S&DS 365 / 665

Intermediate Machine Learning

Smoothing and Kernels

Wednesday, February 2

Yale

Topics for today

- Recap: Smoothing methods
- Demo of smoothing with various kernels
- Mercer kernels

Nonparametric Regression

Given $(X_1, Y_1), \ldots, (X_n, Y_n)$ predict Y from X.

Assume only that $Y_i = m(X_i) + \epsilon_i$ where where $m(x)$ is a smooth function of x.

The most popular methods are *kernel methods*. However, there are two types of kernels:

1. Smoothing kernels
2. Mercer kernels

Smoothing kernels involve local averaging.
Mercer kernels involve regularization.

Smoothing Kernels

- Smoothing kernel estimator:

$$\widehat{m}_h(x) = \frac{\sum_{i=1}^n Y_i \, K_h(X_i, x)}{\sum_{i=1}^n K_h(X_i, x)}$$

where $K_h(x, z)$ is a *kernel* such as

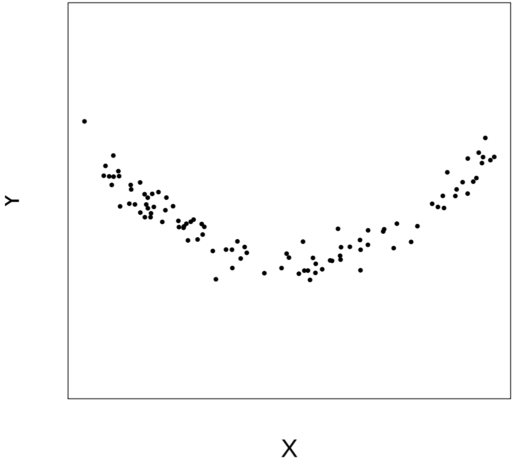
$$K_h(x, z) = \exp\left(-\frac{\|x - z\|^2}{2h^2}\right)$$

and $h > 0$ is called the *bandwidth*.

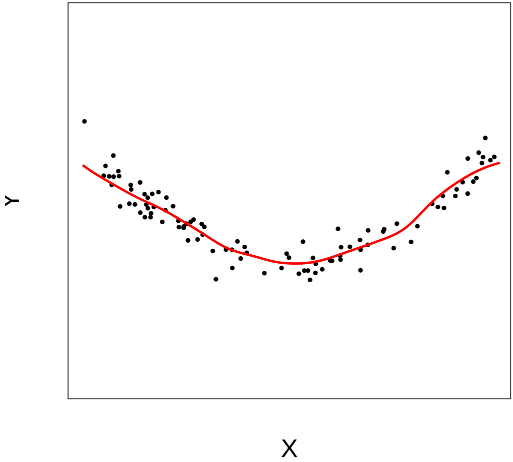
- $\widehat{m}_h(x)$ is just a local average of the Y_i's near x.

- The bandwidth h controls the bias-variance tradeoff:
Small h = large variance while *large h = large bias*.

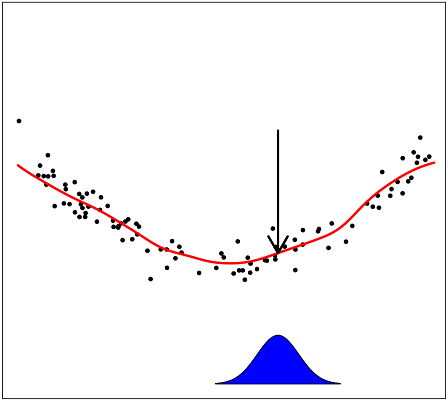
Example: Some Data – Plot of Y_i versus X_i



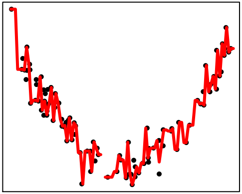
Example: $\widehat{m}(x)$



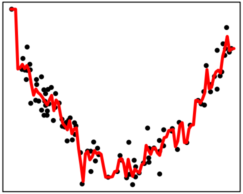
$\widehat{m}(x)$ is a local average



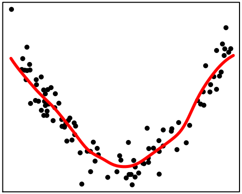
Effect of the bandwidth h



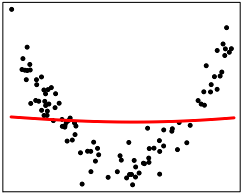
very small bandwidth



small bandwidth



medium bandwidth



large bandwidth

Smoothing Kernels

Risk $= \mathbb{E}(Y - \widehat{m}_h(X))^2 = \text{bias}^2 + \text{variance} + \sigma^2$.

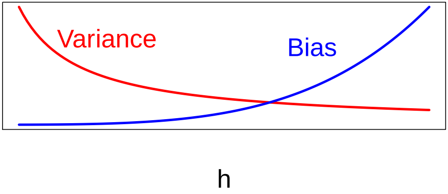
bias$^2 \approx h^4$,

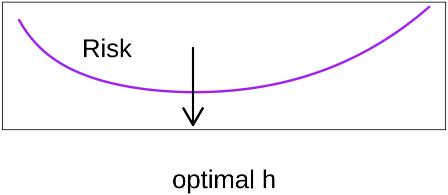
variance $\approx \dfrac{1}{nh^p}$ where p = dimension of X.

$\sigma^2 = \mathbb{E}(Y - m(X))^2$ is the unavoidable prediction error.

small h: low bias, high variance (undersmoothing)
large h: high bias, low variance (oversmoothing)

Risk Versus Bandwidth





Estimating the Risk: Cross-Validation

To choose h we need to estimate the risk $R(h)$. We can estimate the risk by using *cross-validation*.

1. Omit (X_i, Y_i) to get $\widehat{m}_{h,(i)}$, then predict: $\widehat{Y}_{(i)} = \widehat{m}_{h,(i)}(X_i)$.
2. Repeat this for all observations.
3. The cross-validation estimate of risk is:

$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_{(i)})^2.$$

Shortcut formula: Whenever $\widehat{Y} = LY$ we can use the shortcut

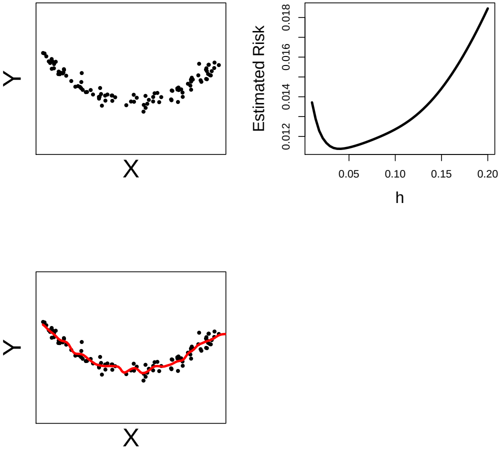
$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{Y}_i}{1 - L_{ii}}\right)^2.$$

In this case $L_{ii} = K_h(X_i, X_i)/\sum_t K_h(X_i, X_t)$.

Summary so far

1. Compute $\widehat{m}_h$ for each h.
2. Estimate the risk $\widehat{R}(h)$.
3. Choose bandwidth $\widehat{h}$ to minimize $\widehat{R}(h)$.
4. Let $\widehat{m}(x) = \widehat{m}_{\widehat{h}}(x)$.

Example



Let’s revisit the notebook

Another Approach: Mercer Kernels

Instead of using local smoothing, we can optimize the fit to the data subject to regularization (penalization). Choose $\widehat{m}$ to minimize

$$\sum_i (Y_i - \widehat{m}(X_i))^2 + \lambda \,\mathrm{penalty}(\widehat{\mathrm{m}})$$

where penalty($\widehat{\mathrm{m}}$) is a *roughness penalty*.

λ is a parameter that controls the amount of smoothing.

How do we construct a penalty that measures roughness? One approach is: *Mercer Kernels* and *RKHS = Reproducing Kernel Hilbert Spaces.*

What is a Mercer Kernel?

A *Mercer Kernel* $K(x, x')$ is symmetric and positive definite:

$$\int \int f(x)f(x')K(x, x')\, dx\, dx' \geq 0 \quad \text{for all } f.$$

Example: $K(x, x') = e^{-||x-x'||^2/2}$.

Think of $K(x, x')$ as the *similarity* between x and x'. We will create a set of *basis functions* based on K.

Fix z and think of $K(z, x)$ as a function of x. That is,

$$K(z, x) = K_z(x)$$

is a function of the second argument, with the first argument fixed.

Mercer Kernels

Let

$$\mathcal{F} = \left\{ f(\cdot) = \sum_{j=1}^{k} \beta_j K(z_j, \cdot) \right\}$$

Define a norm: $\|f\|_K = \sum_j \sum_k \beta_j \beta_k K(z_j, z_k)$. $\|f\|_K$ *small means f smooth.*

If $f = \sum_r \alpha_r K(z_r, \cdot)$, $g = \sum_s \beta_s K(w_s, \cdot)$, the inner product is

$$\langle f, g \rangle_K = \sum_r \sum_s \alpha_r \beta_s K(z_r, w_s).$$

$\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS) because

$$\langle f, K(x, \cdot) \rangle = f(x)$$

Nonparametric Regression: Mercer Kernels

Representer Theorem

Let $\widehat{m}$ minimize

$$J(m) = \sum_{i=1}^{n} (Y_i - m(X_i))^2 + \lambda \|m\|_K^2.$$

Then

$$\widehat{m}(x) = \sum_{i=1}^{n} \alpha_i K(X_i, x)$$

for some $\alpha_1, \ldots, \alpha_n$.

So, we only need to find the coefficients

$$\alpha = (\alpha_1, \ldots, \alpha_n).$$

Nonparametric Regression: Mercer Kernels

Plug $\widehat{m}(x) = \sum_{i=1}^{n} \alpha_i K(X_i, x)$ into J:

$$J(\alpha) = \|Y - \mathbb{K}\alpha\|^2 + \lambda\alpha^T \mathbb{K}\alpha$$

where $\mathbb{K}_{jk} = K(X_j, X_k)$

Now we find α to minimize J. We get: $\widehat{\alpha} = (\mathbb{K} + \lambda I)^{-1} Y$ and $\widehat{m}(x) = \sum_i \widehat{\alpha}_i K(X_i, x)$.

The estimator depends on the amount of regularization λ. Again, there is a bias-variance tradeoff. We choose λ by cross-validation. This is like the bandwidth in smoothing kernel regression.

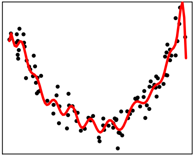
Smoothing Kernels *Versus* Mercer Kernels

Smoothing kernels: the bandwidth h controls the amount of smoothing.

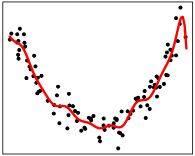
Mercer kernels: norm $\|f\|_K$ controls the amount of smoothing.

In practice these two methods give answers that are very similar.

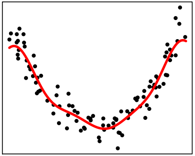
Mercer Kernels: Examples



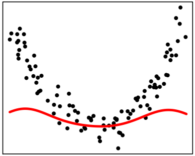
very small lambda



small lambda



medium lambda



large lambda

Multiple Regression

Both methods extend easily to the case where X has dimension $p > 1$. For example, just use

$$K(x, x') = e^{-\|x-x'\|^2/2}.$$

However, this is hard to interpret and is subject to the curse of dimensionality. This means that the *statistical performance* and the *computational complexity* degrade as dimension p increases.

An alternative is to use something less nonparametric such as additive model where we restrict $m(x_1, \ldots, x_p)$ to be of the form:

$$m(x_1, \ldots, x_p) = \beta_0 + \sum_j m_j(x_j).$$

Additive Models

Model: $m(x) = \beta_0 + \sum_{j=1}^{p} m_j(x_j)$.

We can take $\widehat{\beta}_0 = \overline{Y}$ and we will ignore β_0 from now on.

We want to minimize

$$\sum_{i=1}^{n} \left(Y_i - (m_1(X_{i1}) + \cdots + m_p(X_{ip})) \right)^2$$

subject to m_j smooth.

Additive models: Backfitting Algorithm

Input: Data (X_i, Y_i)

Iterate until convergence:

For each $j = 1, \ldots, p$:

Compute residual: $R_j = Y - \sum_{k \neq j} \widehat{m}_k(X_k)$

Smooth $\widehat{m}_j = \mathcal{S}_j R_j$

Output: Estimator $\widehat{m}(X_i) = \sum_j \widehat{m}_j(X_{ij})$.

Here, $\mathcal{S}_j R$ is any 1-dimensional nonparametric regression smoother

But what if p is large?

Sparse Additive Models

Additive Model: $Y_i = \sum_{j=1}^{p} m_j(X_{ij}) + \varepsilon_i, \quad i = 1, \ldots, n$

High dimensional: $n \ll p$, with most $m_j = 0$.

Optimization:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\Big(Y - \textstyle\sum_j m_j(X_j)\Big)^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} \sqrt{\mathbb{E}(m_j^2)} \le L_n \\ & \mathbb{E}(m_j) = 0 \end{aligned}$$

This generalizes the lasso!

---

https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2009.00718.x

Sparse Backfitting Algorithm

Input: Data (X_i, Y_i), regularization parameter λ.

Iterate until convergence:

For each $j = 1, \ldots, p$:

Compute residual: $R_j = Y - \sum_{k \neq j} \widehat{m}_k(X_k)$

Smooth $\widehat{m}_j = \mathcal{S}_j R_j$

Estimate norm: $s_j = \sqrt{\mathbb{E}(\widehat{m}_j^2)}$

Soft-threshold: $\widehat{m}_j \leftarrow \left[1 - \dfrac{\lambda}{\widehat{s}_j}\right]_+ \widehat{m}_j$

Output: Estimator $\widehat{m}(X_i) = \sum_j \widehat{m}_j(X_{ij})$.

This generalizes coordinate descent algorithm from last time.

Example: Boston Housing Data

Predict house value Y from 10 covariates.

We added 20 irrelevant (random) covariates to test the method.

$Y =$ house value; $n = 506$, $p = 30$.

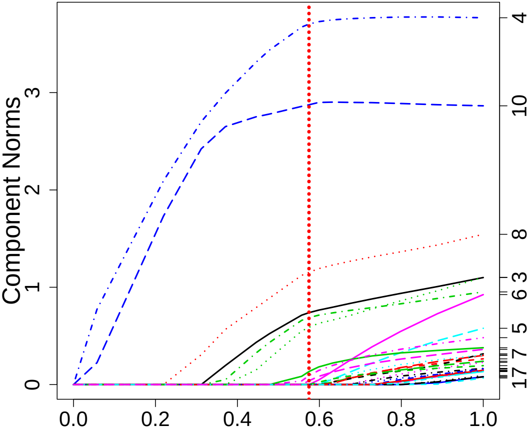
$$Y = \beta_0 + m_1(\text{crime}) + m_2(\text{tax}) + \cdots + \cdots m_{30}(X_{30}) + \epsilon.$$

Note that $m_{11} = \cdots = m_{30} = 0$.

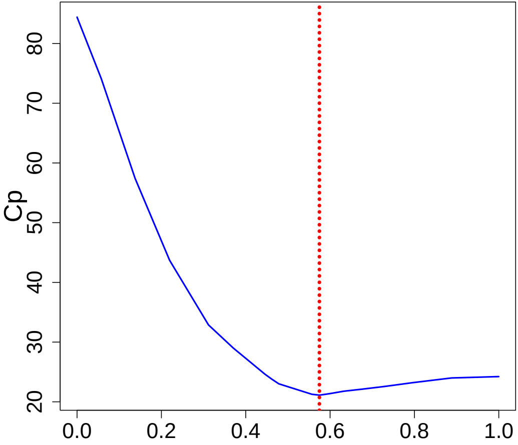
We choose λ by minimizing the estimated risk.

SpAM yields 6 nonzero functions. It correctly reports that $\widehat{m}_{11} = \cdots = \widehat{m}_{30} = 0$.

L_2 norms of fitted functions versus $1/\lambda$



Estimated Risk Versus λ



Summary for today

- Smoothing methods compute local averages, weighting points by a kernel
- The details of the kernel don't matter much
- Mercer kernels using penalization rather than smoothing
- Defining property: Matrix $\mathbb{K}$ is always positive-definite (non-negative eigenvalues)
- Equivalent to a type of ridge regression in function space
- The curse of dimensionality limits use of both approaches to low dimensions
- A compromise between nonparametric and linear models is to use additive models