Model Selection in Nonparametric Regression

Paweł Polak

February 4, 2016

STAT W4413: Nonparametric Statistics - Lecture 7

Selecting bandwidth h and degree of polynomial M

So far:

- we have studied the theoretical properties of locally polynomial/kernel regression and showed under what conditions these algorithms can perform well.
- we showed that the performance of these algorithms is highly dependant on the bandwidth h and also on the polynomial degree M.
- while theoretical results may provide some guidelines, they still do not provide a good practical way to set the parameters.
- this is specially true since in most cases, we don't know the characteristics of g such as an upper bound on its derivatives.
- now we give a few ways to find the optimal parameters of a model or even the optimal model itself.

We discuss several different methods in this lecture. But the only part that is mandatory for this course is *cross validation*.

Cross validation

Probably the simplest to understand and the most popular approach in tuning the parameters such as the bandwidth h and the polynomial degree M is *cross validation*.

Let $g : [0,1] \to \mathbb{R}$ be the function that we would like to estimate. For every value of h we have the estimate $\hat{g}_h(x)$ for every $x \in [0,1]$. For instance in Kernel regression $\hat{g}_h(x)$ is given by

$$\hat{g}_h(x) = \frac{\sum_{i=1}^n K(\frac{|x_i - x|}{h}) y_i}{\sum_{i=1}^n K(\frac{|x_i - x|}{h})}$$

What do we mean when we say the "best value of h"?

- This is the value of h that gives us an estimate $\hat{g}_h$ closest to g.
- We calculated the optimal value of h for kernel regression.
- But, that required a good bound on the first derivative of the function g that is not available in practice.
- In order to define the optimal value of h we should first define a measure under which we would like to measure the closeness of g and $\hat{g}_h$.

Cross validation

- We considered the MSE at point x as the "closeness" criteria.
- However, since usually we are interested in the closeness of our estimate over the entire interval $[0, 1]$, the following criterion is more popular:

$$R(g, \hat{g}_h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(g(x_i) - \hat{g}_h(x_i))^2, \tag{1}$$

  where the expected value is with respect to $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$.

- We call this the average MSE of $\hat{g}_h$.
- We call it average risk since it is the average of the risk at the $x_1, x_2, \ldots, x_n$.
- The optimal value of h is the value that minimizes this average risk. Given a value of h, can we calculate the average MSE?
- NO. The main issue is that g is not known. Otherwise, we would not even try to estimate it!

Cross validation

- To resolve this issue we first note that

$$y_i = g(x_i) + \epsilon_i.$$

- So, if the noise ϵ_i is not too large, then one idea would be to replace $g(x_i)$ with y_i.
- If we do so, we obtain

$$R_T(g, \hat{g}_h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(y_i - \hat{g}_h(x_i))^2.$$

  R_T is called the *training error*.

- Is the training error a useful criterion for estimating the parameter h?

Cross validation

- To address this question let's look at the extreme case where we use box kernel regression with very small value of h.
- If h is very small then $\hat{g}_h(x_i) = y_i$. Explain why?
- Hence the training error will be zero. In other words, according to R_T the optimal value of h is $h = 0$!!
- But, from our bias, variance calculations we know that small values of h suffer from the high variance and will lead to over-fitting.
- Therefore, $R_T(g, \hat{g}_h)$ is not a good approximation for $R(g, \hat{g}_h)$.
- The reason this is happening is that by minimizing the training error we let the algorithm choose the parameters such that the fitted curve passes through the data points. This always happens if we use the training error for setting the parameters.

Is there any better way to estimate $R(g, \hat{g}_h)$?

Cross Validation

Suppose that when we are approximating $(g(x_i) - \hat{g}_h(x_i))^2$ with $(y_i - \hat{g}_h(x_i))^2$, we ensure that our estimator is not using the (x_i, y_i) sample. This ensures that our estimator cannot fit to the noise at location i. I show this new estimator with $\hat{g}_h^{\setminus i}$, meaning that $\hat{g}_h^{\setminus i}$ is constructed exactly the same as before except that it does not use the (x_i, y_i) sample to obtain its estimate at x_i. Now we approximate $R(g, \hat{g}_h)$ with

$$RCV_1(g, \hat{g}_h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(y_i - \hat{g}_h^{\setminus i}(x_i))^2.$$

This quantity does not have the problem of the training error. In fact, we have

A number cov(epsilon,else)=0

$$\mathbb{E}(y_i - \hat{g}_h^{\setminus i}(x_i))^2 = \mathbb{E}(g(x_i) + \epsilon_i - \hat{g}_h^{\setminus i}(x_i))^2 = \sigma^2 + \mathbb{E}(g(x_i) - \hat{g}_h^{\setminus i}(x_i))^2.$$

If the number of samples n is large and h is large enough, then we can show that $\hat{g}_h^{\setminus i}(x_i) \approx \hat{g}_h(x_i)$.[1] Can you explain intuitively why? Therefore

$$\mathbb{E}(y_i - \hat{g}_h^{\setminus i}(x_i))^2 \approx \sigma^2 + \mathbb{E}(g(x_i) - \hat{g}_h(x_i))^2.$$

This says that minimizing the $RCV_1(g, \hat{g}_h)$ provides a very similar result to the average mean square error in (1).

[1] Note that for small values of h this approximation is not correct. But, the estimated risk for these small values of h is larger than the actual risk, since we are using fewer samples in our estimate.

Leave-One-Out Cross Validation

This is the value we need to calculate

The quantity $RCV_1(g, \hat{g}_h)$ still cannot be calculated from the data. In fact, the expected value, which is with respect to noise, cannot be calculated in many cases since the statistics of the noise are not known either. However, it is easy to obtain an unbiased estimate of $RCV_1(g, \hat{g}_h)$ in the following way:

$$\widehat{RCV}_1(g, \hat{g}_h) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{g}_h^{\setminus i}(x_i))^2$$

Under certain assumptions on the estimator and the noise, one can prove that for large values of n, $\widehat{RCV}_1(g, \hat{g}_h)$ is close to $RCV_1(g, \hat{g}_h)$. However, this is beyond the scope of this course and we will not prove it here.

Now in order to find the optimal value of h, we find the value of h that minimizes $\widehat{RCV}_1$.

Consider M different values of h, i.e., $\{h_1, h_2, \ldots, h_M\}$ and suppose that we would like to obtain the best value of h. Here is what we can do:

!!! m not order of poly

h=seq(*******)

1. Calculate $\widehat{RCV}_1(g, \hat{g}_{h_1}), \ldots, \widehat{RCV}_1(g, \hat{g}_{h_M})$.
2. Find which $\widehat{RCV}_1(g, \hat{g}_{h_i})$ is the smallest and pick that as the optimal value of h.

The procedure that is introduced above is known as *leave-one out cross validation*. It must be clear why it is called "leave-one out". Why?

B-fold Cross Validation

The idea of leave-one-out cross validation can (obviously) be extended:

- Suppose that instead of taking one sample out from our estimate we partition our data into B blocks of size n/B. Call the blocks $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_B$.
- At every iteration we remove one block from our data and use the remaining $B-1$ blocks to estimate g.
- We call our estimate $\hat{g}^{\setminus i}_{B,h}$ meaning that the i^{th} block does not contribute to our estimate.
- Then we estimate the risk of our estimate on the j^{th} block from

$$\widehat{RCV_j^B} = \frac{B}{n} \sum_{\{i:(x_i,y_i)\in\mathcal{B}_j\}} (y_i - \hat{g}^{\setminus i}_{B,h}(x_i))^2.$$

- Then we calculate the final cross validation risk as

$$\widehat{RCV^B} = \frac{1}{B}\sum_{j=1}^{B} \widehat{RCV_j^B}.$$

underestimates the variance

This is called B-fold cross validation. This offers a new estimate of $R(g, \hat{g}_h)$. Can you explain how you would use this risk to obtain the optimal value of h?

Cross Validation: Remarks

- Note that cross validation is a method to estimate the actual risk $R(g, \hat{g}_h)$.
- Therefore, the estimates that we have calculated here by leave-one-out or B-fold cross validation are not exactly the same as $R(g, \hat{g}_h)$ and suffer from bias and variance.
- As we described the leave-one-out cross validation is a low bias estimate. But it may have a large variance.
- On the other hand as we decrease B the variance decreases and the bias increases.
- However, the impact of the parameter B is not very huge in practice and we can essentially use any value of B.
- $B = 5$ or $B = 10$ are the choices that are used very often in practice.
- However if the sample size is very small, it is better to consider the leave one out cross validation.

Another issue that we should consider in choosing the value of B is the computational complexity. In leave-one-out cross validation we are essentially solving the same problem n times. This is for many algorithms (such as spline) computationally very expensive and when we reduce B we are decreasing the computational complexity as well. That is one of the points that you may want to keep in mind regarding the leave-one out cross validation.

Generalized cross validation for linear fitting methods (optional)

Generalized cross validation is another model selection technique that is computationally less expensive than CV, but has a much more limited scope. Most of the estimators we have considered in this class fall in the category of linear estimators. An estimator is called linear, if its output is a linear function of $(Y_1, Y_2, \ldots, Y_n)$. It is therefore clear that for any linear estimator we have

$$[\hat{g}(x_1), \hat{g}(x_2), \hat{g}(x_3), \ldots, \hat{g}(x_n)]^T = S[y_1, y_2, \ldots, y_n]^T.$$

HAT MATRIX
X(X'X)-1X'

S is an $n \times n$ matrix that is usually called smoothing matrix. For certain linear estimators (such as kernel regression, locally polynomial regression, and spline) $\widehat{RCV}_1$ can be calculated from:

$$\widehat{RCV}_1 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{g}(x_i)}{1 - S_{ii}} \right)^2.$$

The generalized cross validation replaces this with

$$GCV = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{g}(x_i)}{1 - trace(S)/n} \right)^2.$$

In other words if all the diagonal elements of S are equal to each other then GCV and leave-one-out cross validation provide the same result.

Other model selection techniques (optional)

- There are a few more model selection techniques including AIC, BIC, and Mallow's C_p.
- Most of these techniques are originally designed for parametric schemes.
- However, they can be modified to cover the non-parameteric curve fitting problem as well.

Here we first review how these techniques have been derived for the parametric settings and will then describe how they can be modified to nonparametric curve fitting problem.

Mallow's C_p

Consider $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where $y_i = g(x_i) + \epsilon_i$. Suppose that with a certain method we obtain estimates $\hat{g}(x_i)$. Define the prediction error as

$$PE = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(Y_i^{new} - \hat{g}(x_i))^2,$$

where $Y_i^{new} = g(x_i) + \epsilon_i^{new}$ is independent of $\epsilon_1, \ldots, \epsilon_n$. Note that

$$PE = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(g(x_i) + \epsilon_i^{new} - \hat{g}(x_i))^2 = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[(g(x_i) - \hat{g}(x_i))^2 + (\epsilon_i^{new})^2] = R(g, \hat{g}) + \sigma^2,$$

where we defined $R(g, \hat{g})$ in (1). Therefore, if we could estimate PE then it could help us in tuning the parameters. But, we cannot do it, since it requires a new sample Y_i^{new} that is not available in practice.

Our goal in the next few slides is to show that in fact we can calculate this quantity. Remember one of the quantities we could easily estimate was (the training error)

$$\hat{R}_T = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{g}(x_i))^2.$$

Note that this is an unbiased estimate of RT. It turns out that these two quantities have a simple connection with each other.

Mallow's C_p

Lemma (Mallow 1973)

The prediction error and the training error satisfy

$$PE = \mathbb{E}(R_T) + \frac{2}{n}\sum_{i=1}^{n} Cov(\hat{g}(x_i), y_i).$$

Proof.

Let's start with the calculation of $\mathbb{E}(Y_i^{new} - \hat{g}(x_i))^2$:

$$\begin{aligned}
\mathbb{E}(Y_i^{new} - \hat{g}(x_i))^2 &= \mathbb{E}(Y_i^{new} - y_i + y_i - \hat{g}(x_i))^2 = \mathbb{E}(Y_i^{new} - y_i)^2 + \mathbb{E}(y_i - \hat{g}(x_i))^2 \\
&+ 2\mathbb{E}((Y_i^{new} - y_i)(y_i - \hat{g}(x_i))) \\
&= 2\sigma^2 + \mathbb{E}(y_i - \hat{g}(x_i))^2 + 2\mathbb{E}((Y_i^{new} - y_i)(y_i - \mathbb{E}(\hat{g}(x_i)))) \\
&+ 2\mathbb{E}((Y_i^{new} - y_i)(\mathbb{E}(\hat{g}(x_i)) - \hat{g}(x_i))) \\
&= 2\sigma^2 + \mathbb{E}(y_i - \hat{g}(x_i))^2 - 2\sigma^2 + 2\mathbb{E}((g(x_i) - y_i)(\mathbb{E}(\hat{g}(x_i)) - \hat{g}(x_i))) \\
&= \mathbb{E}(y_i - \hat{g}(x_i))^2 + 2Cov(\hat{g}(x_i), y_i).
\end{aligned}$$

p/n*sigma^2

Summation over i completes the proof. □

Mallow's C_p

- Note that even if we replace the expected values with empirical averages (that is a reasonable thing to do since n is usually not very small), neither the left nor the right hand side of this equation cannot be calculated.
- The left hand side is clear, since it requires a new sample and we don't have access to g to generate that sample.
- For the calculation of the right hand side we also need to calculate $\mathbb{E}(y_i)$ which is hidden in the covariance and that is equal to $g(x_i)$.
- So, at this level it seems that Lemma 1 is not useful.
- However, we would like to show that under certain conditions we can actually calculate $\frac{1}{n}\sum_{i=1}^{n} Cov(\hat{g}(x_i), y_i)$.

Mallow's C_p

Let's start with the simple situation that covers most of the instances we have covered in this course. Suppose that our estimated can be written as

$$\begin{bmatrix} \hat{g}(x_1) \\ \hat{g}(x_2) \\ . \\ . \\ . \\ \hat{g}(x_n) \end{bmatrix} = S \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix}, \tag{2}$$

where S does not depend on $y_1, y_2, \ldots, y_n$. The Kernel method, locally linear regression, spline and many other methods fall into this category. Our next step is to show that under this model we have

$$\frac{1}{n}\sum_{i=1}^{n} Cov(\hat{g}(x_i), y_i) = \frac{1}{n} Tr(S)\sigma^2. \quad \texttt{p/n*sigma^2}$$

Mallow's C_p

Lemma

Under the model specified in 20 we have

$$\frac{1}{n}\sum_{i=1}^{n} Cov(\hat{g}(x_i), y_i) = \frac{1}{n} Tr(S)\sigma^2.$$

Proof.

We have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} Cov(\hat{g}(x_i), y_i) &= \frac{1}{n}\mathbb{E}\left([y_1 - g(x_1), y_2 - g(x_2), \ldots, y_n - g(x_n)] S \begin{bmatrix} y_1 - g(x_1) \\ y_2 - g(x_2) \\ \vdots \\ y_n - g(x_n) \end{bmatrix} \right) \\
&= \frac{1}{n}\mathbb{E}\left([\epsilon_1, \epsilon_2, \ldots, \epsilon_n] S \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \right) = \frac{1}{n} Tr(S)\sigma^2 \qquad (3)
\end{aligned}$$

□

Note that S can always be calculated from the data. Based on these calculations Mallows proposed the following criterion for model selection problem:

$$C_p = \hat{R}_T + 2Tr(S)\hat{\sigma}^2/n,$$

where $\hat{\sigma}^2$ is the estimate of the noise variance σ^2. There are several methods in estimating σ^2. We will discuss them later in the course.

Stein Unbiased Risk Estimation (SURE) (optional)

Can we extend this approach to the cases where the estimator is not linear? (optional because all the estimators in the course are linear estimators, but this is very useful in many applications)

Charles Stein has shown that if we make more assumption on the noise and assume that the noise is iid Gaussian we can still have an unbiased estimate of the $Cov(\hat{g}(x_i), y_i)$ without requiring $g(x)$. To understand this claim let us start with the following simple lemma.

Lemma (Stein's lemma)

Let $Z_1, Z_2, \ldots, Z_n \sim N(\mu, \sigma^2)$ and h: $\mathbb{R}^n \to \mathbb{R}$ a measurable function. Then

$$\mathbb{E}(Z_1 - \mu)h(Z_1, Z_2, \ldots, Z_n) = \sigma^2 \mathbb{E}\left(\frac{\partial h(Z_1, Z_2, \ldots, Z_n)}{\partial Z_1}\right)$$

The proof is a simple application of integration by parts and is left as an exercise. But even if you cannot prove it, you may google it! Based on this lemma we can prove the following useful proposition:

Proposition

Let $y_i = g(x_i) + \epsilon_i$ with $\epsilon_1, \epsilon_2, \ldots, \epsilon_n \overset{iid}{\sim} N(0, \sigma^2)$. Then

$$Cov(y_i, \hat{g}(x_i)) = \sigma^2 \mathbb{E}\left(\frac{\partial \hat{g}(x_i)}{\partial y_i}\right).$$

Stein Unbiased Risk Estimation (SURE) (optional)

Proof.

The proof is a simple application of Stein's lemma. We have

$$Cov(y_i, \hat{g}(x_i)) = \mathbb{E}(\epsilon_i(\hat{g}(x_i) - \mathbb{E}\hat{g}(x_i))) = \mathbb{E}(\epsilon_i \hat{g}(x_i)) = \sigma^2 \mathbb{E}\left(\frac{\partial \hat{g}(x_i)}{\partial \epsilon_i}\right) = \sigma^2 \mathbb{E}\left(\frac{\partial \hat{g}(x_i)}{\partial y_i}\right).$$

□

Based on these calculation we can consider minimizing the following criterion in the model selection step:

$$\text{SURE} = \hat{R}_T + \frac{2\hat{\sigma}^2}{n} \sum_{i=1}^{n} \frac{\partial \hat{g}(x_i)}{\partial y_i}.$$

Note that *SURE* is a good estimate of *PE* that we introduced in the last section. This is known as Stein Unbiased Risk Estimate (SURE). Note that in the special case that $\hat{g}$ is a linear transformation of $y_1, y_2, \ldots, y_n$ this reduces to Mallow's C_p. For more information refer to [4, 5]

AIC

General model- We first start with the parametric model. Let $X_1, X_2, \ldots, X_n \overset{iid}{\sim} g$. We would like to model the data with a parametric model $p_\theta(x)$, where $\theta \in \Theta \subset \mathbb{R}^k$. Also, assume that the interior of Θ is nonempty. Then AIC for this model is defined as

$$AIC = \frac{2k}{n} - \frac{2}{n}\sum_{i=1}^{n} \log p_{\hat{\theta}}(x_i),$$

where $\hat{\theta}$ is the maximum likelihood estimate of θ. To compare two different models we can calculate AIC for each of them. The one that gives a higher AIC can be considered as a better model.

The first question we would like to answer is about the origin of this quantity. One simple idea to do the model selection is to compare the actual distribution g with a distribution from the model that is closest to g. Suppose I define a distance d between two distributions that we denote with $d(g, p_\theta)$. Then the value of $\min_\theta d(g, p_\theta)$ can be considered as a criterion to understand which model is the best. One distance between two distributions is called the KL-divergence and is defined as

Kullback-Leibler divergence

$$d(g, p_\theta) = \int g(x) \log \frac{g(x)}{p_\theta(x)} dx.$$

NOT a distance!!!!!!

AIC

Since g is fixed and is the same when we are comparing different models, we can even simplify the expression to $\int g(x) \log p_\theta(x) dx$ and consider the following criteria for model selection.

$$L = \max_\theta \int g(x) \log p_\theta(x) dx.$$

There are two major challenges in here. (1) $g(x)$ is not known. (2) Optimal value of the parameter, θ^*, can not be calculated. To address the first issue we can estimate $\int g(x) \log(p_{\theta*}(x)) dx$ with $\frac{1}{n} \sum_{i=1}^n \log(p_{\theta*}(X_i))$ and we know that as $n \to \infty$ it will converge to the corresponding integral. This in fact, is a good estimate of L and as n goes to infinity it will converge to L. Why? Therefore we define

$$L_n = \frac{1}{n} \sum_{i=1}^n \log(p_{\theta*}(X_i)).$$

Note that L_n still cannot be calculated. The reason is that θ^* is not known. Therefore, we can estimate it by replacing θ^* with the maximum likelihood estimate of θ denoted by θ^*_{ML}.

AIC

Therefore, our estimate of L_n is given by

$$\hat{L}_n = \frac{1}{n} \sum_{i=1}^{n} \log(p_{\hat{\theta}_{ML}}(X_i)).$$

But how good is this estimate? We can prove that ([**?**] page 515)

$$n(\hat{L}_n - L_n) \xrightarrow{d} \chi^2(k),$$

where k is the dimension of the space of parameters. Therefore, the estimate seems to be biased. Therefore, it seems more appropriate to use the estimate $\hat{L} - k/n$ instead of $\hat{L}$. In fact when both k and n are large $\hat{L} - \frac{k}{n}$ is a very good estimate for L. That's why AIC is defined the way it is defined.

AIC for nonparametric setting under Gaussian noise model

Consider the nonparametric setting where we see $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. y_is are modeled as $y_i = g^*(x_i) + \epsilon_i$. Here we assume that $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. We further assume that the locations of x_is are fixed and we would like to estimate the function at those values only. If we form the log likelihood for this problem we obtain

$$L = \min_{g \in \mathcal{C}_g} \frac{1}{n} \sum_{i=1}^{n} \int \underbrace{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i - g^*(x_i))^2}{2\sigma^2}}}_{g} \underbrace{\frac{(y_i - g(x_i))^2}{2\sigma^2}}_{\log(p(\Theta))} dy_i = \min_{g \in \mathcal{C}_g} \frac{1}{2n\sigma^2} \sum_{i=1}^{n} \mathbb{E}(y_i - g(x_i))^2.$$

Note that $\frac{1}{\sqrt{2\pi}\sigma} e^{-(y_i - g^*(x_i))^2/2}$ corresponds to $g(x)$ in the last section, and $\frac{(y_i - g(x_i))^2}{2\sigma^2}$ corresponds to $\log(p_\theta)$ in the last section. Further, C_g specifies the class of functions we are doing the optimization over, that does not necessarily include g^* (for instance for some restricted assumptions). Suppose $g_1 \in \mathcal{C}_g$ achieves L. Now we would like to calculate the distance between L and the expected value of the training error.

AIC for nonparametric setting under Gaussian noise model

Lemma

Under the assumptions of this section we have

$$2\sigma^2 L - \mathbb{E}\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{g}(x_i))^2 = \frac{1}{n}(g^*(x_i) - g_1(x_i))^2 - (g^*(x_i) - \hat{g}(x_i))^2 + \frac{2}{n}\sum_{i=1}^{n} cov(Y_i, \hat{g}(x_i)).$$

The proof is simple and left as a homework. Usually our $\hat{g}$ is calculated by minimizing $\min_{g \in \mathcal{C}_g} \frac{1}{n}\sum_{i=1}^{n}(y_i - g(x_i))^2$. In such cases, under certain regularity conditions we can prove that as $n \to \infty$, $\frac{1}{n}(g^*(x_i) - g_1(x_i))^2 - (g^*(x_i) - \hat{g}(x_i))^2 \to 0$. Hence, AIC will be equivalent to SURE.

BIC

Consider the model we described for AIC. The Bayesian information criterion (BIC) is defined as

$$BIC = \frac{k \log n}{n} - \frac{2}{n} \sum_{i=1}^{n} \log p_{\hat{\theta}}(x_i),$$

where $\hat{\theta}$ is the maximum likelihood estimator of θ.

It is clear that BIC is very similar to AIC, except that the factor 2 in AIC is replaced with $\log n$. Therefore, BIC usually penalizes the more complex models (models with more parameters) more heavily and therefore tends to select simpler models. Note that for smaller sample sizes BIC may lead to better result, while for large sample sizes AIC seems to be a better choice. We will understand the reason once we describe where BIC is derived from.
The way BIC is derived is very much different from the way AIC is derived. For a nice and short explanation of this derivation refer to page 233-234 of [**?**].
As before, if we are in nonparametric setting and the estimator is linear, then we can still replace k with the degrees of freedom of the model.
Comparison with CV and Bootstrap- Bradley Efron has a nice paper on the comparison of C_p and AIC with cross validation and Bootstrap [**?**]. I suggest you read this paper. Clearly, this is an optional part of the course.

Local regression in $\mathbb{R}^p$

So far we have considered a simple problem where the observations y_i can be represented as a smooth function of only one other random variable, namely x_i. We call x a feature. In many practical situations we have more than one feature. For instance, suppose that we would like to estimate the price of an apartment in Manhattan. The features that may have impact on the price may include area, neighborhood, floor level, and age of the building. How do we do regression in such settings?

Local regression in $\mathbb{R}^p$

Before we answer this question, let us formally state our notation. We observe $\{X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}\}_{i=1}^n$, and we believe that

$$Y_i = g(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}) + \epsilon_i.$$

As before, ϵ_i denotes the noise in the system and is considered to be i.i.d. with mean zero and variance equal to σ^2. The objective now is to "learn" or estimate g at a given point $(x^{(1)}, x^{(2)}, \ldots, x^{(p)})$. It turns out that both techniques we described in the last few sections can be easily extended to this situation. The first step is to define the local neighborhood of $(x^{(1)}, x^{(2)}, \ldots, x^{(p)})$. The most straightforward and most popular definition of the local neighborhood is the set of points whose Euclidean distance from $(x^{(1)}, x^{(2)}, \ldots, x^{(p)})$, i.e.,

$$||(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}) - (x^{(1)}, x^{(2)}, \ldots, x^{(p)})|| = \sqrt{\sum_{j=1}^{p} (X_i^{(j)} - x^{(j)})^2}$$

is small. Now that the local neighborhood is defined the extension of kernel regression is straightforward. For instance, if we would like to do Kernel regression with kernel K, then we have

$$\hat{g}(x^{(1)}, x^{(2)}, \ldots, x^{(p)}) = arg \min_{\beta_0} \sum_{i=1}^{n} K\left(\frac{||(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}) - (x^{(1)}, x^{(2)}, \ldots, x^{(p)})||}{h}\right)(y_i - \beta_0)^2.$$

Local regression in $\mathbb{R}^p$

This clearly leads to the formula

$$\hat{g}(x^{(1)}, x^{(2)}, \ldots, x^{(p)}) = \frac{\sum_i K\left(\frac{||(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}) - (x^{(1)}, x^{(2)}, \ldots, x^{(p)})||}{h}\right) y_i}{\sum_i K\left(\frac{||(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}) - (x^{(1)}, x^{(2)}, \ldots, x^{(p)})||}{h}\right)}$$

for the kernel regression technique. Locally polynomial regression can also be extended to this problem. For the simplicity of notation we only mention the locally linear regression. To do locally linear regression we use

$$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p) = arg \min_{\beta_0, \beta_1, \ldots, \beta_p} \sum_{i=1}^{n} K\left(\frac{||(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(p)}) - (x^{(1)}, x^{(2)}, \ldots, x^{(p)})||}{h}\right) (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_i^{(j)})^2,$$

and then estimate g according to

$$\hat{g}(x^{(1)}, x^{(2)}, \ldots, x^{(p)}) = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x^{(j)}.$$

Can you explain how we can have locally polynomial regression in higher dimensions?