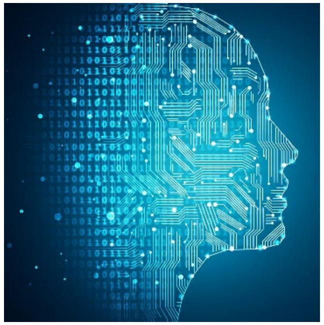
Machine Learning

Reproducing Kernel Hilbert Spaces

Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Sets

- What is a set?
- A set is a collection of objects/things
- Examples of things:
  - Numbers
  - Letters
  - Words/strings
  - Functions?
- We often like to think about sets of objects that share certain properties
  - E.g., the rational numbers, natural numbers, irrational numbers
- It can also be helpful to think about sets of functions that share certain properties
  - E.g., the set of all continuous functions, the set of all differentiable functions

Overview

- A reproducing kernel Hilbert space (RKHS) is a space (i.e. set) of real-valued functions defined in terms of a positive definite kernel.
- By optimizing over a RKHS, we can derive many ML algorithms

Rough definition

- The following can be made rigorous but we'll focus on intuition
- Let $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ where $\mathcal{X}$ is the input space.
- We previously stated the following are equivalent:
  1. k is a symmetric and positive definite kernel function
  2. $\exists$ an inner product space $\mathcal{H}$ and a feature map $\Phi: \mathcal{X} \rightarrow \mathcal{H}$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$
- We previously proved $(2) \Rightarrow (1)$.
- Now we'll prove $(1) \Rightarrow (2)$

A space of functions

- Define

$$\mathcal{H}_0 = \left\{ \sum_{i=1}^{m} \alpha_i k(\cdot, \boldsymbol{x}_i) \,\middle|\, m \in \mathbb{N}, \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathcal{X} \right\}$$

  - Note that in this case $k(\cdot, \boldsymbol{x}')$ can be viewed as a function $\boldsymbol{x} \mapsto k(\boldsymbol{x}, \boldsymbol{x}')$ in the first argument if we assume that $\boldsymbol{x}'$ is fixed
- From this point of view, $\mathcal{H}_0$ is a space of functions that map from $\mathcal{X}$ to $\mathbb{R}$.
- For example, if k is the Gaussian kernel, then the following depicts an element of $\mathcal{H}_0$

An inner product

- We want to define an inner product for $\mathcal{H}_0$
- That means we need to define an inner product between *functions:*

$$\left\langle \sum_{i=1}^{m} \alpha_i k(\cdot, \boldsymbol{x}_i), \sum_{j=1}^{n} \beta_j k(\cdot, \boldsymbol{x}'_j) \right\rangle \coloneqq \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}'_j)$$

- It can be shown that this is a valid inner product

The feature map

- Consider the feature map $\Phi: \mathcal{X} \to \mathcal{H}_0$ given by

$$\Phi(\boldsymbol{x}) = k(\cdot, \boldsymbol{x})$$

- Be definition of the inner product, we get

$$\begin{aligned} \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle &= \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}') \rangle \\ &= k(\boldsymbol{x}, \boldsymbol{x}'). \end{aligned}$$

- This establishes (2) from three slides ago that given a kernel, we can find a corresponding inner product space and feature map such that the inner product of the feature space is equal to the kernel function
- Φ in this case is the *canonical feature map*
- This procedure works for any SPD kernel

The reproducing property

- The reproducing property states that for any $f \in \mathcal{H}_0$ and $\boldsymbol{x} \in \mathcal{X}$,
$$f(\boldsymbol{x}) = \langle f, k(\cdot, \boldsymbol{x}) \rangle.$$

- To see this since $f \in \mathcal{H}_0$, we can write $f = \sum_{i=1}^{n} \alpha_i k(\cdot, \boldsymbol{x}_i)$

- Then
$$\begin{aligned}
\langle f, k(\cdot, \boldsymbol{x}) \rangle &= \left\langle \sum_{i=1}^{n} \alpha_i k(\cdot, \boldsymbol{x}_i), k(\cdot, \boldsymbol{x}) \right\rangle \\
&= \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) \\
&= \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) \\
&= f(\boldsymbol{x})
\end{aligned}$$

Some functional analysis

- For technical reasons, we need to enlarge $\mathcal{H}_0$ slightly
- Recall the definition of a *vector space*
  - A collection of objects (called vectors) that can be added together and multiplied by scalars
- $\mathcal{H}_0$ is a vector space where the "vectors" are functions
- $\mathcal{H}_0$ is also an *inner product space*
  - A vector space with an associated inner product
- Let $\mathcal{H}$ be the *completion* of $\mathcal{H}_0$
  - Add to $\mathcal{H}_0$ all functions $g \notin \mathcal{H}_0$ that can be approximated by functions in $\mathcal{H}_0$ with arbitrary accuracy
  - This guarantees that $\mathcal{H}$ is a *Hilbert space* (which is defined as a complete inner product space)

RKHS

- It can be shown that the reproducing property still holds for all $f \in \mathcal{H}$
- $\mathcal{H}$ is known as the *reproducing kernel Hilbert space* associated with the SPD kernel k
- k is called the *reproducing kernel* of $\mathcal{H}$

The Representer Theorem

- Previously, we derived kernel methods by optimizing over a class of linear models and then kernelizing
- Alternatively, we can optimize over the RKHS directly
- Even though an RKHS may be infinite dimensional, optimization problems of a certain type reduce to finite-dimensional problems

The Representer Theorem

Theorem: Let $\mathcal{H}$ be an RKHS consisting of functions defined on $\mathcal{X}$. Consider an optimization problem of the form

$$\min_{f\in\mathcal{H}} J(f)$$

where

$$J(f) = L\big(f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_n)\big) + \Lambda\big(\|f\|_{\mathcal{H}}^2\big)$$

for some $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, and where Λ is nondecreasing and $\|f\|_{\mathcal{H}}^2 = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$. Then there exists a minimizer of the form

$$f = \sum_{i=1}^{n} \alpha_i k(\cdot, \boldsymbol{x}_i).$$

Furthermore, if Λ is strictly increasing, then <u>every</u> minimizer has this form.

The Representer Theorem

- **Remark**: The notation $L\big(f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_n)\big)$ indicates that this term does not depend on values of f outside of $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$.

- **Proof**: Uses the projection theorem from linear algebra and the reproducing property. See notes at http://web.eecs.umich.edu/~cscott/past_courses/eecs545f16/31_rkhs.pdf

Example: SVM

- Let

$$L\big(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\big) = \frac{1}{n}\sum_{i=1}^{n} \max\big(0, 1 - y_i f(\boldsymbol{x}_i)\big)$$

$$\Lambda(\|f\|^2) = \frac{\lambda}{2}\|f\|^2$$

- By the representer theorem, the minimizer has the form

$$f = \sum_{i=1}^{n} r_i k(\cdot, \boldsymbol{x}_i), \qquad r_i \in \mathbb{R}$$

- Denoting $\boldsymbol{r} = [r_1, \ldots, r_n]^T$ and substituting $C = \frac{1}{\lambda}$ reduces the optimization problem to

$$\min_{\boldsymbol{r}} \frac{C}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \sum_{j=1}^{n} r_j k\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big)\right) + \frac{1}{2}\boldsymbol{r}^T K \boldsymbol{r}$$

Example: SVM

- Introducing slack variables makes this equivalent to

$$\min_{\boldsymbol{r},\xi} \quad \frac{1}{2} r^T K r + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

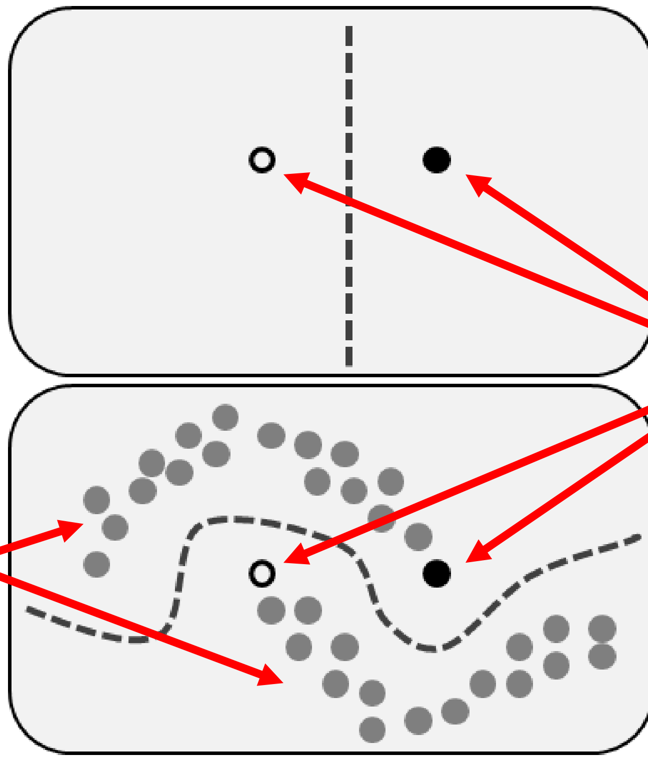
$$s.t. \quad y_i \sum_{j=1}^{n} r_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

- This is a convex, differentiable optimization problem with affine constraints so strong duality holds. The KKT conditions also hold.
- It an be shown that applying Lagrangian dual theory and the KKT conditions recovers the SVM dual problem (without offset)

Example: Semi-supervised learning

- Semi-supervised learning: we have both labeled and unlabeled samples
- **Goal**: leverage unlabeled data to improve the performance of a method that uses only labeled data
- Classification example:



https://commons.wikimedia.org/w/index.php?curid=19514958

Example: Semi-supervised learning

- Let $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m), \boldsymbol{x}_{m+1}, \dots, \boldsymbol{x}_{m+n}$ denote the training data
- Consider a regression problem
- One approach: create a weighted adjacency matrix $W = [w_{ij}]_{i,j=m+1}^{m+n}$ from the unlabeled data.
  - Include this to force the regression function to have similar values at similar points

Example: Semi-supervised learning

- A possible optimization problem for regression:

$$\min_{f \in \mathcal{H}} \lambda \|f\|_{\mathcal{H}}^2 + \frac{1}{m}\sum_{i=1}^{m} \left(y_i - f(\boldsymbol{x}_i)\right)^2 + \frac{\gamma}{2}\sum_{i,j=m+1}^{m+n} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2$$

- Last term can be written as $\frac{\gamma}{2}\boldsymbol{f}^T W \boldsymbol{f}$ where

$$f = [f(\boldsymbol{x}_{m+1}), \ldots, f(\boldsymbol{x}_{m+n})]^T$$

- The solution can then be derived using the representer theorem

Final Remarks

- The representer theorem can be applied to derive other kernel methods in a lot of different settings
  - **Examples**: kernel ridge regression, kernel logistic regression, one-class SVM (see the Michigan lecture notes for these examples)

- Other approaches for semi-supervised learning exist

Further reading

- Michigan lecture notes: http://web.eecs.umich.edu/~cscott/past_courses/eecs545f16/31_rkhs.pdf
- ESL Sections 5.8 and 12.3.3