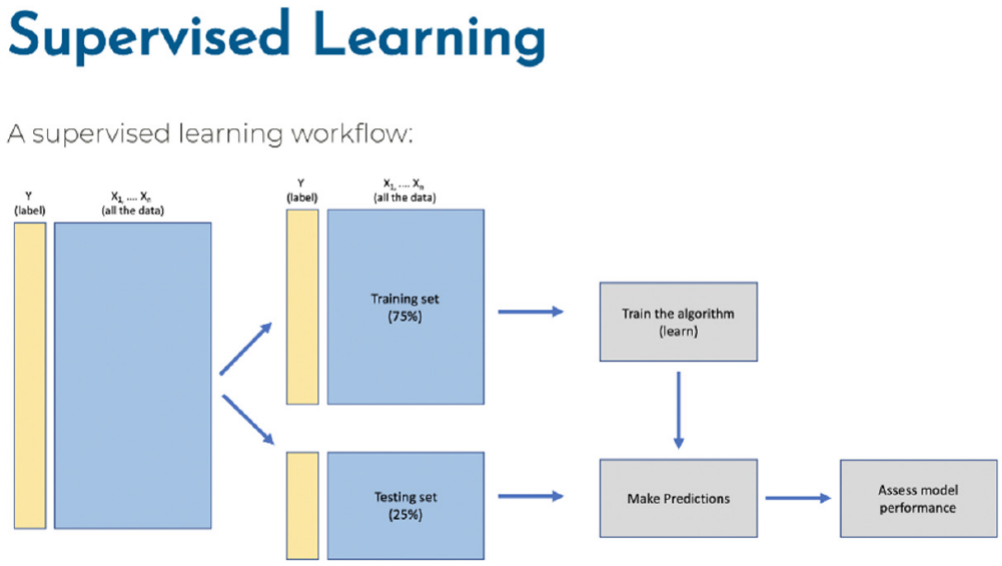
We acknowledge the Australian Aboriginal and Torres Strait Islander peoples as the traditional owners of the lands and waters where we live and work.

Week 2: Introduction to Data Mining

Dr Max Cao
Performance and Insights Manager at Brisbane City Council & Lecturer at James Cook University

Supervised Learning



Predictions Errors-Bias-Variance Tradeoff

Data Visualization in tabular format-Training dataset in ISLR chapter 1

Observation	Response	Predictor 1	Predictor 2			Predictor p
1	y_1	x_{11}	x_{12}	.	.	x_{1p}
2	y_2	x_{21}	x_{22}	.	.	x_{2p}
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
n	y_n	x_{n1}	x_{n2}	.	.	x_{np}

- We wish to find a model for real observations –
  $y_i = f(x_{i1}, x_{i2}, \dots, x_{ip})(a\ model\ of\ reality) + \varepsilon_i\ (irreducible\ error)$
- Objective, find a good approximate
  $\hat{f}(x_{i1}, x_{i2}, \dots, x_{ip})$ for $f(x_{i1}, x_{i2}, \dots, x_{ip})$
- So that prediction of y_0 becomes possible
  $\hat{y}_0 = \hat{f}(x_{01}, x_{02}, \dots, x_{0p})$

Predictions Errors-Bias-Variance Tradeoff

A good model must emulate the data closely. One way to summarize this is to obtain the average 'gap' between observed y_i and estimated/predicted data $\hat{y}_i$.

1. Compute from training data

   $Average\{(y_1 - \widehat{y_1})^2, (y_2 - \widehat{y_2})^2, \dots, (y_n - \widehat{y_n})^2\}$: *the Training MSE*.

   Easy to compute but how useful?

2. But we are more interested in

   $Average\{(y_0 - \widehat{y_0})^2\}$: *the Test MSE*

   But often harder to compute, sometimes not possible.

3. What is test data? Strategies that often work
   a. Train, Test split
   b. Or we set aside some data, *randomly* (random sub-sampling). E.g – Cross-validation

Predictions Errors-Bias-Variance Tradeoff

A model that exhibits small variance and high bias will **underfit** the target, while a model with high variance and little bias will **overfit** the target. How to minimize these errors?

What is MSE? Why MSE is important in Bias –Variance Trade off?

MSE can be written as the sum of the variance of the estimator and the squared bias of the estimator, providing a useful way to calculate the MSE and implying that in the case of unbiased estimators, the MSE and variance are equivalent.

Fundamental equation of Mean Squared Error: Test Data(MSE)

$$E\{(y_0 - \widehat{y_0})^2\} = Var\{\hat{f}(x_{01}, x_{02}, \dots, x_{0n})\} + \left[Bias\{\hat{f}(x_{01}, x_{02}, \dots, x_{0n})\}^2\right] + Var\ (\varepsilon_0)$$

$Var\{\hat{f}(x_{01}, x_{02}, \dots, x_{0n})\}$: Error due to sampling-Overfitting(complex and too noisy)

- Error caused by using a different training data (or sample).
- A large training helps reduce test variance.
- A simpler model $\hat{f}(.)$ (that is less flexible) usually has a lower variance on test data.
- As the model flexibility increases the test MSE gradually increases.

$Bias\{\hat{f}(x_{01}, x_{02}, \dots, x_{0n})\}$: Error due to modelling- Underfitting(overlooks irregularities in the data)

- Error due to simplification of reality by a mathematical model- (e.g)if the data is non-linear in the linear regression, or data becomes skewed after transformations.
- Even if your you have a large training sample you can't reduce the bias if $\hat{f}(.)$ is very different from $f(.)$
- But if $\hat{f}(.)$ is too complex (that is fits the training data too well) it would have poor test MSE.

$Var\ (\varepsilon_0)$: is irreducible error.

- So this is the lower bound –minimum achievable value- of the test MSE $E\{(y_0 - \widehat{y_0})^2\}$.

Predictions Errors-Bias-Variance Tradeoff

As the model complexity increases in test data

1. Bias: $Bias\{\hat{f}(x_{01}, x_{02}, \dots, x_{0n})\}$ can decrease rapidly and then saturates. Why?
2. Variance: $Var\{\hat{f}(x_{01}, x_{02}, \dots, x_{0n})\}$ increases gradually but then sharply. Why?
3. Test MSE initially decreases (due to 1) reaches a minimum and then starts increasing (due to 2).
4. How to attain the sweet spot? Middling the parameters?

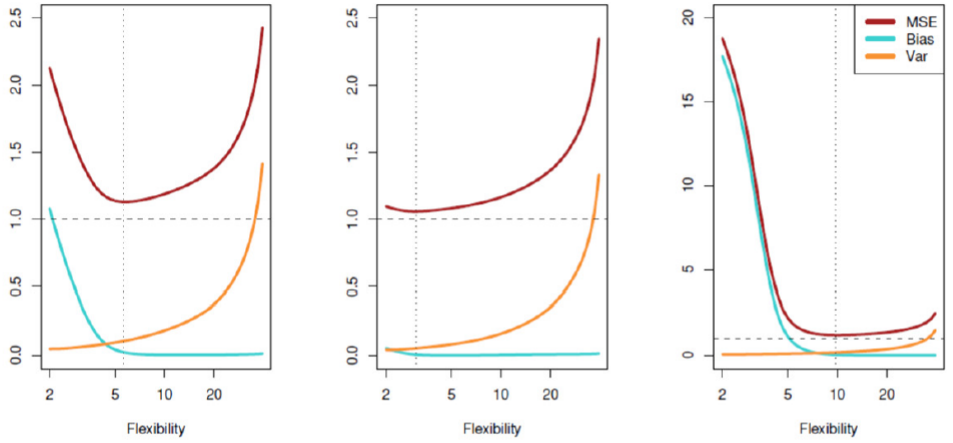
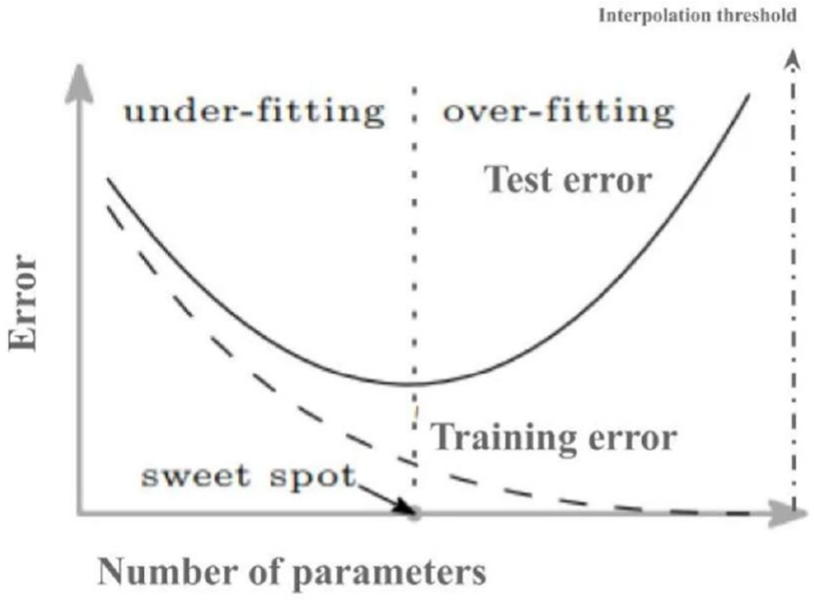


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), $Var(\epsilon)$ (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Source: ISLR Chapter 2

The flexibility of a design increases, the usability and performances of the design decreases.

Predictions Errors-Bias-Variance Tradeoff



Typical plot of error relative to the number of parameters fit in a model. The training error (dashed line) approaches zero as the interpolation threshold (dot dashed line) is approached. The test error (solid line) shows the classical U shape from the bias-variance trade-off. The number of parameters to minimize test error is indicated by the "sweet spot." Figure by the author, based on figure 1a from [1].

More parameters always improve the *fit* to a data set, just like more pixels on a camera always improve the realism of the photo. With enough parameters, the model can interpolate the data, meaning the error is zero. This is called the interpolation threshold, and it happens when the number of parameters equals the number of examples, allowing the examples to be fit perfectly. You can add still more parameters, but the additional parameters cannot reduce error **because it's already zero.**

However, if these models are used to predict a different sample of data, such as the test data, then the error typically increases as the interpolation threshold is approached. Plotting the error on the test data set relative to the number of parameters typically results in a **U-shaped curve.** The point where the model again starts to perform well on the testing data is called the **interpolation threshold.**

To minimize test error, the optimal number of parameters lies between 0 and the interpolation threshold.

Predictions
Errors-Bias-Variance Tradeoff

Underfitting (Bias)	Overfitting(Variance)
Main Reasons -More Outliers, assumption fails	Main Reasons-Less outliers, fit all the assumptions
Few ways to reduce -Increase model complexity, Increase the features	Few ways to reduce -Simplify the model, Fearure selection(Variable Importance)
ML Techniques-Transformations	ML Techniques-Cross-Validation, Regularization.

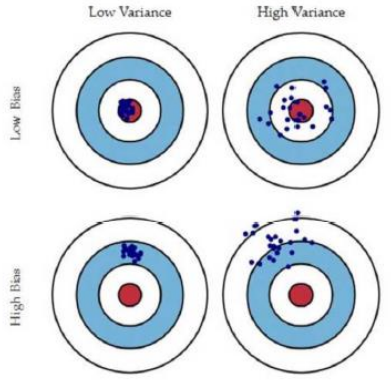


Fig. 6. Graphical illustration of bias and variance (Fortmann-Roe, 2012).

In figure at the right, each point represents one model that is trained by data. The centre of the area which all points occupy represents bias and the degree of dispersal of the points represents variance.

The centre of the target represents the area of zero error that can predict the correct value. As we move away from the centre, the error of the model would increase and predictions would get worse.

Naïve Bayes Classifier

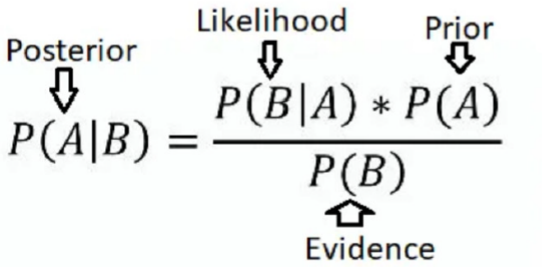
Supervised, probabilistic classifier based on Bayes Theorem

Bayes theory

Bayes theorem, by Reverend Thomas Bayes, is about **conditional probability as P(A|B);** the probability of A given that B occurred [also called evidence/predictor prior probability]. We encounter a new observation for which we know the values of the predictors X, but not the class Y, so we would like to make a guess about Y based on the information we have (our sample). The key insight of Bayes' theorem is that the probability of an event can be adjusted as new data is introduced.

Parameter estimation for naive Bayes models uses the method of maximum likelihood.

Given data the maximum likelihood estimate (MLE) for the parameter p is the value of p that maximizes the likelihood P(data |p). That is, the MLE is the value of p for which the data is most likely.



Prior: Probability distribution representing knowledge or uncertainty of a data object prior or before observing it.
Posterior: Conditional probability distribution representing what parameters are likely after observing the data object.
Likelihood: The probability of falling under a specific category or class.

Bayes' theorem (intuition)

Machine 1

Machine 2

Bayes' theorem (intuition)

JAMES COOK UNIVERSITY AUSTRALIA

What is the probability?

M2

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Bayes' theorem (intuition)

JAMES COOK UNIVERSITY AUSTRALIA

- Machine 1 (M1): 40 units/hour
- Machine 2 (M2): 30 units/hour
- Of all parts produced in a batch, there are 2% defective
- Of all defective product 50% from M1 and 50% from M2

Q1: What is the probability that a wrench produced by M2 is defective

Q2: What is the probability that a wrench produced by M1 is not defective

Bayes' theorem (intuition)

JAMES COOK UNIVERSITY AUSTRALIA

- Machine 1 (M1): 40 units/hour — P(M1) = 40/70 = 0.572
- Machine 2 (M2): 30 units/hour — P(M2) = 30/70 = 0.428
- There are 2% defective products — P(defect) = 2% = 0.02
- Of all defective product 50% from M1 and 50% from M2

P(M1|defect) = P(M2|defect) = 50% = 0.50

Q1: P(defect|M2)

Q2: 1 – P(defect|M1)

$$P(\text{defect}|M2) = \frac{P(M2|defect)*P(defect)}{P(M2)} \qquad \frac{0.50 * 0.02}{0.428} = 0.0233 = 2.33\%$$

Lets verify this with traditional theory

JAMES COOK UNIVERSITY AUSTRALIA

- 8400 produced in a batch
- M1 produced: 4800
- M2 produced: 3600
- There are 168 defective products (which is 2% of the production)
- M1 produced 84 and M2 produced 84 defective products

$$P(\text{defect} \mid M2) = \frac{Total\ defective\ by\ M2}{Total\ Production\ by\ M2} \qquad \frac{84}{3600} = 0.0233 = 2.33\%$$

Naïve Bayes Classifier

Actually, in order to apply the Bayes rule of classification we don't really need the exact probability values, we only need to know which probability is the largest. Since the denominator in Eq. (2) – that is, the prior $P(X_1 = x_1 \& \cdots \& X_n = x_n)$ – is the same irrespective of the class label, we actually only need to compute the numerator, that is, we only need to estimate $P(X_1 = x_1 \& \cdots \& X_n = x_n | Y = y)$ and $P(Y = y)$ for each class label $Y = y$. The problem is that the number of observations required to get a rough frequentist estimate of the former term, in principle, grows exponentially with the number of predictors, n, thus making the frequentist approach infeasible in most practical applications. The naive Bayes classifier circumvents this problem by making the assumption that the predictors are *statistically independent within each class*s, which means that no particular value taken by a certain predictor affects the probabilities associated with the other predictors within a given class. Mathematically, if the class conditional independence assumption holds true (for any given class label, $Y = y$), then:

$$P(X_1 = x_1 \& \cdots \& X_n = x_n | Y = y) = P(X_1 = x_1 | Y = y) * P(X_2 = x_2 | Y = y) * \cdots * P(X_n = x_n | Y = y)$$

This property makes it much easier to estimate $P(X_1 = x_1 \& \cdots \& X_n = x_n | Y = y)$ by independently estimating and multiplying the probabilities associated with each individual predictor. Under the independence

$$P(Y|X_1 \& \cdots \& X_n) = \frac{P(X_1 = x_1|Y = y)P(X_2 = x_2|Y = y) \cdots P(X_n = x_n|Y = y)P(Y)}{P(X_1 = x_1|Y = y) \quad P(X_2 = x_2|Y = y) \quad \cdots P(X_n = x_n|Y = y)} = \frac{P(Y) \prod_{i=1}^{n} P(X_i = x_i|Y = y)}{\sum_{j=1}^{k} P(Y = y_j) * \prod_{i=1}^{n} P(X_i = x_i|Y = y)}$$

Equation (2)

$$P(Y|X_1 \& \cdots \& X_n) = \frac{P(X_1 \& \cdots \& X_n|Y)P(Y)}{P(X_1 \& \cdots \& X_n)}$$

JAMES COOK UNIVERSITY AUSTRALIA

Naïve Bayes classifier – an example

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Naïve Bayes classifier – MAP rule

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

Zero frequency problem??

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

The final posterior probabilities can be standardized between 0 and 1

Naïve Bayes classifier – numerical predictor

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$ Mean

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5}$$ Standard deviation

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$ Normal distribution

		Humidity	*Mean*	*StDev*
Play Golf	yes	86 96 80 65 70 80 70 90 75	79.1	10.2
	no	85 90 70 95 91	86.2	9.7

$$P(\text{humidity} = 74 \mid \text{play} = \text{yes}) = \frac{1}{\sqrt{2\pi}(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}} = 0.0344$$

$$P(\text{humidity} = 74 \mid \text{play} = \text{no}) = \frac{1}{\sqrt{2\pi}(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}} = 0.0187$$

Naïve Bayes Classifier

Assumptions:

1) Predictors are conditionally independent.

Chi-squared test - < 0.05 (Ha True)

2) Can handle both discrete(Categorical) and continuous(Numeric)variables.

3) It also assumes that all features contribute equally to the outcome and most suitable for real time data(equal weight).

Naïve Bayes classifier types:

Factor variable - Multinomial Naive Bayes Classifier

Factor variables are categorical variables that can be either numeric or string variables. A "factor" is a vector whose elements can take on one of a specific set of values(levels).

Uses frequencies from the training data to calculate probabilities of each predictor in each class, or multinomial. Example, whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

Naïve Bayes Classifier

Bernoulli Naive Bayes:

This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

Continuous variables - Gaussian Naive Bayes Classifier

Uses Gaussian or Kernal (non-parametric) density estimate, to calculate the likelihood of each predictor in a class.

In a nut shell, Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. **Does Naïve Bayes performs good even if the assumptions are violated?**

Linear Discriminant Analysis

LDA: Linear Discriminant Analysis (LDA) is a generalization of Fisher's linear discriminant, a method used in Statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events.

Fisher's linear discriminant attempts to find the vector that maximizes the separation between classes of the projected data. Maximizing "separation" can be ambiguous. The criteria that Fisher's linear discriminant follows to do this is to maximize the distance of the projected means and to minimize the projected within-class variance.

This method projects a dataset onto a lower-dimensional space with good class-separability to avoid overfitting ("curse of dimensionality- as the number of features increase, our data become sparser, which results in overfitting, and we therefore need more data to avoid it"), and to reduce computational cost. Linear Discriminant Analysis or LDA is a **dimensionality reduction** technique. It is used as a pre-processing step in machine learning and applications of pattern classification.

It is based on "Guassian Bayes Classifier(normal distribution), uses probability density function.

Probability density function (PDF), density function, or density of an absolutely continuous random variable, is a function whose value at any given sample (or point) in the sample space (the set of possible values taken by the random variable) can be interpreted as providing a relative likelihood that the value of the random variable would be equal to that sample.

Linear Discriminant Analysis

How does LDA work?

LDA focuses primarily on projecting the features in higher dimension space to lower dimensions.

Firstly, calculate the separability between classes which is the distance between the mean of different classes. This is called the *between-class variance*.

•Secondly, calculate the distance between the mean and sample of each class. It is also called the within-class variance.

•Finally, construct the lower-dimensional space which maximizes the between-class variance and minimizes the within-class variance. P is considered as the lower-dimensional space projection, also called Fisher's criterion.

LDA reduces dimensionality from original number of feature to C — 1 features, where C is the number of classes. In this case, we have 3 classes, therefore the new feature space will have only 2 features.

Linear Discriminant Analysis

How to prepare data from LDA?

Some suggestions you should keep in mind while preparing your data to build your LDA model:

•LDA is mainly used in classification problems where you have a categorical output variable. It allows both binary classification and multi-class classification.

•The standard LDA model makes use of the Gaussian Distribution of the input variables. You should check the univariate distributions of each attribute and transform them into a more Gaussian-looking distribution. For example, for the exponential distribution, use log and root function.

•Outliers can skew the primitive statistics used to separate classes in LDA, so it is preferable to remove them.

•Since LDA assumes that each input variable has the same variance, it is always better to standardize your data before using an LDA model. Keep the mean to be 0 and the standard deviation to be 1.

Linear Discriminant Analysis

Like the Naive Bayes classifier, LDA also uses Bayes' theorem to indirectly estimate the probability $P(Y|X)$, but it does not make the assumption of (class conditional) independence of predictors,

the denominator is rewritten based on the fact that the class labels impose a partition of the sampling space and, therefore, if there are k classes, $Y = 1, \cdots, k$, then $P(X_1 \& \cdots \& X_n) = \sum_{i=1}^{k} P(Y = i) * P(X_1 \& \cdots \& X_n | Y = i)$. This property reads as "The probability of observing X_1 and X_2 and ... and X_n equals the probability of observing these predictor values given that the corresponding observation belongs to class $Y = 1$, plus the probability of observing these predictor values given that the corresponding observation belongs to class $Y = 2$, plus ... plus the probability of observing these predictor values given that the corresponding observation belongs to class $Y = n$". This is true because the class labels of

$$P(Y = y|X_1 = x_1 \& \cdots \& X_n = x_n) = \frac{P(X_1 = x_1 \& \cdots \& X_n = x_n|Y = y)\, P(Y = y)}{\sum_{i=1}^{k} P(X_1 = x_1 \& \cdots \& X_n = x_n|Y = i) * P(Y = i)}$$

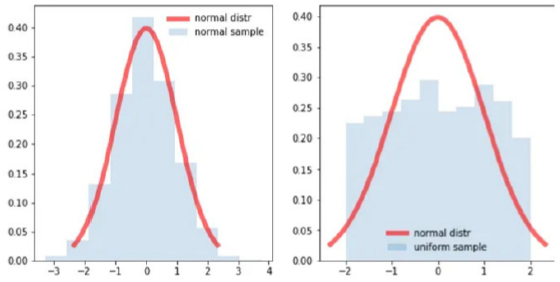
Linear Discriminant Analysis

Assumptions-LDA

1)Predictors are normally distributed (No Outliers) and it is a parametric model

Shapiro-Wilk test(Analytical)

Histogram, QQplot(Graphical)



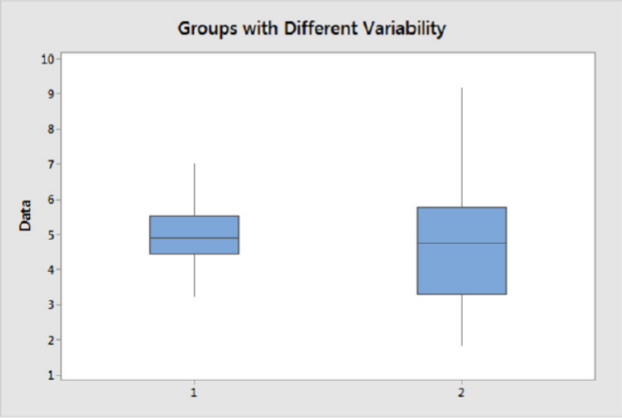
Parametric tests are those that make assumptions about the parameters of the population distribution from which the sample is drawn. This is often the assumption that the population data are normally distributed.

Non-parametric tests are “distribution-free” and, as such, can be used for non-Normal variables.

Linear Discriminant Analysis

2) Variances among group variables are the same across levels of predictors(X)against each levels of response variable(Y(0,1). This make is linear different from QDA(covariance matrix is not identical for different classes). It can be checked using standard deviation or F-test.

To assess variability in a box and whisker plot, remember that half your data for each group falls within the interquartile box. The longer the box and whiskers, the greater the variability of the distribution. The total length of the whiskers represents the range of the data. In the plot below, Group 2 has more variability than Group 1 because it has a longer box and whiskers. Group 1 ranges from approximately 3 to 7 while Group 2 ranges from roughly 1.5 to 9 or



3) The predictors can't be categorical variables and it has to be continuous variables .

Quadratic Discriminant Analysis

QDA follows the same model as LDA, except that it drops the assumption that the class conditional distributions must share a common covariance matrix. In other words, QDA models the distribution of each class by means of an independent multivariate Normal probability density function.

1)Predictors are normally distributed and it is a parametric model(No Outliers).

2)Each predictors can have its own covariance each levels of response variable(Y).

3) The predictors can't be categorical variables and it has to continuous variables.

The main implications of allowing class conditional distributions with different covariances are twofold. On the one hand, the model is more flexible in the sense that its assumptions are less restrictive in practice (i.e., they are more likely to be met, at least approximately) as well as in the sense that the resulting decision boundary is no longer necessarily linear (it can be non-linear). On the other hand, there is the need to estimate multiple covariance matrices, which means that the number of parameters to be estimated is larger, thus making the model and its training mathematically and computationally more complex as well as more prone to overfitting.

JAMES COOK UNIVERSITY AUSTRALIA

LDA and QDA assumption

- Both LDA and QDA assume the predictor variables *X* are drawn from a multivariate Gaussian (ie, *normal*) distribution
- LDA assumes equality of covariances among the predictor variables *X* across each all levels of *Y*
- QDA assumes that each class has its own covariance matrix
- LDA and QDA require the number of predictor variables (*p*) to be less then the sample size (*n*).
- Performance will severely decline as *p* approaches *n*. A simple rule of thumb is to use LDA & QDA on data sets where $n \geq 5 \times p$.

JAMES COOK UNIVERSITY AUSTRALIA

Relations between Naive Bayes, LDA and QDA

- LDA is just a particular case of QDA with more restrictive assumptions
- Both LDA and QDA require numerical predictors
- Naive Bayes can operate with categorical and mixed (i.e., numerical and categorical) predictors as well
- when all predictors are numerical, Naive Bayes is a particular case of LDA and, accordingly, it is also a particular case of QDA
- These classifiers are not affected by variable rescaling

Cross-Validations

Why cross-validation?

The most popular cross-validation is k-fold cross-validation.

K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation.

When the cv argument is an integer, cross_val_score uses the K-Fold or Stratified K-Fold strategies by default.

Few other cross-validation techniques: Repeated K-Fold – Repeated K-Fold repeats K-Fold n times. It can be used when one requires to run K-Fold n times, producing different splits in each repetition.

Leave One Out Cross Validation: Leave out one data point and build the model on the rest of the data set.

Cross-Validations

The k-fold cross-validation method evaluates the model performance on different subset of the training data and then calculate the average prediction error rate. The algorithm is as follow:

1. Randomly split the data set into k-subsets (or k-fold) (for example 5 subsets)
2. Reserve one subset and train the model on all other subsets
3. Test the model on the reserved subset and record the prediction error
4. Repeat this process until each of the k subsets has served as the test set.
5. Compute the average of the k recorded errors. This is called the cross-validation error serving as the performance metric for the model.

For example, let's suppose that we have a dataset $S = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ containing 6 samples and that we want to perform a 3-fold cross-validation.

First, we divide S into 3 subsets randomly. For instance:

$S_1 = \{x_1, x_2\}$
$S_2 = \{x_3, x_4\}$
$S_3 = \{x_5, x_6\}$

Then, we train and evaluate our machine-learning model 3 times. Each time, two subsets form the training set, while the remaining one acts as the test set. In our example:

Cross-Validations

In the leave-one-out (LOO) cross-validation, we train our machine-learning model n times where n is to our dataset's size. Each time, only one sample is used as a test set while the rest are used to train our model.

We'll show that LOO is an extreme case of k-fold where{k=n}. If we apply LOO to the previous example, we'll have 6 test subsets: Cross-Validations

$S_1 = \{x_1\}$
$S_2 = \{x_2\}$
$S_3 = \{x_3\}$
$S_4 = \{x_4\}$
$S_5 = \{x_5\}$
$S_6 = \{x_6\}$

Iterating over them, we use $S \setminus S_i$ as the training data in iteration $i = 1, 2, \ldots, 6$, and evaluate the model on S_i

References

Textbook- An introduction to Statistical Learning with Application in R- Authors Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani .

Texbook online reference -https://www.stat.berkeley.edu/users/rabbee/s154/ISLR_First_Printing.pdf

https://github.com/MarthaCooper/

https://www.stat.cmu.edu

https://www.geeksforgeeks.org/

https://medium.com/

https://towardsdatascience.com/

https://scikit-learn.org