolate: Kernel" Ridgeless" Regression Can Generalize[J]. arXiv preprint arXiv:1808.00387, 2018.

[9] Amari S, Wu S. Improving support vector machine classifiers by modifying kernel functions[J]. Neural Networks, 1999, 12(6): 783-789.

[10] Owhadi H, Yoo G R. Kernel Flows: from learning kernels from data into the abyss[J]. arXiv preprint arXiv:1808.04475, 2018.

[11] Kamnitsas K, Castro D C, Folgoc L L, et al. Semi-Supervised Learning via Compact Latent Space Clustering[J]. ICML, 2018.

[12] Jean N, Xie S M, Ermon S. Semi-supervised Deep Kernel Learning: Regression with Unlabeled Data by Minimizing Predictive Variance. NIPS 2018.

[13] Ng Y C, Colombo N, Silva R. Bayesian Semi-supervised Learning with Graph Gaussian Processes[C]//Advances in Neural Information Processing Systems. 2018: 1688-1699.

[14] Zhu X, Lafferty J, Ghahramani Z. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions[C]//ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining. 2003, 3.

[15] Daniely A, Frostig R, Singer Y. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity[C]//Advances In Neural Information Processing Systems. 2016: 2253-2261.

[16] Daniely A. SGD learns the conjugate kernel class of the network[C]//Advances in Neural Information Processing Systems. 2017: 2422-2430.

[17] Xie B, Liang Y, Song L. Diverse neural network learns true target functions. AISTAT 2018.

[18] Cortes C, Kloft M, Mohri M. Learning kernels using local rademacher complexity[C]//Advances in neural information processing systems. 2013: 2760-2768.

[19] Jacot A, Gabriel F, Hongler C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks[J]. arXiv preprint arXiv:1806.07572, 2018. (accepted by NIPS2018)

[20] Du S S, Lee J D, Li H, et al. Gradient descent finds global minima of deep neural networks[J]. arXiv preprint arXiv:1811.03804, 2018.