K-Nearest-Neighbour Methods

A very simple method that illustrates several key concepts is the method of ***K*-nearest neighbours**.

KNN (or K-NN) can be used for both regression and classification. A less formal version is often used by real-estate agents.

For regression, suppose we want to predict $f(\boldsymbol{x})$ for a specified vector of inputs $\boldsymbol{x}_0$.

- Find the K observations closest to $\boldsymbol{x}_0$. This could be done using more than one metric. We may need to rescale the predictors so that the distance is not dominated by one or a few of them.
- Let $\mathcal{N}_0$ denote the set of the K closest observations.
- Compute the average of the y_i over all the observations that belong to $\mathcal{N}_0$.
- The prediction $\hat{f}(\boldsymbol{x}_0)$ is simply $(1/K)\sum_{i\in\mathcal{N}_0} y_i$.

Rescaling the predictors can be extremely important. It is part of many machine-learning procedures.

- Suppose that x_1 varies from 1212 to 9873, and x_2 varies from -0.234 to 0.456. The scaling here is vastly different!
- If we use the squared Euclidean distance $(\boldsymbol{x}_i - \boldsymbol{x}_0)^\top(\boldsymbol{x}_i - \boldsymbol{x}_0)$ and do not rescale, $\mathcal{N}_0$ will consist solely of observations that are close to $\boldsymbol{x}_0$ in terms of x_1, but they may be very far away in terms of x_2.

Note that we choose the value(s) of $\boldsymbol{x}_0$. They may or may not correspond to some or all of the sample observations.

KNN assigns the same weight (1) to the closest K observations and zero weights to all other observations.

This seems too extreme. Other methods give positive weight to many of the observations, perhaps even to all of them, but the weights diminish as $\boldsymbol{x}_i$ becomes further from $\boldsymbol{x}_0$.

The Bias-Variance Tradeoff for KNN

If we set $K = 1$, then $\mathcal{N}_0$ simply contains the observation closest to $\boldsymbol{x}_0$. This is called 1NN, or **single nearest neighbour**.

For 1NN, there will not be much bias, unless $f(\boldsymbol{x}_i) - f(\boldsymbol{x}_0)$ is large even for the very closest observation.

That may occur if the $\boldsymbol{x}_i$ are very sparse, so that none of them is close to $\boldsymbol{x}_0$, and/or if $f(\cdot)$ is very steep in some directions near $\boldsymbol{x}_0$.

If $\boldsymbol{x}_0$ belongs to the sample, then the closest observation will be itself, and there will be no bias at all.

When $K = 1$, there will almost certainly be a large variance. When $\boldsymbol{x}_0 = \boldsymbol{x}_i$, then $\hat{f}(\boldsymbol{x}_i)$ is just y_i:

$$\hat{f}(\boldsymbol{x}_i) = y_i = f(\boldsymbol{x}_i) + u_i. \tag{1}$$

Therefore, $y_i - \hat{f}(\boldsymbol{x}_i) = 0$.

(1) implies that the expected squared reducible error is

$$\begin{aligned} \mathrm{E}\big(f(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i)\big)^2 &= \mathrm{E}\big(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_i) - u_i\big)^2 && (2)\\ &= \mathrm{Var}(u_i). && (3) \end{aligned}$$

Unless $\mathrm{Var}(u_i)$ is small, this is likely to be much bigger than we would like to see.

- In the statistical learning literature, it is usually assumed that $\mathrm{Var}(u_i)$ is the same for all i.
- This is not what econometricians typically assume nowadays.

In general, for KNN,

$$\hat{f}(\boldsymbol{x}_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} y_i = \frac{1}{K} \sum_{i \in \mathcal{N}_0} f(\boldsymbol{x}_i) + \frac{1}{K} \sum_{i \in \mathcal{N}_0} u_i. \tag{4}$$

The first term on the r.h.s. of (4) potentially causes bias.

If there are many observations near $\boldsymbol{x}_0$, so that $\boldsymbol{x}_i \approx \boldsymbol{x}_0$ for all $\boldsymbol{x}_i \in \mathcal{N}_0$, then this term is approximately $f(\boldsymbol{x}_0)$, and there is (almost) no bias.

In that case, from (4), the expected squared reducible error is approximately

$$\mathrm{E}\big(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0)\big)^2 = \frac{1}{K}\,\mathrm{Var}(u_i). \tag{5}$$

This is K times smaller than (3). Note that (5) depends on the crucial assumption that $\mathrm{Cov}(u_i, u_i') = 0$, which is needed for the result that $\mathrm{Var}(\sum_{i\in\mathcal{N}_0} u_i) = \sum_{i\in\mathcal{N}_0} \mathrm{Var}(u_i) = K\,\mathrm{Var}(u_i)$.

Of course, in practice, $f(\boldsymbol{x}_i) \neq f(\boldsymbol{x}_0)$ for all $i \in \mathcal{N}_0$. So the reducible error includes a bias term, which increases as K increases.

- When $K = 1$, the bias should usually be small. In fact, as we have seen, it is zero if $\boldsymbol{x}_0$ is a point in the sample.
- When $K > 1$, the bias may or may not be small. It depends on how close the K nearest values of $\boldsymbol{x}_i$ are to $\boldsymbol{x}_0$ and on how much $f(\boldsymbol{x}_i)$ differs from $f(\boldsymbol{x}_0)$ for those values.

In general, from (4),

$$f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0) = f(\boldsymbol{x}_0) - \frac{1}{K}\sum_{i\in\mathcal{N}_0} f(\boldsymbol{x}_i) - \frac{1}{K}\sum_{i\in\mathcal{N}_0} u_i, \tag{6}$$

so that the expected squared reducible error is

$$\mathrm{E}\big(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0)\big)^2 = \mathrm{E}\left(f(\boldsymbol{x}_0) - \frac{1}{K}\sum_{i\in\mathcal{N}_0} f(\boldsymbol{x}_i)\right)^2 + \frac{1}{K}\,\mathrm{Var}(u_i). \tag{7}$$

The reducible error in (7) consists of two parts.

- The first part is the squared bias, and the second part is the variance.
- Increasing K makes the squared bias larger and the variance smaller. Thus K is a classic tuning parameter.
- The optimal K increases as n increases, but much more slowly. In some cases, theory suggests that $K = O(n^{1/5})$.

KNN can be used for classification as well as regression.

- For binary classification, the y_i are all 0 or 1.
- Pick K to be odd, so that $(1/K)\sum_{i\in\mathcal{N}_0} y_i$ cannot equal $1/2$.
- Then classify the (unobserved) y_0 as 1 when $(1/k)\sum_{i\in\mathcal{N}_0} y_i > 1/2$ and as 0 when $(1/k)\sum_{i\in\mathcal{N}_0} y_i < 1/2$.
- The estimated probability that $y_0 = 1$ is simply $(1/k)\sum_{i\in\mathcal{N}_0} y_i$.

There is little difference between regression and binary classification, since $(1/k)\sum_{i\in\mathcal{N}_0} y_i$ is both the regression fitted value and the estimated probability.

We can generalize KNN to classification involving 3 or more responses. To avoid ties, choose K so that the number of classes does not divide evenly by it.

Although binary classification by KNN is very similar to KNN regression, choosing among three *unordered* responses is quite different. See Chapter 4.

A Digression on Big-O Notation

The **same-order notation** is a way of saying that something increases (or decreases) at the same rate as something else.

For example, if $g(n)$ is roughly proportional to n^2 (exactly proportional for large enough n), then we may write $g(n) = O(n^2)$.

If $g(n) = O(n^r)$ and $h(n) = O(n^q)$, then $g(n)h(n) = O(n^{r+q})$.

In the same case, $g(n) + h(n) = O(n^{\max(r,q)})$.

For asymptotic theory, we often have results like

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{u} = O(n^{-1})O(n^{1/2}) = O(n^{-1/2}). \tag{8}$$

This means that the OLS estimator $\hat{\boldsymbol{\beta}}$ tends to the true parameter vector $\boldsymbol{\beta}_0$ at a rate proportional to $1/\sqrt{n}$.

The result (8) depends on the assumptions that $\boldsymbol{X}^\top \boldsymbol{X} = O(n)$ and $\boldsymbol{X}^\top \boldsymbol{u} = O(n^{1/2})$.

For asymptotic theory, estimators generally converge to the true parameter values at some rate as $n \to \infty$, so we see negative powers of n, like $O(n^{-1})$ or $O(n^{-1/2})$.

We may also be concerned with computational cost as n becomes large, so we see positive powers of n.

Computationally efficient algorithms are often $O(n)$.

- This is true of OLS, since the cost of forming $\boldsymbol{X}^\top\boldsymbol{X}$ is $O(np^2)$.
- This cost usually dominates the cost of inverting $\boldsymbol{X}^\top\boldsymbol{X}$ and multiplying it by $\boldsymbol{X}^\top\boldsymbol{y}$, unless p is large relative to n.

Efficient sorting methods are generally $O(n \log n)$.

Algorithms that are $O(n^2)$ or $O(n^3)$ are impractical when n is large.

Sometimes, there exists more than one algorithm to implement a particular method. If so, think carefully about how cost will increase with n before deciding which one to use.

Goodness of Fit

We cannot observe either the expected squared reducible error (7) or the irreducible error $\mathrm{Var}(u_i)$.

We can observe the **mean squared error**, or **MSE**, for the training set:

$$\mathrm{MSE}_{\mathrm{train}} = \sum_{i=1}^{n} \big(y_i - \hat{f}(\boldsymbol{x}_i)\big)^2. \tag{9}$$

But this **training MSE** is not measuring what we really care about.

For methods that over-fit severely, such as KNN with K small, the training MSE may grossly under-estimate prediction errors.

If we have a test dataset, we can estimate $\mathrm{E}(y_0 - \hat{f}(\boldsymbol{x}_0))^2$ by using

$$\mathrm{MSE}_{\mathrm{test}} = \sum_{i=1}^{n_T} \big(y_{0i} - \hat{f}(\boldsymbol{x}_{0i})\big)^2, \tag{10}$$

where n_T is the number of observations in the test sample.

But MSE_{test} is just an estimate! It might not be a good one if n_T is small or if the test sample is not representative of what we really care about.

See Figure 2.9. Black is f. Orange is linear (insane over-smoothing), green is insane under-smoothing, blue minimizes test MSE.

- The right panel shows training (grey) and test (orange) MSEs for a range of values of the smoothing parameter.
- The left panel shows the (x_i, y_i) pairs, the true function $f(x)$, and three sets of fitted values.
- Linear regression (orange curve) is severely biased, while the very flexible spline (green curve) has large variance. The less flexible spline (blue curve) performs best.
- The right panel shows that training MSE always falls as the model gets more flexible, eventually being much smaller than the irreducible MSE.
- In contrast, the test MSE first falls and then rises. The minimum is for the blue curve, but it is greater than the irreducible MSE.

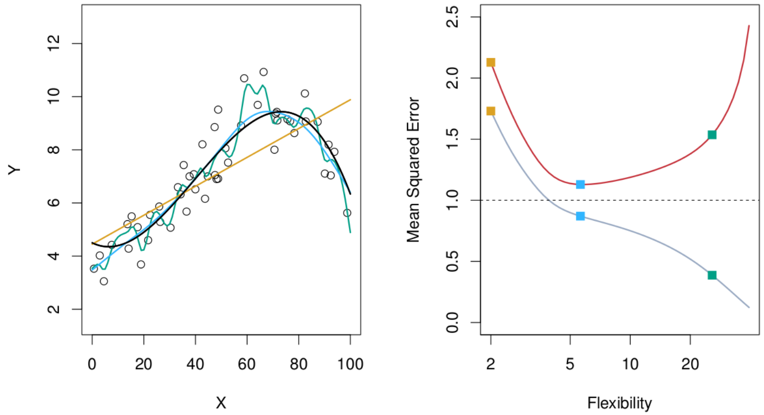

FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

The blue curve in Figure 2.9 was chosen to minimize MSE for the test sample, so that it is also being used as a validation sample.

In principle, it would be better if the validation and test samples were separate. Because they are the same in this case, the blue curve is guaranteed to minimize test MSE.

The choice of the smoothing parameter depends on the function $f(\boldsymbol{x})$ and on the sample.

- In Figure 2.10, $f(\boldsymbol{x})$ is almost (but not quite) linear, and it is therefore optimal to impose much more smoothness.
- In contrast, in Figure 2.11, $f(\boldsymbol{x})$ is highly nonlinear, and it is therefore optimal to impose much less smoothness.

It is obvious that Figures 2.9–2.11 did not use KNN! In fact, the orange curves are based on linear regression, and the blue and green curves are based on smoothing splines; see Chapter 7.

Goodness of Fit

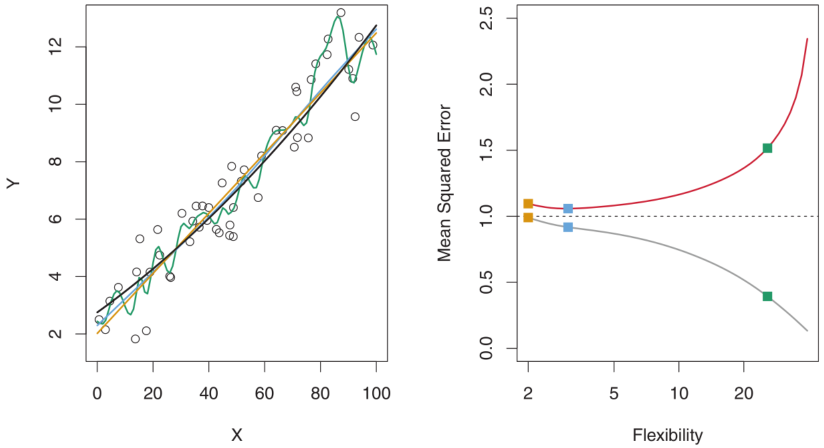

FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

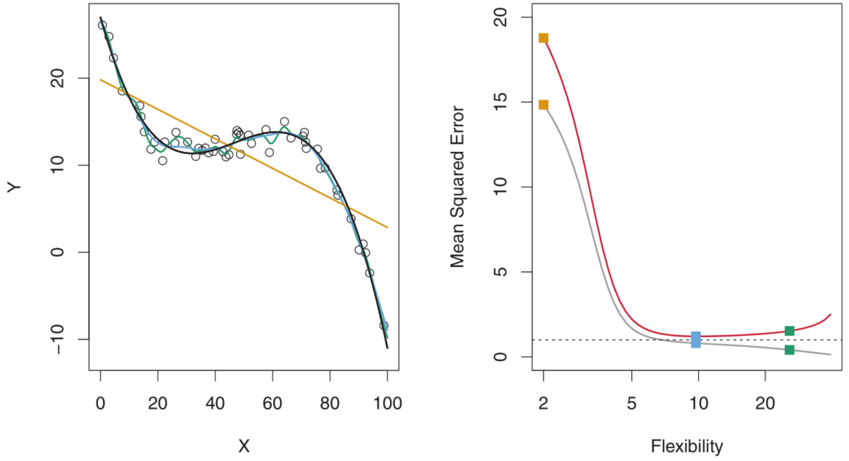

FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

In general, there is a tradeoff between bias and variance. The choice of smoothing parameter (*K* for KNN) determines where we land on that tradeoff.

For any regression method (not just KNN), we can write

$$\mathrm{E}\big(y_0 - \hat{f}(\boldsymbol{x}_0)\big)^2 = \mathrm{Var}\left(\hat{f}(\boldsymbol{x}_0)\right) + \Big(\mathrm{Bias}\big(\hat{f}(\boldsymbol{x}_0)\big)\Big)^2 + \mathrm{Var}(u_0). \tag{11}$$

The expected test MSE is the sum of the variance, the squared bias, and the irreducible error.

For KNN, the variance declines as *K* increases, but the squared bias increases; see (7).

Similar things happen for other methods, such as smoothing splines.

Imposing greater smoothness (allowing less flexibility) reduces variance but increases squared bias.

Figure 2.12 shows three curves for each of the three cases shown in Figures 2.9, 2.10, and 2.11. The flexibility (smoothness) parameter is on the horizontal axis.

The blue curves show squared bias. It always declines as smoothness (flexibility parameter) increases.

The orange curves show variance. It always increases as smoothness (flexibility parameter) increases.

The red curves show test MSE, which is the sum of variance and squared bias. It always drops at first and then rises.

- Very nonlinear $f(x)$ (rightmost) $\longrightarrow$ bad results with too much smoothness.
- Almost linear $f(x)$ (middle) $\longrightarrow$ bad results with too little smoothness.
- Intermediate $f(x)$ (leftmost) $\longrightarrow$ bad results at both ends.

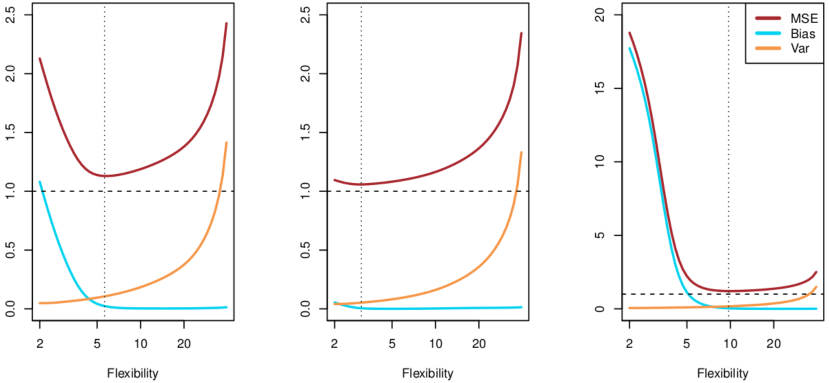

FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

In Figures 2.9–2.11, there is only one input (predictor, regressor).

The **curse of dimensionality** implies that methods like KNN suffer from much more bias as p increases.

The curse of dimensionality says that a given number of observations fill a space of dimension p less densely as p increases.

Thus the K nearest neighbours become farther away from $\boldsymbol{x}_0$ as p increases, and the bias increases.

- Some methods (KNN, smoothing splines) are designed for low-dimensional problems (p small). They can approximate very complicated, nonlinear relationships.
- Other methods (lasso, ridge regression) are designed for high-dimensional problems. They are much less flexible.
- But there is always a bias-variance tradeoff, and there is (at least implicitly) a smoothing (or tuning) parameter that determines where we end up on the tradeoff curve.