SM2 Assessed Coursework

Rachel Wood

2023-03-14

This portfolio considers a model for n observations $\{(y_i^0, x_i^0)\} \in \mathbb{R} \times \mathbb{R}^p$ defined by

$$Y_i^0 \sim f(y; \mu_i, \phi)dy, \qquad g(\mu_i) = \alpha + f(x_i^0), \qquad \text{for} i = 1, \dots, n \tag{1}$$

where $\alpha \in \mathbb{R},\ \phi \in (0, \infty)$ and $f \in \mathcal{F} = \mathcal{H}_k$.

Here $(\mathcal{H}_{||}, \langle \cdot, \cdot \rangle)$ is a reproducible kernel Hilbert space (RKHS) with positive semi definite kernel k. Then we have that $\mathcal{H}_k$ satisfies the reproducibility property

$$f(x) = \langle f, k(x, \cdot) \rangle \qquad \forall x \in \mathcal{X}, \forall f \in \mathcal{H}_k \tag{2}$$

Part 1

Question 1

Here we consider the identifiability of (1). A kernel will lead to identifiable models if and only if the corresponding RKHS does not contain constant functions.

Unidentifiable kernel: We can take the kernel to be

$$k(x, y) = 1$$

which is positive semi-definite, we can see that taking $\mathcal{H}_k = \{f : f(x) = c\ \forall x\}$ and $\langle f, g \rangle_k = fg$ satisfies the reproducing property:

$$f(x) = \langle f,\ k(x, \cdot) \rangle = \langle f, 1 \rangle = f \qquad \forall x \in \mathcal{X},\ \forall f \in \mathcal{H}_k$$

and thus $(\mathcal{H}_k, \langle \cdot, \cdot \rangle_k)$ is the corresponding RKHS. Since $\mathcal{H}_k$ is made up of constant functions, k does not produce identifiable models.

Identifiable Kernel: An example of an identifiable kernel is the Gaussian kernel:

$$k_\lambda(x, x') = \exp\left(-\frac{||x - x'||^2}{\lambda}\right)$$

Any function in the corresponding RKHS $f \in \mathcal{H}_k$ must satisfy:

$$f(x) = \langle f, k(x, \cdot) \rangle = \frac{1}{\lambda} \left\langle f, -\exp||x - \cdot||^2 \right\rangle$$

It is clear that there is no constant function f (excepting the zero function) which satisfies this. Hence the Gaussian kernel leads to identifiable models.

Question 2

For this question, we consider the solution $\hat{f}_\lambda$ to the optimisation problem

$$(\hat{\alpha_\lambda}, \hat{\phi_\lambda}, \hat{f_\lambda}) \in \underset{\alpha\in\mathbb{R},\phi\in(0,\infty),f\in\mathcal{H}_k}{\text{argmax}} \frac{1}{2n} \sum_{i=1}^{n} \log f\left(y_i; g^{-1}\left(\alpha + f(x_i^0)\right), \phi\right) - \lambda||f||^2_{\mathcal{H}_k} \tag{3}$$

We assume $\mathcal{H}_k = \tilde{\mathcal{H}_n} \oplus \tilde{\mathcal{H}_n}^\perp$, hence we can write $\hat{f}_\lambda = f_1 + f_2$ where $f_1 \in \tilde{\mathcal{H}_n}$ and $f_2 \in \tilde{\mathcal{H}_n}^\perp$. Thus there exists coefficients $\hat{\beta}_\lambda = \left(\hat{\beta}_{\lambda,1}, \dots, \hat{\beta}_{\lambda,n}\right) \in \mathbb{R}^n$ such that

$$f_1 = \sum_{i=1}^{n} \hat{\beta}_{\lambda,i} k(x_i^0, \cdot)$$

and we can write $f = \sum_{i=1}^n \hat{\beta}_{\lambda,i} k(x_i^0, \cdot) + f_2$. Further, by the reproducing property, for any x_j:

$$f(x_j) = \left\langle \sum_{i=1}^{n} \hat{\beta}_{\lambda,i} k(x_i^0, \cdot) + f_2,\ k(x_j^0, \cdot) \right\rangle = \sum_{i=1}^{n} \hat{\beta}_{\lambda,i} \left\langle k(x_i^0, \cdot), k(x_j^0, \cdot) \right\rangle = \sum_{i=1}^{n} \hat{\beta}_{\lambda,i} k(x_i^0, x_j^0)$$

and so $f(x_j)$ does not depend on f_2 and as a consequence, the first term in (3) also does not depend on f_2. Hence to choose f_2 we only need to consider minimizing the regularisation term. Using that $\langle f_1, f_2 \rangle = 0$, we see

$$||f||^2_{\mathcal{H}_k} = \langle f_1 + f_2, f_1 + f_2 \rangle = ||f_1||^2_{\mathcal{H}_k} + ||f_2||^2_{\mathcal{H}_k} \geq ||f_1||^2_{\mathcal{H}_k}$$

with the last inequality becoming an equality when $f_2 = 0$. Hence the minimiser $\hat{f}_\lambda$ must have $f_2 = 0$ and can be written as

$$\hat{f_\lambda} = \sum_{i=1}^{n} \hat{\beta}_{\lambda,i} k(x_i^0, \cdot) \tag{4}$$

Question 3

We now want to substitute the results of (4) into (3). The regularisation term becomes:

$$\begin{aligned}
||f||^2_{\mathcal{H}_k} &= \left\langle \sum_{i=1}^{n} \hat{\beta}_{\lambda,i} k(x_i^0, \cdot), \sum_{i=j}^{n} \hat{\beta}_{\lambda,j} k(x_j^0, \cdot) \right\rangle \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \hat{\beta}_{\lambda,i} \left\langle k(x_i^0, \cdot), k(x_j^0, \cdot) \right\rangle \hat{\beta}_{\lambda,j} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \hat{\beta}_{\lambda,i} k(x_i^0, x_j^0) \hat{\beta}_{\lambda,j} \\
&= \hat{\beta}_\lambda^T K \hat{\beta}_\lambda
\end{aligned}$$

and so (3) becomes

$$(\hat{\alpha_\lambda}, \hat{\phi_\lambda}, \hat{\beta_\lambda}) \in \underset{\alpha\in\mathbb{R},\phi\in(0,\infty),\beta\in\mathbb{R}^n}{\text{argmax}} \frac{1}{2n} \sum_{i=1}^{n} \log f\left(y_i; g^{-1}\left(\alpha + \sum_{j=1}^{n} \beta_i k(x_i^0, x_j^0)\right), \phi\right) - \lambda \beta^T K \beta \tag{5}$$

Question 4

Given $m \leq n+2$, we want to obtain an m-dimensional problem. Then $d = m-2$ is the length of the new $\tilde{\beta}_\lambda$ vector to estimate.

The Nyström method approximates k by $\tilde{f}^{(m)} = k^{(m)}$, given by

$$\tilde{k}^{(d)}(x, x') = k_d(x)^T K_{d,d}^{-1}\ k_d(x')$$

where $K_{d,d} \in \mathbb{R}^{d\times d}$ is the first d rows and columns of the gram matrix K and $k_d(x) = (k(x_1^0, x), \ldots, k(x_d^0, x))$

Then we can write $f(x)$ as

$$f = \sum_{i=1}^{n} \beta_i\ \tilde{k}_d(x_i^0, \cdot) = \sum_{i=1}^{n} \beta_i\ k_d(x_i^0)^T K_{d,d}^{-1}\ k_d(\cdot) = \left(\sum_{i=1}^{n} \beta_i\ k_d(x_i^0)^T K_{d,d}^{-1} \right) k_d(\cdot)$$

and take the new vector of coefficients to be:

$$\tilde{\beta}^T = \sum_{i=1}^{n} \beta_i\ k_d(x_i^0)^T \left(K_d^0\right)^{-1} = \beta^T K_{n,d} K_{d,d}^{-1}$$

where $K_{n,d} \in \mathbb{R}^{n\times d}$ is the first d columns of K We can now rewrite f as:

$$f = \tilde{\beta}^T k_d(\cdot) = \sum_{j=1}^{d} \tilde{\beta}_j k(x_j, \cdot)$$

We now consider the regularisation term, again we substitute $\tilde{k}^{(d)}$:

$$\beta^T \tilde{K}^{(d)} \beta = \beta^T K_{n,d}\ K_{d,d}^{-1}\ K_{n,d}^T\ \beta = \left(\beta^T K_{n,d}\ K_{d,d}^{-1}\right) K_{d,d}^T \left(K_{d,d}^{-T}\ K_{n,d}\ \beta\right) = \tilde{\beta}^T K_{d,d}^T\ \tilde{\beta}$$

Hence we can now write our m-dimension optimisation problem:

$$(\hat{\alpha_\lambda}, \hat{\phi_\lambda}, \tilde{\beta_\lambda}) \in \operatorname*{argmax}_{\alpha\in\mathbb{R}, \phi\in(0,\infty), \tilde{\beta}\in\mathbb{R}^d} \frac{1}{2n} \sum_{i=1}^{n} \log f \left(y_i; g^{-1} \left(\alpha + \sum_{j=1}^{d} \tilde{\beta}_j k(x_j, x_i) \right), \phi \right) - \lambda \tilde{\beta}^T K_{d,d}^T\ \tilde{\beta} \tag{6}$$

Question 5

For this question we first need to obtain the penalty term as $\omega^T\omega$ for some vector ω. We can first obtain the eigen-decomposition of $K_{d,d}^T$:

$$K_{d,d}^T = V\Lambda V^T$$

Then writing $\Lambda = \Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}} = \Lambda^{\frac{1}{2}}\left(\Lambda^{\frac{1}{2}}\right)^T$, we get

$$\tilde{\beta}^T K_{d,d}\ \tilde{\beta} = \tilde{\beta}^T V \Lambda^{\frac{1}{2}} \left(\Lambda^{\frac{1}{2}}\right)^T V^T \tilde{\beta} = \omega^T \omega$$

where $\omega^T = \tilde{\beta}^T V \Lambda^{\frac{1}{2}}$, or equivalently $\tilde{\beta}^T = \omega^T \Lambda^{-\frac{1}{2}} V^T$.

Then our objective function becomes

$$(\hat{\alpha_\lambda}, \hat{\phi}_\lambda, \hat{\omega}_\lambda) \in \underset{\alpha\in\mathbb{R},\phi\in(0,\infty),\omega\in\mathbb{R}^d}{\operatorname{argmax}} \frac{1}{2n}\sum_{i=1}^{n} \log f\left(y_i; g^{-1}\left(\alpha+\omega^T\Lambda^{-\frac{1}{2}}V^T K_{d,n}\right), \phi\right) - \lambda\omega^T\omega \tag{7}$$

so we can use $X' = \Lambda^{-\frac{1}{2}}V^T K$, where K is the Gram matrix of the original data, as the input for `glmnet()` and this will give the estimated ω.

Part2

This section implements the theory from Part 1 on the `wesdr` data from the `gss` package, which has the binary response variable `ret` and explanatory variables `dur`, `gly` and `bmi`

```
library(dplyr)
library(kableExtra)
library(gss)
data("wesdr")
head(wesdr)
```

```
##    dur  gly  bmi ret
## 1 10.3 13.7 23.8   0
## 2  9.9 13.5 23.5   0
## 3 15.6 13.8 24.8   0
## 4 26.0 13.0 21.6   1
## 5 13.8 11.1 24.6   1
## 6 31.1 11.3 24.6   1
```

Question 6

We can now use `glmnet` to write a function implementing Question 4 and 5:

```
library(kernlab)
estimate <- function(X, y, lambda, kernel, m){
  d <- m - 2
  K <- kernelMatrix(kernel = kernel, x = X)
  K_d <- K[1:d, 1:d]

  K_eigen <- eigen(K_d)
  V <- K_eigen$vectors
  Lambda <- (K_eigen$values)


  X_new <- diag(1/sqrt(Lambda)) %*% t(V) %*% K

  fit <- glmnet(X_new, y, family = "binomial", alpha = 0, lambda = lambda)
  summary(fit)
  alpha <- fit$beta[1]
  omega <- fit$beta[-1]
}
```

```r
X <- as.matrix(wesdr[,1:3])
y <- wesdr[,4]



#estimate(X, y, lambda = 0.3, kernel = rbfdot(sigma = 1), m = 300)
```