Statistical Learning for Data Science

Lecture 06

唐晓颖

电子与电气工程系

南方科技大学

March 6, 2023

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

1. Train with more data

With the increase in the training data, the crucial features to be extracted become prominent. The model can recognize the relationship between the input attributes and the output variable. The only assumption in this method is that the data to be fed into the model should be clean; otherwise, it would worsen the problem of overfitting.

2. Data augmentation

An alternative method to training with more data is data augmentation, which is less expensive and safer than the previous method. Data augmentation makes a sample data look slightly different every time the model processes it.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

3. Addition of noise to the input data

Another similar option as data augmentation is adding noise to the input and output data. Adding noise to the input makes the model stable without affecting data quality and privacy while adding noise to the output makes the data more diverse. Noise addition should be done in limit so that it does not make the data incorrect or too different.

4. Feature selection

Every model has several parameters or features depending upon the number of layers, number of neurons, etc. The model can detect many redundant features or features determinable from other features leading to unnecessary complexity. We very well know that the more complex the model, the higher the chances of the model to overfit.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

5. Cross-validation

Cross-validation is a robust measure to prevent overfitting. The complete dataset is split into parts. In standard K-fold cross-validation, we need to partition the data into k folds. Then, we iteratively train the algorithm on k-1 folds while using the remaining holdout fold as the test set. This method allows us to tune the hyperparameters of the neural network or machine learning model and test it using completely unseen data.

6. Simplify data

Till now, we have come across model complexity to be one of the top reasons for overfitting. The data simplification method is used to reduce overfitting by decreasing the complexity of the model to make it simple enough that it does not overfit. Some of the procedures include pruning a decision tree, reducing the number of parameters in a neural network, and using dropout on a neutral network.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

7. Regularization

If overfitting occurs when a model is too complex, reducing the number of features makes sense. Regularization methods like Lasso, L1 can be beneficial if we do not know which features to remove from our model. Regularization applies a "penalty" to the input parameters with the larger coefficients, which subsequently limits the model's variance.

8. Ensembling

It is a machine learning technique that combines several base models to produce one optimal predictive model. In Ensemble learning, the predictions are aggregated to identify the most popular result. Well-known ensemble methods include bagging and boosting, which prevents overfitting as an ensemble model is made from the aggregation of multiple models.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

9. Early stopping

This method aims to pause the model's training before memorizing noise and random fluctuations from the data. There can be a risk that the model stops training too soon, leading to underfitting. One has to come to an optimum time/iterations the model should train.

10. Adding dropout layers

Large weights in a neural network signify a more complex network. Probabilistically dropping out nodes in the network is a simple and effective method to prevent overfitting. In regularization, some number of layer outputs are randomly ignored or “dropped out” to reduce the complexity of the model.

Our tip: If one has two models with almost equal performance, the only difference being that one model is more complex than the other, one should always go with the less complex model. In data science, it's a thumb rule that one should always start with a less complex model and add complexity over time.

The Classification Setting

- *Training error rate*

$$\frac{1}{n}\sum_{i=1}^{nTr} I(y_i \neq \hat{C}(x_i))$$

- *Test error rate*

$$\mathrm{Ave}_{i\in Te}\left(I(y_i \neq \hat{C}(x_i))\right)$$

The Classification Setting

- *The Bayes Classifier*

Assigns each observation to the most likely class, given its predictor values

Let $p_k(x) = \Pr(Y = k \mid X = x)$

Then the Bayes Optimal Classifier at x is

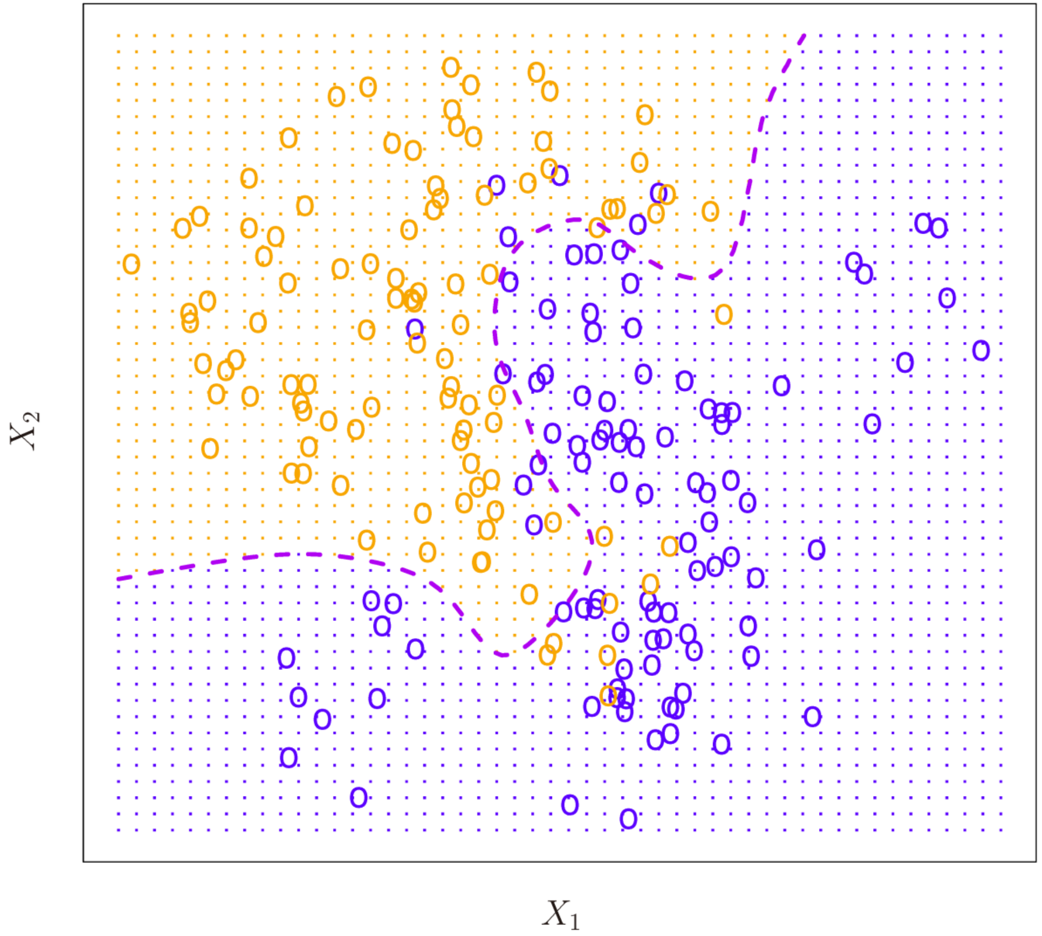
$$C(x) = j \text{ if } p_j(x) = \max\left\{p_1(x), p_2(x), \ldots, p_K(x)\right\}$$

In a two-class classification problem

$$C(x) = 1 \text{ if } p_1(x) > 0.5$$

The Classification Setting

- *The Bayes Classifier*



--orange shaded region

$\Pr(Y = \text{orange} \mid X) > 0.5$

--blue shaded region

$\Pr(Y = \text{blue} \mid X) > 0.5$

--purple dashed line

$\Pr(Y = \text{orange} \mid X)$

$= \Pr(Y = \text{blue} \mid X)$

$= 0.5$

"Bayes Decision Boundary"

The Classification Setting

- *The Bayes Classifier*

The Bayes classifier produces the lowest possible test error rate, called the *Bayes error rate*. This quantity is analogous to the irreducible error.

The error rate at $X = x$ will be

$$1 - \max_k \Pr(Y = k \mid X = x)$$

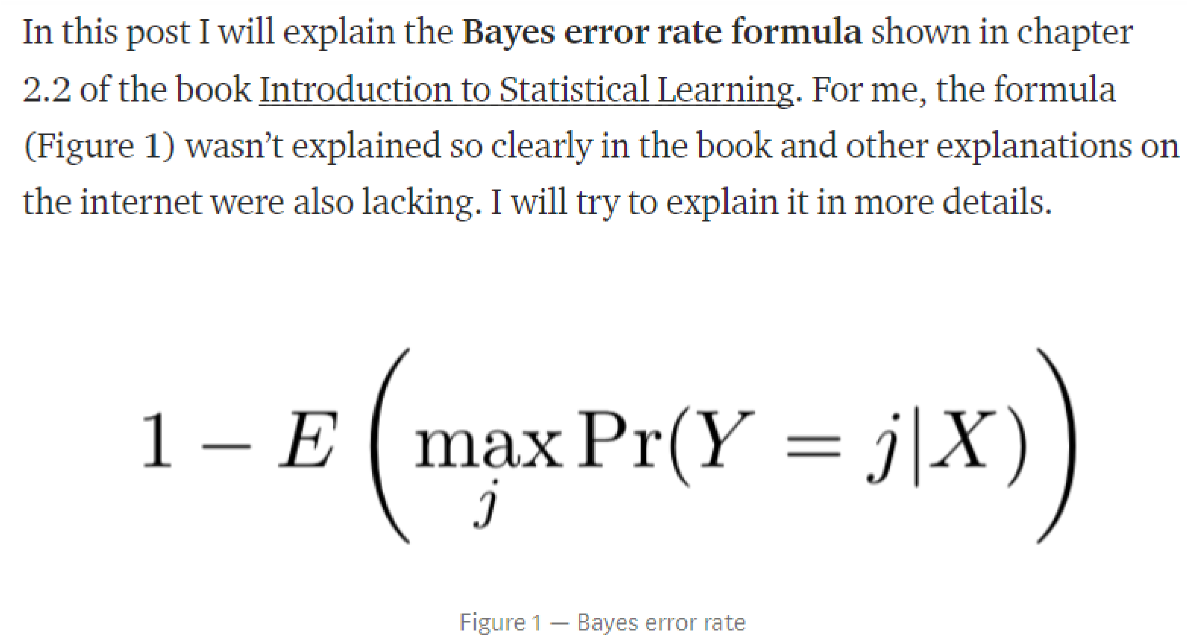
The overall Bayes error rate is given by

$$1 - E\left(\max_k \Pr(Y = k \mid X)\right)$$

https://www.youtube.com/watch?v=_BJccZ7snK4

The Classification Setting

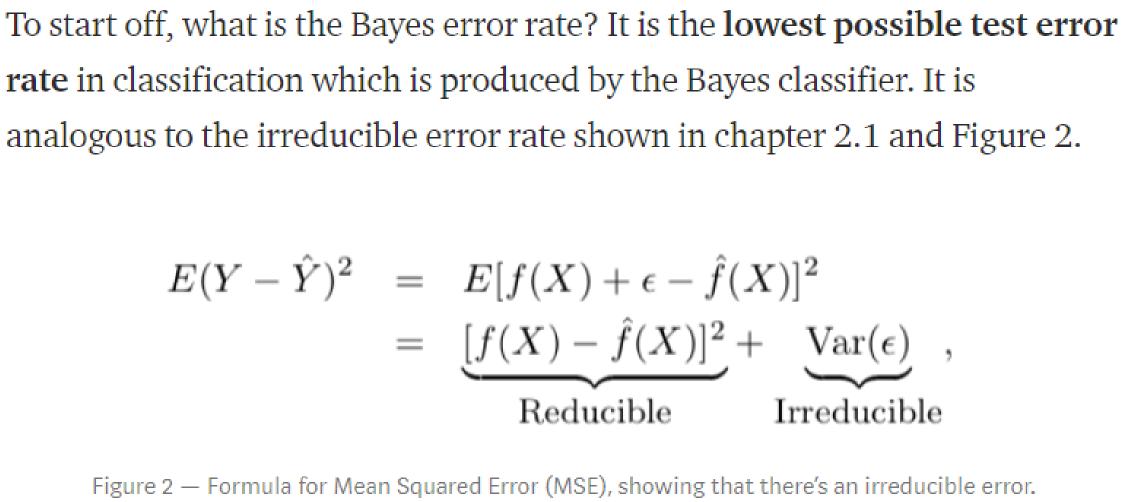
- *The Bayes Error Rate* https://medium.com/@hipstering/bayes-error-rate-1cd1f81c16ec



In this post I will explain the **Bayes error rate formula** shown in chapter 2.2 of the book Introduction to Statistical Learning. For me, the formula (Figure 1) wasn't explained so clearly in the book and other explanations on the internet were also lacking. I will try to explain it in more details.

$$1 - E\left(\max_j \Pr(Y = j|X)\right)$$

Figure 1 — Bayes error rate



To start off, what is the Bayes error rate? It is the **lowest possible test error rate** in classification which is produced by the Bayes classifier. It is analogous to the irreducible error rate shown in chapter 2.1 and Figure 2.

$$\begin{aligned} E(Y-\hat{Y})^2 &= E[f(X)+\epsilon-\hat{f}(X)]^2 \\ &= \underbrace{[f(X)-\hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}}, \end{aligned}$$

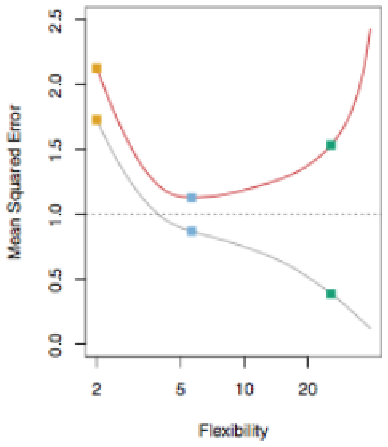
Figure 2 — Formula for Mean Squared Error (MSE), showing that there's an irreducible error.

The Classification Setting

- *The Bayes Error Rate*

https://medium.com/@hipstering/bayes-error-rate-1cd1f81c16ec

This error rate can be seen graphically as well. In red and blue we have respectively the test and training MSEs that vary with the flexibility of a model. The test MSE will never go below the dotted line since that's that representation of the irreducible error.



Test MSE vs Model flexibility

We can calculate the Bayes error rate with the formula shown in Figure 1, but what does it all mean? We'll analyse it by parts.

$$1 - E\left(\max_j \boxed{\Pr(Y = j|X)}\right)$$

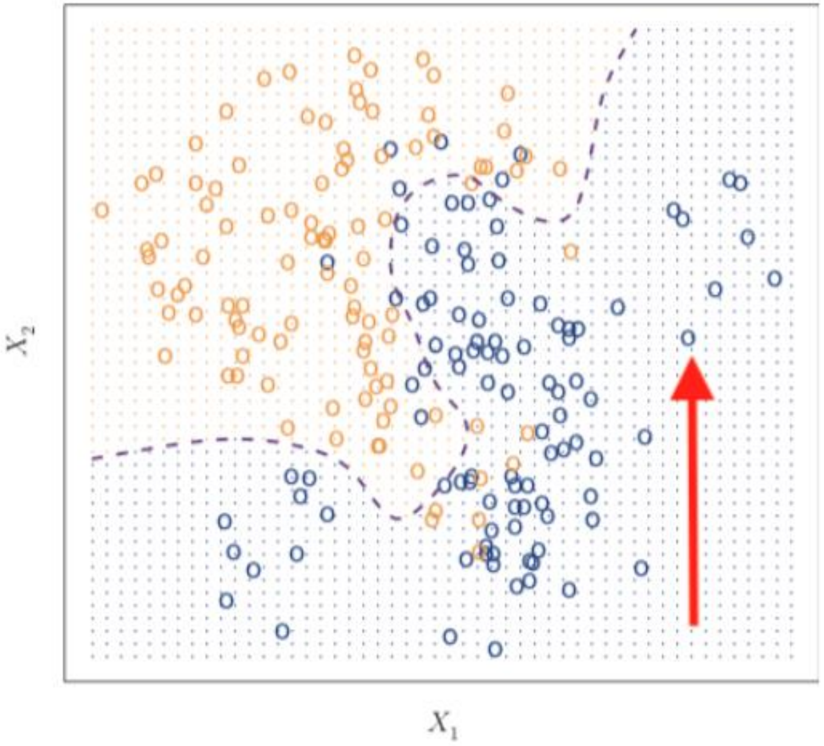
Formula for calculating Bayes error rate

Consider we have data X and an output Y. The Bayes classifier will assign one of two classes to an observation. The highlighted part gives us the probability of it belonging to the class j. In the book this is clear. However, max *j* is not explained so well. max *j* simply means that it will choose the **class with the highest probability**. If the observation has 0.8 chance of being orange, it will choose this probability. In this case j = orange and the value inside the parenthesis = 0.8

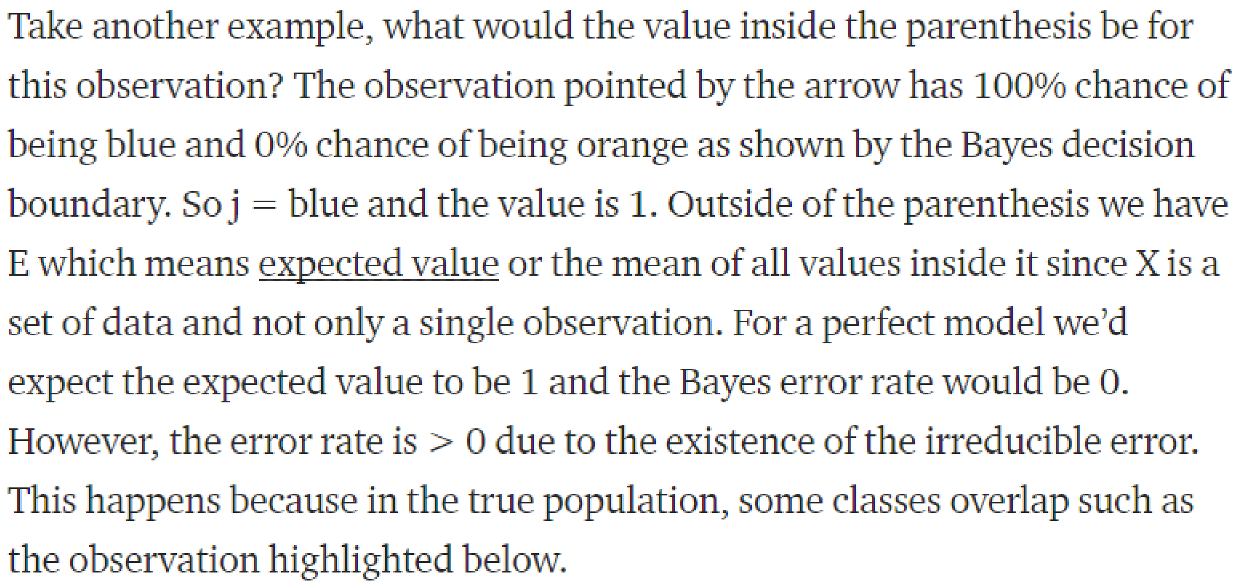
The Classification Setting

- *The Bayes Error Rate*

https://medium.com/@hipstering/bayes-error-rate-1cd1f81c16ec



Observation with 100% chance of being blue

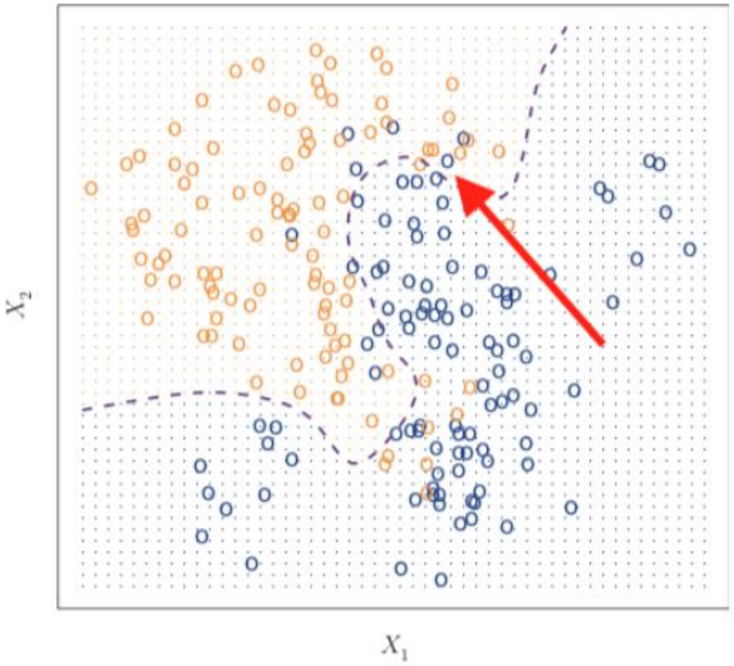


Take another example, what would the value inside the parenthesis be for this observation? The observation pointed by the arrow has 100% chance of being blue and 0% chance of being orange as shown by the Bayes decision boundary. So j = blue and the value is 1. Outside of the parenthesis we have E which means expected value or the mean of all values inside it since X is a set of data and not only a single observation. For a perfect model we'd expect the expected value to be 1 and the Bayes error rate would be 0. However, the error rate is > 0 due to the existence of the irreducible error. This happens because in the true population, some classes overlap such as the observation highlighted below.

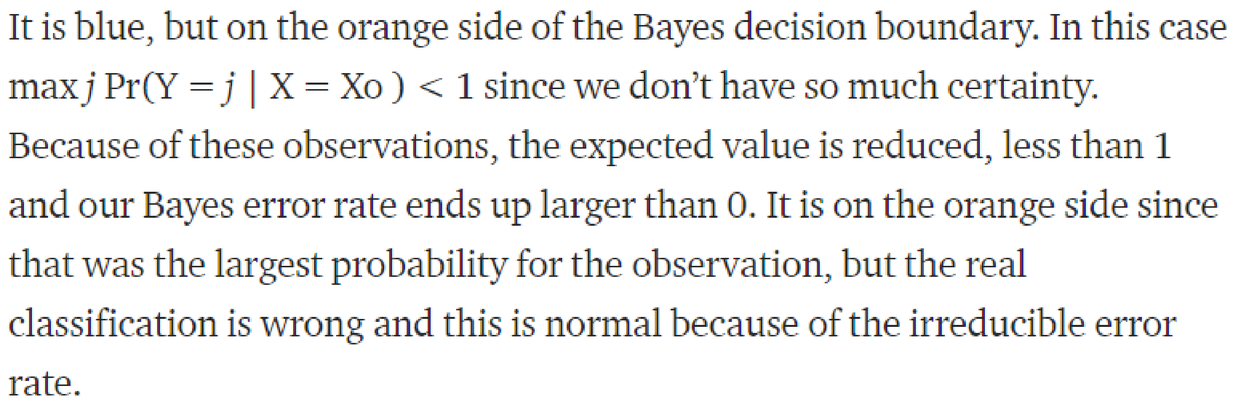
The Classification Setting

- *The Bayes Error Rate*

https://medium.com/@hipstering/bayes-error-rate-1cd1f81c16ec



Observation in which classes are not so clear and overlap. Thus, chance of being a certain class doesn't equal 1.



It is blue, but on the orange side of the Bayes decision boundary. In this case $\max j \Pr(Y = j \mid X = Xo) < 1$ since we don't have so much certainty. Because of these observations, the expected value is reduced, less than 1 and our Bayes error rate ends up larger than 0. It is on the orange side since that was the largest probability for the observation, but the real classification is wrong and this is normal because of the irreducible error rate.

The Classification Setting

- *K-Nearest Neighbors*
  - For real data, we do not know the conditional distribution of Y given X , and so computing the Bayes classifier is impossible.
  - Many approaches attempt to estimate the conditional distribution of Y given X , and then classify a given observation according to the class with highest estimate probability

The Classification Setting

- *K-Nearest Neighbors*

1. Identify the K points in the training data that are closest to $X = x_0$, represented by N_0

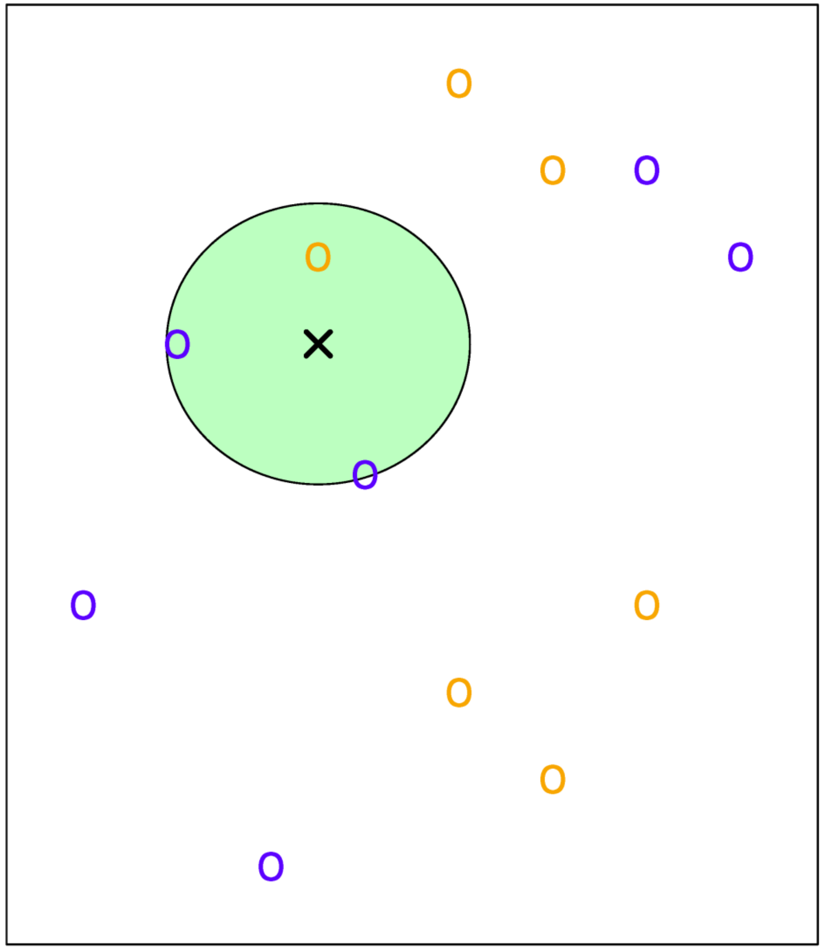
2. Estimate the conditional probability for class j as

$$\Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

3. Apply Bayes rule and classifies the test observation $X = x_0$ to the class with the largest probability.

The Classification Setting

- *K-Nearest Neighbors*



--Two categories: Blue vs. Yellow

--Goal: to predict the class of the point labeled by the black cross

--We use 3-Nearest Neighbors

⬇

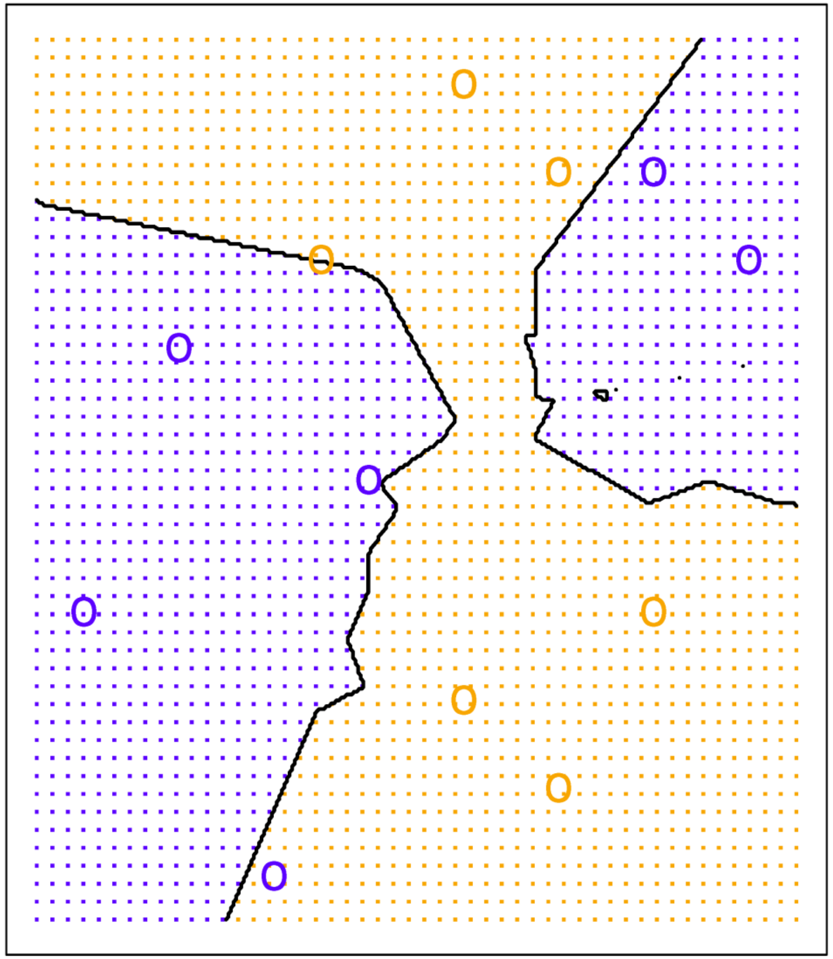
$$\Pr(Y = \text{Blue} \mid X = \text{black cross}) = \frac{2}{3}$$

$$\Pr(Y = \text{Orange} \mid X = \text{black cross}) = \frac{1}{3}$$

"3-NN will predict that the black cross belongs to the blue class"

The Classification Setting

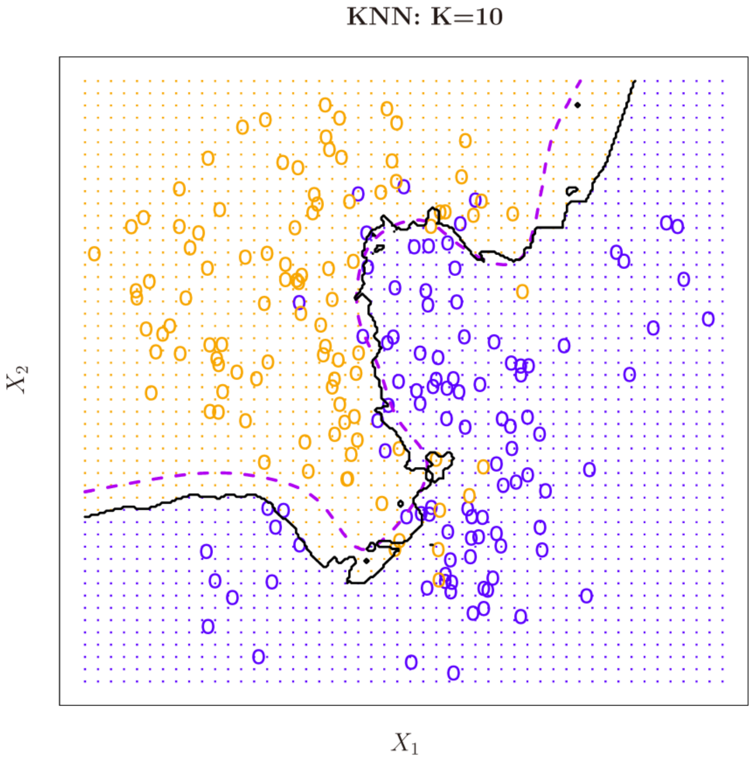
- *K-Nearest Neighbors (K=3)*



The Classification Setting

- *K-Nearest Neighbors*

Despite being very simple, KNN can often produce classifiers that are surprisingly close to the optimal Bayes classifier.



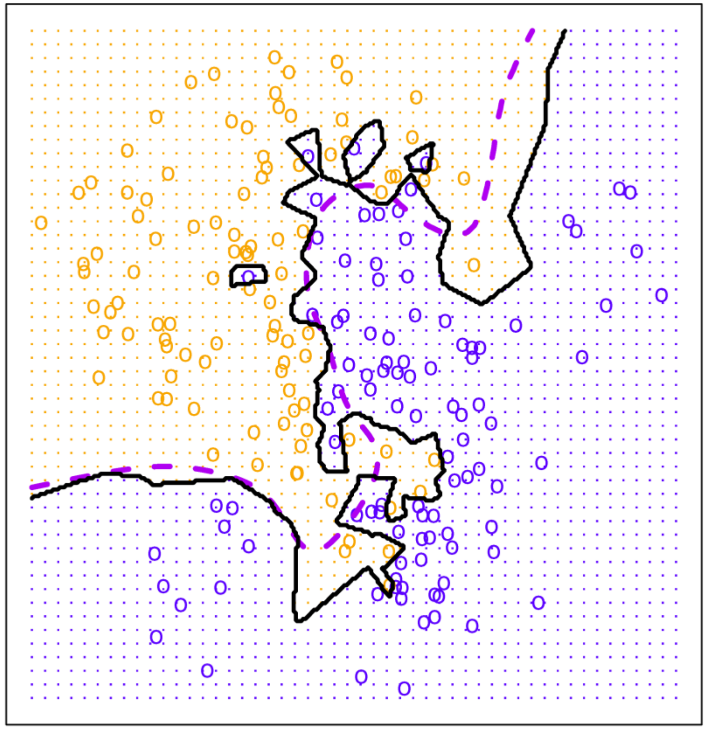
--Bayes error rate: 0.1304
--10-NN test error rate: 0.1363

The Classification Setting

- *K-Nearest Neighbors*

The choice of K has a drastic effect on the KNN classifier's performance

KNN: K=1



--Bayes error rate: 0.1304
--1-NN test error rate: 0.1695

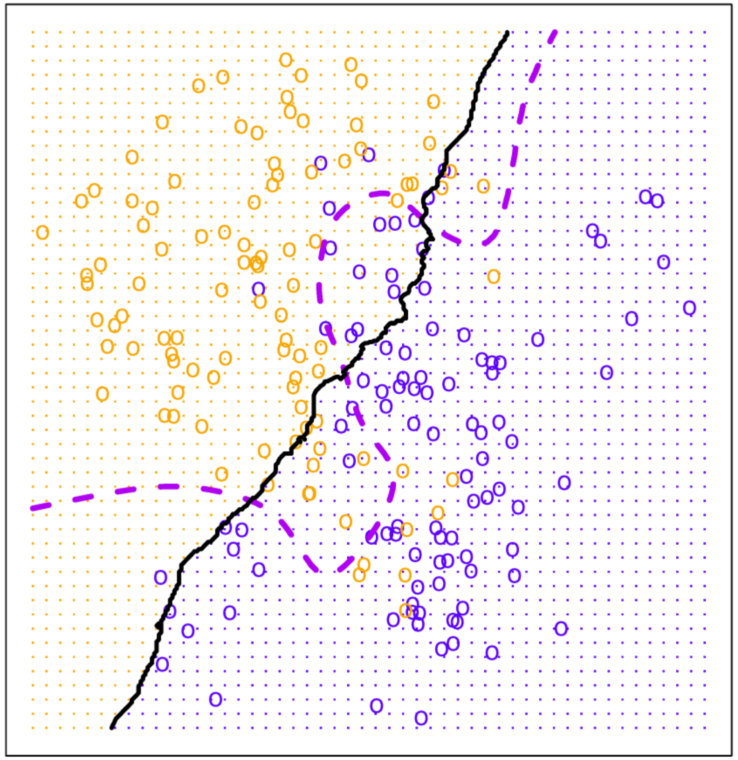
When K=1, the decision boundary is overly flexible. High variance, low bias.

The Classification Setting

- *K-Nearest Neighbors*

The choice of K has a drastic effect on the KNN classifier's performance

KNN: K=100



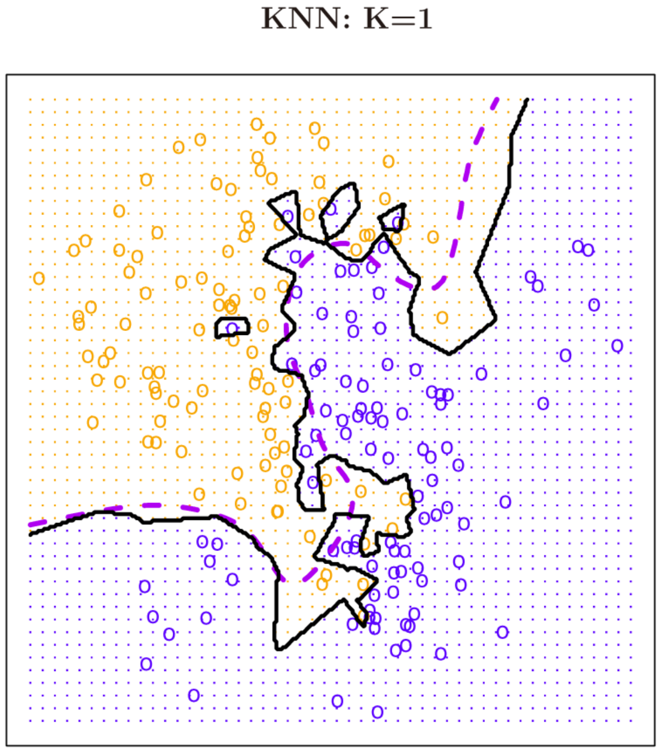
--Bayes error rate: 0.1304
--100-NN test error rate: 0.1925

When K=100, the decision boundary is less flexible, close to linear. Low variance, high bias.

The Classification Setting

- *K-Nearest Neighbors*

Similar to regression, there is no strong relationship between the training error rate and the test error rate.

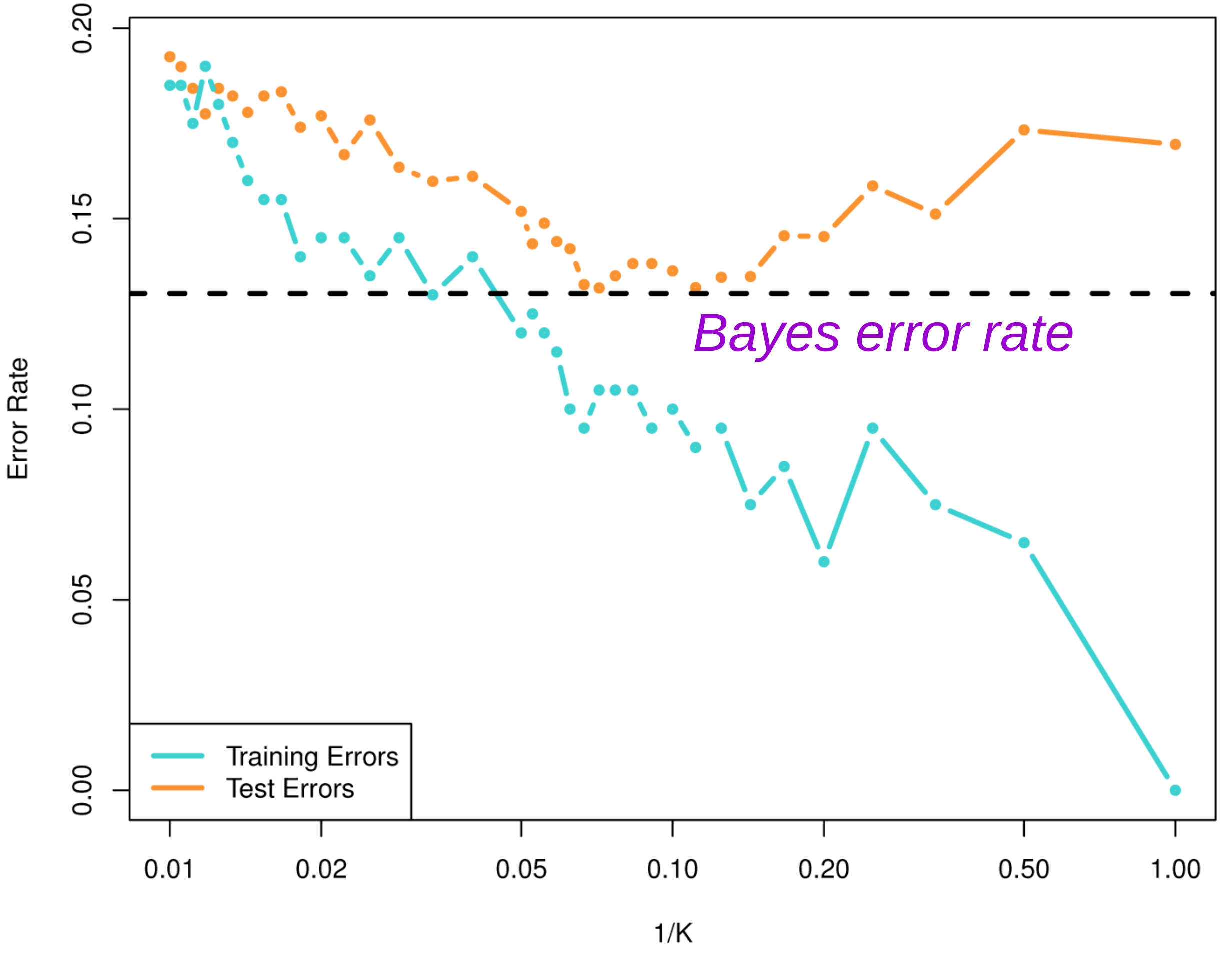


--Bayes error rate: 0.1304
--1-NN test error rate: 0.1695
--1-NN training error rate: 0

In general, as we use more flexible classification methods, the training error rate will decline but the test error rate may not

The Classification Setting

- *K-Nearest Neighbors*



--the training error rate consistently declines as the flexibility increases

--the test error rate exhibits a U-shape, declining at first before increasing again when the method becomes excessively flexible and over-fits

Assessing Model Accuracy

In both the regression and classification settings, choosing the correct level of flexibility is critical to the success of any statistical learning method. The bias-variance trade-off, and the resulting U-shape test error, can make this a difficult task.

Later, we will return to this topic and discuss various methods for estimating test error rates and thereby choosing the optimal level of flexibility for a given statistical learning method.