Institute of Mathematics, EPFL MATH524

Chapter 2: From linear to nonlinear regression

Lecturer: Rajita Chandak *Spring 2025*

In this section, we develop models and tools using nonlinear methods to show how we can start to generalize our estimation methods from the simple linear models.

As always, we start with the following data setup: $\{X_i, Y_i\}_{i=1}^n$ are n i.i.d. observations of the data governed by the model:

$$Y_i = r(X_i) + \varepsilon_i, \qquad \mathbb{E}[\varepsilon_i] = 0.$$

Here r is called the regression function and X the feature/covariate. In linear models, we make the very strong assumption that r is a linear function. However, as we know, this assumption does not always apply. We want to estimate (or in machine learning terms "*learn*") the function r with weak (or minimal) assumptions. The estimate of the the regression function, given by $\hat{r}_n$, is often called the smoother in this case.

To start, we will make the assumption that the variance of the error $\mathbb{V}(\varepsilon_i|X_i) = \mathbb{V}(\varepsilon_i) = \sigma^2$ is a constant that does not depend on X_i. Furthermore, we will assume the covariate (or feature) is fixed as $X_i = x_i$ for now. To generalize this to random covariates, the interpretation of $\hat{r}_n$ would change to a conditional expectation: $\hat{r}_n(X) = \mathbb{E}[Y|X]$. In practice, all the models and theory we will establish in this section for the fixed covariate setting will apply to the random covariate setting. For simplicity, we will deal with the fixed covariate setting. Differences in the two methods will be outlined where relevant.

1 Linear smoothers

We first define a linear smoother:

Definition 1 (Linear smoothers)**.** *The estimator $\hat{r}_n$ is a linear smoother of r if for each x, there exists a vector $l(x) = (l_1(x), \dots, l_n(x))$ such that*

$$\hat{r}_n(x) = \sum_{i=1}^n l_i(x) Y_i.$$

Then, the fitted values are given by $\mathbf{r} = (\hat{r}_n(x_1), \dots, \hat{r}_n(x_n))$ which is equivalent to

$$\mathbf{r} = LY$$

where $Y = (Y_1, \dots, Y_n)$ and L is an $n \times n$ matrix where the *i-th* row is given by $l(x_i)$.

Definition 2 (Smoothing matrix and effective kernel)**.** *L is defined as the smoothing matrix and the i-th row of L is called the effective kernel for estimating $r(x_i)$.*

Note that the smoothers are constructed such that for all x, $\sum_i l_i(x) = 1$ and so constant curves are preserved under this smoothing. That is, if $Y_i = c$ for all i, $\hat{r}_n(x) = c$.

Example 1 (Local averaging)**.** *Suppose we have some known constant $h > 0$ and we define the ball around x, $B_x = \{i : |x_i - x| < h\}$ and the number of points in B_x, $n_x = \|B_x\|$. Then, consider the estimator*

$$\hat{r}_n(x) = \frac{1}{n_x} \sum_{i \in B_x} Y_i.$$

This is known as the local averaging estimator, a special case of the kernel estimator (more on this in the next lecture).

1.1 MSE and bias-variance trade-off

The risk function is simply the average of the pointwise loss function,

$$R = \mathbb{E}[L(r(x), \hat{r}_n(x))].$$

There are different loss functions L that may be of interest based on the application. Here we list a few common loss functions:

1. Log-loss (classification): $L(Y_i, \hat{Y}_i) = -\frac{1}{n} \sum_i Y_i \log \hat{Y}_i$.
2. Hinge loss (classification): $L(Y_i, \hat{Y}_i) = \max\{0, 1 - Y_i \hat{Y}_i\}$
3. Mean absolute error: $L(Y_i, \hat{Y}_i) = -\frac{1}{n} \sum_i |Y_i - \hat{Y}_i|$
4. Huber loss:

$$L(Y_i, \hat{Y}_i; \delta) = \begin{cases} \frac{1}{2}(Y_i - \hat{Y}_i)^2 & \text{if } |Y_i - \hat{Y}_i| \leq \delta \\ \delta |Y_i - \hat{Y}_i| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

The most popular loss function for regression problems is the squared error loss. We will use **risk** and **MSE** interchangeably to refer to this quantity.

Recall the bias-variance trade-off the comes from the decomposition of the MSE.

$$MSE = R = \text{bias}^2 + \mathbb{V}$$

where

$$\text{bias} = \mathbb{E}[\hat{r}_n] - r(x)$$

and

$$\mathbb{V} = \mathbb{V}(\hat{r}_n).$$

Note that in the above local averaging example, we have a hyper-parameter h that needs to be determined apriori somehow. h is also often referred to as the smoothing parameter. A common method for estimating h is by minimizing the MSE, similar to how the parameters of linear models are chosen (recall OLS). The ideal formulation for choosing h would be to minimize

$$R(h) = \mathbb{E}\left[\frac{1}{n}\sum_i (\hat{r}_n(x_i) - r(x_i))^2\right],$$

but since r is unknown, a naive estimator of $R(h)$, the *training error*, could be used instead:

$$\hat{R}(h) = \frac{1}{n}\sum_i (Y_i - \hat{r}_n(x_i))^2.$$

However, it can be shown that this estimator is downwards biased and can lead to undersmoothing (or in ML terms, "*overfitting*") since each data point is used twice: once in estimating $\hat{r}$ and once in estimating the risk $\hat{R}(h)$.

Since the training error MSE estimate is known to be inefficient, in practice cross-validation is used to estimate the risk function.

Definition 3 (leave-one-out cross validation)**.** *The cross-validation estimate for selecting h is done by minimizing*

$$\hat{R}_{cv}(h) = \frac{1}{n}\sum_i (Y_i - \hat{r}_{-i}(x_i))^2,$$

where $\hat{r}_{-i}$ is the estimator of r obtained when omitting the i-th data point.

$$\hat{r}_{-i}(x) = \sum_j Y_j l_{j,-i}(x),$$

where

$$l_{j,-i} = \begin{cases} \frac{l_j(x)}{\sum_{k\neq i} l_k(x)} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases}$$

Note that this formulation of $l_{j,-i}$ sets the weight of x_i to 0 and re-normalizes the other weights such that the property $\sum_i l_i(x) = 1$ is preserved.

The formula for $\hat{R}_{\text{cv}}$, as is currently given is algorithmically very slow to compute (order of n^2). The following reformulation speeds up the computation by a factor of n:

$$\hat{R}_{\text{cv}} = \frac{1}{n}\sum_i \left(\frac{Y_i - \hat{r}_n(x_i)}{1 - L_{ii}}\right)^2,$$

where L_{ii} is the *i-th* diagonal entry of L.

Note that neither definition of $\hat{R}_{\text{cv}}$ implies the existence of a well-defined minimum that can determine the parameter h. One must be careful in characterizing the estimator for a reasonable range of values and selecting the empirically optimal value for h.

Another estimator for $R(h)$ is a simple generalization of the leave-one-out estimator, called the **generalized cross-validation** estimator, $\hat{R}_{\text{gcv}}$.

$$\hat{R}_{\text{gcv}} = \frac{1}{n}\sum_i \left(\frac{Y_i - \hat{r}_n(x_i)}{1 - \nu/n}\right)^2,$$

where $\nu = tr(L)$ is the effective degrees of freedom.

2 Local regression

The idea of linear smoothers provides a more general estimation model than the classical linear models. While there are many different types of smoothers that can be employed, here we will focus on a specific collection of smoothers for which we can establish strong theoretical properties and analyze limiting behavior, the collection of weighted average estimators.

We will start with the Nadaraya-Watson kernel estimator.

Definition 4 (Nadaraya-Watson kernel estimator)**.** *Let $h > 0$ be a constant called the bandwidth. The Nadaraya-Watson estimator is defined as*

$$\hat{r}_n = \frac{\sum_i K\left(\frac{x-x_i}{h}\right) Y_i}{\sum_j K\left(\frac{x-x_j}{h}\right)}.$$

Note here if we use the linear smoother formulation from before we will set $l_i(x) = K((x - x_i)/h)/\sum_j K((x - x_j)/h)$.

In general, the magnitude of the chosen bandwidth h will determine how smooth the estimator is. A smaller h will lead to larger fluctuations between estimation points and thus a more wiggly estimator which a larger bandwidth will smooth out the variance between neighboring points and provide a smoother estimate. As discussed before, the optimal choice for h is usually determined by the empirical minimizing the mean squared error and can often depend on the sample size, i.e., $h = h_n$.

Of course here we also have to address the question of which kernel function to use. The most common kernel functions are listed below.

- Boxcar: $K(u) = \frac{1}{2}\mathbf{1}(|u| < 1)$
- Gaussian: $K(u) = \frac{1}{\sqrt{2\pi}} \exp^{-u^2/2}$
- Epanechnikov: $K(u) = \frac{3}{4}(1 - u^2)\mathbf{1}(|u| < 1)$
- Tricube: $K(u) = \frac{70}{81}(1 - |u|^3)^3\mathbf{1}(|u| < 1)$

These kernels are all considered **second-order** kernels since they have the following properties:

$$\int K(u)\mathrm{d}u = 1 \qquad \int uK(u)\mathrm{d}u = 0 \qquad \int u^2K(u)\mathrm{d}u = \sigma_K^2 < \infty.$$

In practice the choice of the kernel function does not largely shift the resulting estimates, however, certain kernel functions may be preferred by researchers based on the mathematical properties of the kernel that allow for easier analysis of properties of the regression function estimator. It turns out that the choice of h is significantly more influential than the choice of K .

2.1 Consistency

We need to now ensure that this class of kernel estimators is actually useful in practice. One method of showing this is through proving consistency of the estimator. We will do this with a very popular result that applies to a more general class of functions than just kernel estimators. This theorem was first published in Stone (1977).

Theorem 1 (Stone's theorem)
Consider the weighted average regression function estimator:

$$\hat{r}(x) = \sum_i W_{ni}(x) Y_i$$

Assume the following conditions:

1. *There exists a constant C such that for every Borel measurable function $g : \mathbb{R} \to \mathbb{R}$ with $\mathbb{E}[|g(x)|] < \infty$,*
$$\mathbb{E}\left[\sum |W_{ni}(x_i)||g(x_i)|\right] \leq C\mathbb{E}[|g(x)|]$$
for all $n \geq 1$.
2. *There exists a constant $D \geq 1$ such that*
$$\mathbb{P}(\sum_i |W_{ni}(x)| \leq D) = 1 \ \forall n \geq 1.$$
3. *For all $\varepsilon > 0$, as $n \to \infty$,*
$$\sum_i |W_{ni}(x)| \mathbf{1}(||x_i - x|| > \varepsilon) \xrightarrow{p} 0.$$
4. $\sum_i W_{ni}(x) \xrightarrow{p} 1$
5. $\max_i |W_{ni}(x)| \xrightarrow{p} 0$ *as $n \to \infty$.*

Then, for all $p \geq 1$ and $\mathbb{E}[||Y||^p] < \infty$, then the corresponding weighted regression function estimate $\hat{r}_n$ is L_p consistent,

$$\mathbb{E}[||\hat{r}_n(x) - r(x)||^p] \xrightarrow{n\to\infty} 0.$$

Furthermore, the converse statement is also true.

Note that if the weights W_{ni} satisfy $W_{ni}(x) \geq 0,\ 1 \leq i \leq n$ and $\sum_i W_{ni} = 1$ then conditions (2) and (4) are automatically satisfied. The N-W estimator is a particular case of this, with the weights prescribed by the kernel function, that satisfies the remaining constraints and therefore is L_p consistent by Stone's theorem.

An important remark to consider is that by the DCT we can replace condition (5) in Theorem 1 with $\mathbb{E}[\sum_i W_{ni}^2(x)] \to 0$, which in some cases may be easier to verify.

It can be shown that minimizing the risk of the Nadaraya-Watson estimator with respect to h gives an optimal bandwidth of order $n^{-1/5}$. This implies that the risk decreases polynomially in n. When this is compared to the MLE rate of convergence in parametric models $(1/n)$, clearly the cost of using a more general, non-parametric model is reflected in the slower rate of convergence.

3 Local polynomial regression

In the exercises we will see that the Nadaraya-Watson estimator has a larger bias near the boundary of the support than at any interior point. This phenomenon is widely known as the **boundary bias**. Local polynomial regression was proposed as a modification to reduce the effect of this boundary bias.

Lets first consider some linear estimator $\hat{r}_n$ that minimizes the sum of squared errors (SSE). It is easy to see that the resulting estimator will be a constant function, $\hat{r}_n = \bar{Y}$, equivalent to the global mean of the response variables. We already know that this estimator will perform very poorly on any non-linear data generating process. Now, if we consider the kernel weight function $K_h(x_i; x) = K((x_i - x)/h)/h$ and instead choose an estimator that minimizes the weighted SSE

$$\sum_i K_h(x_i; x)(Y_i - \hat{r}_n)^2,$$

we will find that the new optimal estimator takes the form

$$\hat{r}_n = \frac{\sum_i K_h(x_i; x) Y_i}{\sum_i K_h(x_i; x)},$$

which is exactly the Nadaraya-Watson estimator. This alternative construction of the N-W estimator allows us to think of the estimator as a *local constant* estimator, derived by minimizing *locally weighted least squares error*. This interpretation leads to the natural question of whether using a *local polynomial* estimator of some order $p > 1$ will perform better than the local constant estimator. Let us formulate this general local polynomial estimator now. Denote the polynomial basis function of order p as

$$p(u) = \left[1, u, \frac{u^2}{2!}, \ldots, \frac{u^p}{p!}\right]$$

Then, the aim is to find the optimal coefficient vector β such that the weighted SSE is minimized near some evaluation point x:

$$\min_{\beta \in \mathbb{R}^p} \sum_i K_h(x_i, x)(Y_i - \beta^{\mathrm{T}} p(x_i - x))^2.$$

This generalization introduces a new parameter, p, that must be chosen apriori. It turns out that choosing p to be some odd number provides optimal bias reduction.

The local polynomial regression estimator can be written in the following matrix form that allows us to identify some of its nice properties. Let P denote the polynomial matrix:

$$P(\mathbf{u}) = [p(u_1); p(u_2); \ldots; p(u_n)]$$

and W denote the $n \times n$ diagonal weights matrix:

$$W_h(x) = \text{diag}([K_h(x_1; x), K_h(x_2, x), \ldots, K_h(x_n; x)]).$$

Then, the SSE can be written as

$$(Y - P(\mathbf{x} - x)^{\mathrm{T}}\beta)W_h(Y - P(\mathbf{x} - x)^{\mathrm{T}}\beta)$$

and the optimal coefficients of estimator as

$$\hat{\beta} = (P(\mathbf{x} - x)^{\mathrm{T}} W_h(x) P(\mathbf{x} - x))^{-1} P(\mathbf{x} - x) W_h(x) Y,$$

where $\mathbf{x} = (x_1, \ldots, x_n)$.

4 Splines

Instead of studying the optimal estimators that arise out of minimizing weighted least squares, like in the previous section, we can consider a different modification of the loss function that we have seen previously in linear regression: **regularization**. We can control the overfitting of the estimator by introducing some regularize that controls the complexity of the estimator.

$$\sum_i (Y_i - \hat{r}_n)^2 + \lambda J(r) \tag{1}$$

where λ is regularization constant that controls the trade-off between the fit and complexity and $J(r)$ is a function that quantifies the complexity of the estimator in some sense. This is analogous to the Lasso and Ridge regression formulations that use L_1 and L_2 norm functions for J to regularize the OLS estimator. Note that setting $\lambda = 0$ trivially returns the interpolating estimator (assuming the minimization is done over the same class of functions).

Consider the following piecewise polynomial formulation:

Definition 5 (M-spline)**.** *Let $a < \xi_1 < \ldots < \xi_k < b$ be a set of ordered points (knots). A M-spline is a continuous function f such that the following properties hold:*

(i) f is a $(M - 1)$ degree polynomial within each interval $(\xi_1, \xi_2), \ldots, (\xi_{k-1}, \xi_k)$,

(ii) f has $(M - 2)$ continuous derivatives at the knots $(\xi_1, \ldots, \xi_k)$.

A spline that is linear out side of the (a, b) knots is called a ***natural spline****.*

It turns out that the function that minimizes (1) is precisely a **natural cubic spline** ($M = 4$) with knots at the data points $(x_1, \dots, x_n)$.

We now need to construct the exact expression of this cubic spline that determines $\hat{r}_n$. Let $h_1(x) = 1, h_2(x) = x, h_3(x) = x^2, h_j(x) = (x - \xi_{j-4})^3_+$ for $j = 4, \dots, k+4$. Then, the collection of $\{h_j\}$ form a basis for the cubic splines at the knots, called the **truncated power basis**. Then, the cubic spline is written as

$$r(x) = \sum_{j=1}^{k+4} \beta_j h_j(x).$$

While this method may seem starkly different in comparison to the kernel and local polynomial approaches we studied before, Silverman (1984) proved that infact the spline estimator is very similar to the kernel estimator with

$$l_i(x) \approx \frac{1}{f(x_i)h(x_i)} K\left(\frac{x_i - x}{h(x_i)}\right)$$

where $f(x_i)$ is the density of the covariate x_i (the covariates are considered to be random in this case), $h(x) = (\lambda/(nF(x)))^{1/4}$ and

$$K(u) = \frac{1}{2}\exp\Big(-\frac{u}{\sqrt{2}}\Big)\sin\left(\frac{|u|}{\sqrt{2}} + \frac{\pi}{4}\right).$$

5 Confidence intervals

The natural next step of identifying a new type of estimator for the regression function is to be able to provide reasonable confidence intervals of the estimator to better understand the effectiveness. Furthermore, for most practical applications, we need a 'good' estimate of the variance in order to be able to construct the confidence intervals.

The following theorem establishes a consistent variance estimator for linear smoothers under the assumption of homoskadasticity (i.e., the variance σ^2 does not depend on x).

Theorem 2 (Variance estimation)
Assume the data generating process is determined by $Y_i = r(x_i) + \varepsilon_i$ with $\mathbb{E}[\varepsilon_i] = 0$ and $\mathbb{V}(\varepsilon_i) = \sigma^2$. Let $\hat{r}_n(x)$ be the linear smoother evaluated at some point x. Define

$$\hat{\sigma}^2 = \frac{\sum_i (Y_i - \hat{r}(x_i))^2}{n - 2\nu + \tilde{\nu}},$$

where $\nu = tr(L)$, $\tilde{\nu} = tr(L^{\mathrm{T}}L)$. When r is sufficiently smooth, $\nu = o(n)$ and $\tilde{\nu} = o(n)$, $\hat{\sigma}^2$ is a consistent estimator of σ^2.

Proof. Recall the following property of matrices: If Q is a symmetric matrix and Y is a random vector,

$$\mathbb{E}[Y^{\mathrm{T}}QY] = tr(Q\mathbb{V}(Y)) + \mathbb{E}[Y]^{\mathrm{T}}Q\mathbb{E}[Y].$$

Furthermore, recall that with $\mathbf{r} = [\hat{r}_n(x_1), \ldots, \hat{r}_n(x_n)]$ we can write $Y - \mathbf{r} = Y - LY = (I - L)Y$, where I is the identity matrix. Then, we can re-write the estimator

$$\hat{\sigma}^2 = \frac{Y^{\mathrm{T}}(I-L)^{\mathrm{T}}(I-L)Y}{tr((I-L)^{\mathrm{T}}(I-L))}.$$

Thus,

$$\mathbb{E}[\hat{\sigma}^2] = \frac{\mathbb{E}[Y^{\mathrm{T}}(I-L)^{\mathrm{T}}(I-L)Y]}{tr((I-L)^{\mathrm{T}}(I-L))} = \sigma^2 + \frac{\mathbf{r}^{\mathrm{T}}(I-L)^{\mathrm{T}}(I-L)\mathbf{r}}{n - 2\nu + \tilde{\nu}}.$$

Now note that if the assumptions of the theorem are satisfied, the second term in the above equation will converge to 0 as $n \to \infty$ and so $\hat{\sigma}$ is a consistent estimator.

Similar matrix manipulation can be done to show that $\mathbb{V}(\hat{\sigma}^2) \to 0$. ■

Now that we have a consistent estimator of the variance, we can look at the standard confidence interval formulation:

$$\hat{r}_n(x) \pm z_\alpha \hat{\sigma},$$

where z_α is the critical value determined based on the size of the desired confidence interval(α) and the assumption that the CLT applies.

This assumption that the CLT applies is critical, and we will see now why this confidence interval construction is not as straightforward for non-parametric estimators as it was for parametric (linear) estimators. Consider the studentized statistic,

$$\frac{\hat{r}_n(x) - r(x)}{\hat{\sigma}(x)} = \frac{\hat{r}_n(x) - \mathbb{E}[\hat{r}(x)]}{\hat{\sigma}(x)} + \frac{\mathbb{E}[\hat{r}(x)] - r(x)}{\hat{\sigma}(x)} = Z_n(x) + \frac{\mathrm{Bias}(\hat{r}_n(x))}{\sqrt{\mathrm{var}(\hat{r}_n(x))}}, \tag{2}$$

where Z_n is converges in distribution to a standard normal. In parametric settings, the second term converges to zero since the bias is smaller than the standard deviation of the estimator. However, recall that in selecting the bandwidth of our non-parametric estimator, we attempted to balance the trade-off between the bias and the variance terms of the MSE. This means that both terms are of similar order and that in practice this second term in (2) will not vanish to zero as $n \to \infty$, but rather show up as an asymptotic bias term in our approximation. This means that the confidence intervals constructed without correcting for this non-vanishing bias term will be statistically invalid.

While there is no clear universally accepted method for accounting for this bias, many researchers use **undersmoothing** or **bias-correction** to construct asymptotically valid confidence intervals.

Undersmoothing involves (in the context of kernel-based estimators) using a smaller than optimal bandwidth, which is known to asymptotically reduce the bias and increase the variance of the estimator. This will allow for the second term in (2) to approach zero in the limit.

Bias-correction, on the other hand, tries to directly address the lingering asymptotic bias term by approximating it. This approach comes with its own drawbacks, for example, this method usually requires estimating derivatives of the regression function which can be quite involved and require stronger assumptions.

6 Local likelihood

Recall that our regression model assumes that Y is a continuous, real-valued random variable and the noise, ε is mean zero, finite variance that is independent of the regressor x. What kind of non-parametric estimator would be most suitable if instead $Y \in \{0, 1\}$ or Y is a categorical variable?

Lets look briefly at a more general setting. In particular, suppose Y comes from the exponential dispersion family:

$$f(y|x) = \exp\left(\frac{yT(x) - b(T(x))}{a(\phi)} + c(y, \phi)\right).$$

Note that this family for any *fixed* ϕ, is an exponential family. Then,

$$r(x) = \mathbb{E}[Y|x] = b'(T(x))$$
$$\sigma^2(x) = \mathbb{V}(Y|x) = a(\phi)b''(T(x))$$

Recall that we usually assume the parametric model with some link function g, $r(x) = g^{-1}(\beta^{\mathrm{T}}x)$. Then, based on the parametric assumption on the conditional density of $y|x$, $f(y|x)$, and an appropriate link function, the MLE for β can be computed directly.

Lets look at a concrete example. Suppose $Y|X = x \sim \mathrm{Binom}(n, r(x))$. That is,

$$f(y|x) = \binom{n}{y} r(x)^y (1 - r(x))^{n-y}.$$

Then, this conditional distribution is an exponential dispersion family with

$$T(x) = \log\frac{r(x)}{1 - r(x)}, \qquad b(T) = n\log(1 + e^T),$$

and $a(\phi) = 1$. If we take the link function to be $g(t) = \log(t/(m - t))$, we have the logistic regression model.

Lets now look at a nonparametric version of this. Consider the local linear estimation problem wherein $Y \sim \mathrm{Bernoulli}(r(x))$ for some smooth function $0 \leq r(x) \leq 1$. The likelihood is given by

$$\Pi_{i=1}^n r(x_i)^{Y_i}(1 - r(x_i))^{1-Y_i}.$$

Then, with the logistic link function applied to $r(x_i)$, the log-likelihood is

$$l(r) = \sum_i l(Y_i, g(r(x_i)))$$

where

$$l(y,g) = \log\left(\left(\frac{e^g}{1+e^g}\right)^y \left(\frac{1}{1+e^g}\right)^{1-y}\right) = yg - \log(1+e^g).$$

In order to estimate the regression function at some value x, we approximate with the **local** logistic function

$$r(u) \approx \frac{e^{a_0+a_1(u-x)}}{1+e^{a_0+a_1(u-x)}}$$

for values of u near x. Then, the local log-likelihood is defined as

$$\begin{aligned} l_x(a) &= \sum_i K_h(X_i,x) l(Y_i, a_0 + a_1(X_i - x)) \\ &= \sum_i K_h(X_i,x)(Y_i(a_0 + a_1(X_i - x)) - \log(1 + e^{a_0+a_1(X_i-x)})). \end{aligned}$$

Suppose $\hat{a}_0, \hat{a}_1$ maximize l_x, the non-parametric estimate of the regression function is

$$\hat{r}_n(x) = \frac{e^{\hat{a}_0(x)+\hat{a}_1(x)}}{1+e^{\hat{a}_0(x)+\hat{a}_1(x)}}.$$

Of course, as before, cross-validation versions of $\hat{a}_0$ and $\hat{a}_1$ can be estimated for better control on the bias.

References

Silverman, B. W. (1984). Spline smoothing: the equivalent variable kernel method. *The annals of Statistics*, pages 898–916.

Stone, C. J. (1977). Consistent nonparametric regression. *The annals of statistics*, pages 595–620.