Kernel Smoothing Methods
The Elements of statistical learning - Chapter 6

Li Yuqi
2015.4.26
Overview

- Kernel Smoothers
- Local regression models
- Local Likelihood and Other Models
- Kernel Density Estimation and Classification
- Radial Basis Functions and Kernels
6.1 One-Dimensional Kernel Smoothers

- k-nearest-neighbor average to estimate $\mathbb{E}(Y|X = x)$

\[
\hat{f}(x) = \text{Ave}(y_i | x_i \in N_k(x))
\]
An Example

• Nadaraya-Watson kernel-weighted average

\[ \hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i)y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)} \]

with the Epanechnikov quadratic kernel

\[ K_\lambda(x_0, x) = D \left( \frac{|x-x_0|}{\lambda} \right) \]

with

\[ D(t) = \begin{cases} 
\frac{3}{4}(1-t^2) & \text{if } |t| \leq 1; \\
0 & \text{otherwise}. 
\end{cases} \]
Kernel definition

\[ K_\lambda(x_0, x) = D \left( \frac{|x - x_0|}{h_\lambda(x_0)} \right). \]

- \( D \): a decreasing function
- \( h_\lambda(.) \)
  - Constant: metric window widths
  - a scaling function: \( h_k(x_0) = |x_0 - x[k]| \)
Some kinds of kernel

<table>
<thead>
<tr>
<th>Name</th>
<th>$D(t)$</th>
<th>$h\lambda$</th>
<th>Compact or not</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform kernel</td>
<td>$D(t) = I(</td>
<td>t</td>
<td>\leq 1)$</td>
</tr>
<tr>
<td>Epanecnikov Quadratic Kernel</td>
<td>$D(t) = \frac{3}{4}(1-t^2)I(</td>
<td>t</td>
<td>\leq 1)$</td>
</tr>
<tr>
<td>Tri-Cube Kernel</td>
<td>$D(t) = (1-</td>
<td>t</td>
<td>^3)^3I(</td>
</tr>
<tr>
<td>Gaussian Kernel</td>
<td>$D(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$</td>
<td>$h_\lambda = \lambda$</td>
<td>Not</td>
</tr>
</tbody>
</table>
Some kinds of kernel

![Graph showing different kernel functions](image-url)

- Epanechnikov
- Tri-cube
- Gaussian
Local Linear Regression

• Problem
  – locally-weighted averages can be badly biased on the boundaries of the domain (due to the asymmetry)
Local Linear Regression

• Local weighted linear regression
  – Solve weighted least squares at target point $x_0$
    \[
    \min_{\alpha(x_0), \beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) [y_i - \alpha(x_0) - \beta(x_0)x_i]^2.
    \]
  – The estimate is
    \[
    \hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0
    \]
  – Solution (equivalent kernel):
    \[
    \hat{f}(x_0) = b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0)y \\
    = \sum_{i=1}^{N} l_i(x_0) y_i.
    \]
  • where $b(x)^T = (1, x)$, $B$ is $N \times 2$ matrix with ith row $b(x_i)^T$, $W(x_0)$ is $N \times N$ diagonal matrix with ith diagonal element $K_\lambda(x_0, x_i)$
Equivalent Kernel
Equivalent Kernel

\[
\mathbb{E} \hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0) f(x_i)
\]

\[
= f(x_0) \sum_{i=1}^{N} l_i(x_0) + f'(x_0) \sum_{i=1}^{N} (x_i - x_0) l_i(x_0)
\]

\[
+ \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + R,
\]

• where \( \sum_{i=1}^{N} l_i(x_0) = 1 \) and \( \sum_{i=1}^{N} (x_i - x_0) l_i(x_0) = 0 \).

• Then the bias \( \mathbb{E} \hat{f}(x_0) - f(x_0) \) depends only on quadratic and higher-order terms in the expansion of \( f \).
Local Polynomial Regression

- Local polynomial fits of any degree:

\[
\min_{\alpha(x_0), \beta_j(x_0), j=1,...,d} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[ y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0) x_i^j \right]^2
\]

- Bias-variance tradeoff:
  As degree increases, bias is reduced and variance is increased
6.2 Selecting the Width of the Kernel

• $\lambda$ is a parameter that controls its width
  – The window is narrow, variance will be large while bias will tend to be small
  – The window is wide, variance will be small while bias will be higher
6.3 Local Regression in IR^p

Let \( b(X) \) be a vector of polynomial terms in \( X \) of maximum degree \( d \).

At each \( x_0 \in \mathbb{R}^p \) solve

\[
\min_{\beta(x_0)} \sum_{i=1}^{N} K_{\lambda}(x_0, x_i) (y_i - b(x_i)^T \beta(x_0))^2
\]

to produce the fit \( \hat{f}(x_0) = b(x_0)^T \hat{\beta}(x_0) \). Typically the kernel will be a radial function, such as the radial Epanechnikov or tri-cube kernel

\[
K_{\lambda}(x_0, x) = D \left( \frac{||x - x_0||}{\lambda} \right)
\]

\( || \cdot || \) is the Euclidean norm
6.3 Local Regression in IR$^p$

![Aortic Diameter vs Age](chart)
6.4 Structured Local Regression Models in IR\(^p\)

- Structured Kernels
  - use a positive semidefinite matrix \( A \) to weigh the different coordinates:

\[
K_{\lambda, A}(x_0, x) = D \left( \frac{(x - x_0)^T A (x - x_0)}{\lambda} \right)
\]
6.4 Structured Local Regression Models in \( IR^p \)

- Structured Regression Functions

\[
E(Y|X) = f(X_1, X_2, \ldots, X_p)
\]

- Analysis-of-variance decompositions

\[
f(X_1, X_2, \ldots, X_p) = \alpha + \sum_j g_j(X_j) + \sum_{k<\ell} g_{k\ell}(X_k, X_\ell) + \ldots
\]

- Varying coefficient models

\[
f(X) = \alpha(Z) + \beta_1(Z)X_1 + \cdots + \beta_q(Z)X_q.
\]

- divide the \( p \) predictors in \( X \) into a set \((X_1, X_2, \ldots, X_q)\) with \( q < p \), and the remainder is collected in the vector \( Z \).

- Fit the model by locally weighted least squares
6.5 Local Likelihood and Other Models

- each observation $y_i$ has a parameter $\theta_i = \theta(x_i) = x_i^T \beta$
  - Then the log-likelihood:

$$l(\beta) = \sum_{i=1}^{N} l(y_i, x_i^T \beta). \quad l(\beta(x_0)) = \sum_{i=1}^{N} K_{\lambda}(x_0, x_i) l(y_i, x_i^T \beta(x_0)).$$

- A varying coefficient model

$$l(\theta(z_0)) = \sum_{i=1}^{N} K_{\lambda}(z_0, z_i) l(y_i, \eta(x_i, \theta(z_0))).$$
Local multiclass logistic regression model

• Form:

\[
\Pr(G = j | X = x) = \frac{e^{\beta_j^0 + \beta_j^T x}}{1 + \sum_{k=1}^{J-1} e^{\beta_k^0 + \beta_k^T x}}.
\]

• Log-likelihood:

\[
\ell(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta), \quad (4.19)
\]

where \( p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta) \).

\[
\sum_{i=1}^{N} K_\lambda(x_0, x_i) \left\{ \beta_{g_i0}(x_0) + \beta_{g_i}(x_0)^T (x_i - x_0) \right. \\
- \log \left[ 1 + \sum_{k=1}^{J-1} \exp \left( \beta_{k0}(x_0) + \beta_k(x_0)^T (x_i - x_0) \right) \right] \right\}
\]
6.6 Kernel Density Estimation and Classification

- Kernels Density Estimation

\[
\hat{f}_X(x_0) = \frac{\text{#}x_i \in N(x_0)}{N\lambda} \quad \Rightarrow \quad \hat{f}_X(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \quad \text{Parzen estimate}
\]

- With $K_\lambda$ is Gaussian kernel, then:

\[
\hat{f}_X(x) = \frac{1}{N} \sum_{i=1}^{N} \phi_\lambda(x - x_i)
\]

\[
= (\hat{F} \ast \phi_\lambda)(x),
\]

- In $\mathbb{R}^p$, the Gaussian density estimate is:

\[
\hat{f}_X(x_0) = \frac{1}{N(2\lambda^2 \pi)^{p/2}} \sum_{i=1}^{N} e^{-\frac{1}{2}(\|x_i - x_0\|/\lambda)^2}
\]
6.6 Kernel Density Estimation and Classification

- Kernels Density Classification
  - use nonparametric density estimates for classification

\[
\hat{\Pr}(G = j | X = x_0) = \frac{\hat{\pi}_j \hat{f}_j(x_0)}{\sum_{k=1}^{J} \hat{\pi}_k \hat{f}_k(x_0)}
\]
6.6 Kernel Density Estimation and Classification

- The Naïve Bayes Classifier
  - assumes that given a class $G = j$, the features $X_k$ are independent:

$$f_j(X) = \prod_{k=1}^{p} f_{jk}(X_k)$$

- We can derive the logit-transform (using class J as the base):

$$\log \frac{\Pr(G = \ell | X)}{\Pr(G = J | X)} = \log \frac{\pi_{\ell} f_{\ell}(X)}{\pi_{J} f_{J}(X)} = \log \frac{\pi_{\ell} \prod_{k=1}^{p} f_{\ell k}(X_k)}{\pi_{J} \prod_{k=1}^{p} f_{Jk}(X_k)}$$

$$= \log \frac{\pi_{\ell}}{\pi_{J}} + \sum_{k=1}^{p} \log \frac{f_{\ell k}(X_k)}{f_{Jk}(X_k)}$$

$$= \alpha_{\ell} + \sum_{k=1}^{p} g_{\ell k}(X_k).$$
6.8 Mixture Models for Density Estimation and Classification

- The Gaussian mixture model

\[ f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \Sigma_m) \quad \text{with} \quad \sum_m \alpha_m = 1 \]
6.8 Mixture Models for Density Estimation and Classification

Estimate the probability that observation \( i \) belongs to component \( m \)

\[
\hat{r}_{im} = \frac{\hat{\alpha}_m \phi(x_i; \hat{\mu}_m, \hat{\Sigma}_m)}{\sum_{k=1}^{M} \hat{\alpha}_k \phi(x_i; \hat{\mu}_k, \hat{\Sigma}_k)},
\]

Use threshold for classification

<table>
<thead>
<tr>
<th>CHD</th>
<th>Mixture model</th>
<th>( \hat{\delta} = 0 )</th>
<th>( \hat{\delta} = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>232</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>76</td>
<td>84</td>
<td></td>
</tr>
</tbody>
</table>
Mixture Models for Classification

Estimate the probability that observation $i$ belongs to component $m$

$$\hat{r}_{im} = \frac{\hat{\alpha}_m \phi(x_i; \hat{\mu}_m, \hat{\Sigma}_m)}{\sum_{k=1}^{M} \hat{\alpha}_k \phi(x_i; \hat{\mu}_k, \hat{\Sigma}_k)},$$

Use threshold for classification

<table>
<thead>
<tr>
<th></th>
<th>Mixture model</th>
<th>$\hat{\delta} = 0$</th>
<th>$\hat{\delta} = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHD No</td>
<td>232</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>CHD Yes</td>
<td>76</td>
<td>84</td>
<td></td>
</tr>
</tbody>
</table>
6.7 Radial Basis Functions and Kernels

- Expansions in basis functions

\[
 f(x) = \sum_{j=1}^{M} \beta_j h_j(x) \quad \Rightarrow \quad f(x) = \sum_{j=1}^{M} K\lambda_j(\xi_j, x) \beta_j = \sum_{j=1}^{M} D\left(\frac{||x - \xi_j||}{\lambda_j}\right) \beta_j,
\]

- Optimize the sum-of-squares with respect to all the parameters:

\[
 \min_{\{\lambda_j, \xi_j, \beta_j\}} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{M} \beta_j \exp\left\{ -\frac{(x_i - \xi_j)^T(x_i - \xi_j)}{\lambda_j^2} \right\} \right)^2
\]
6.7 Radial Basis Functions and Kernels

• Problem
  – leave holes

• Solution
  – Renormalized radial basis function
    \[ h_j(x) = \frac{D(\|x - \xi_j\|/\lambda)}{\sum_{k=1}^{M} D(\|x - \xi_k\|/\lambda)} \]

  • Nadaraya–Watson kernel regression estimator
    \[ \hat{f}(x_0) = \sum_{i=1}^{N} y_i \frac{K_{\lambda}(x_0, x_i)}{\sum_{i=1}^{N} K_{\lambda}(x_0, x_i)} = \sum_{i=1}^{N} y_i h_i(x_0) \]
Thanks!