Introduction

Pattern Recognition

Gamaliel Moreno Chávez

MCPI

Enero-Julio
2021

Introduction

IS PATTERN RECOGNITION IMPORTANT?

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes. We will refer to these objects using the generic term patterns

- Machine vision
- Character (letter or number) recognition
- Computer-aided diagnosis
- Speech recognition

Introduction

What is machine learning?

Machine Learning or ML

A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

we treat all unknown quantities (e.g., predictions about the future value of some quantity of interest, such as tomorrow's temperature, or the parameters of some model) as random variables, that are endowed with probability distributions which describe a weighted set of possible values the variable may have.

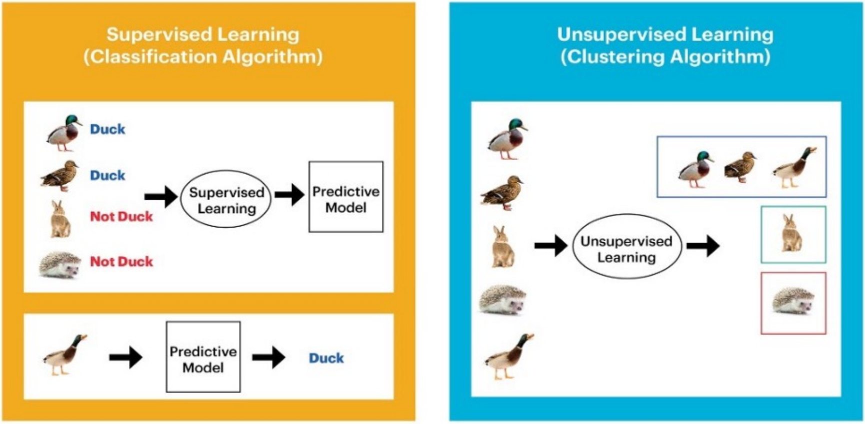
Introduction

Probabilistic modeling is the language used by most other areas of science and engineering

> *Almost all of machine learning can be viewed in probabilistic terms, making probabilistic thinking fundamental. It is, of course, not the only view. But it is through this view that we can connect what we do in machine learning to every other computational science, whether that be in stochastic optimization, control theory, operations research, econometrics, information theory, statistical physics or bio-statistics. For this reason alone, mastery of probabilistic thinking is essential.*

Introduction

Learning types: Supervised and Unsupervised learning



Western Digital

Introduction

Supervised learning

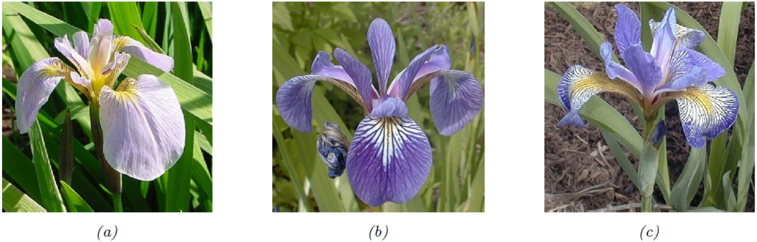
- the task T is to learn a mapping f from inputs $x \in X$ to outputs $y \in Y$.
- The inputs x are also called the features, covariates, or predictors. This is often a fixed dimensional vector of numbers, such as the height and weight of a person, or the pixels in an image.
- $X = \mathbb{R}^D$, where D is the dimensionality of the vector.
- The output y is also known as the label, target, or response.
- The experience E is given in the form of a set of N input-output pairs $\mathcal{D} = \{(xn, yn)\}_{n=1}^{N}$ known as the training set.
- N is called the sample size.

Classification

In classification problems, the output space is a set of C unordered and mutually exclusive labels known as classes, $Y = \{1, 2, \ldots, C\}$. The problem of predicting the class label given an input is also called pattern recognition. If there are just two classes, often denoted by $y \in \{0, 1\}$ or $y \in \{0, 1\}$, it is called binary classification.

Classification

Example: classifying Iris flowers. Consider the problem of classifying iris flowers into their 3 subspecies, setosa, versicolor and virginica.

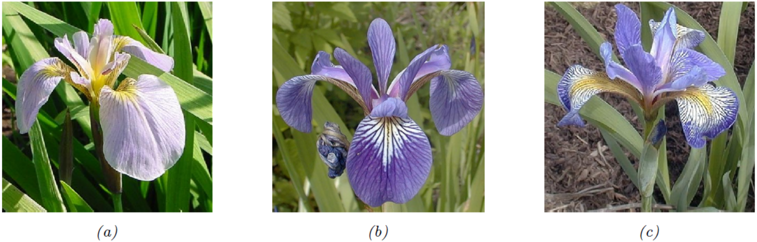


(a) *(b)* *(c)*

Figure 1.1: Three types of iris flowers: setosa, versicolor and virginica. Used with kind permission of Dennis Kramb and SIGNA.

Classification

Example: classifying Iris flowers. Consider the problem of classifying iris flowers into their 3 subspecies, setosa, versicolor and virginica.



(a) (b) (c)

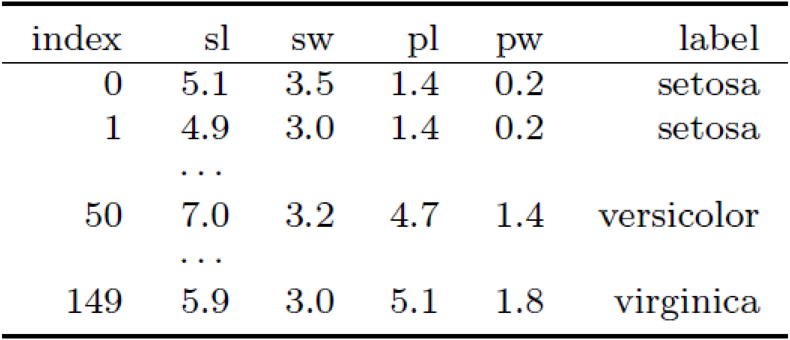
Figure 1.1: Three types of iris flowers: setosa, versicolor and virginica. Used with kind permission of Dennis Kramb and SIGNA.

Classification

In image classification, the input space X is the set of images, which is a very high dimensional space: for a color image with $C = 3$ channels (e.g., RGB) and $D_1 \times D_2$ pixels, we have $X = \mathbb{R}^D$, where $D = C \times D_1 \times D_2$.
Some botanists have already identified 4 simple, but highly informative, numeric feature: sepal length, sepal width, petal length, petal width
We will use this much lower dimensional input space,$X = \mathbb{R}^D$, for simplicity. The iris dataset is a collection of 150 labeled examples of iris flowers, 50 of each type, described by these 4 features.

Classification

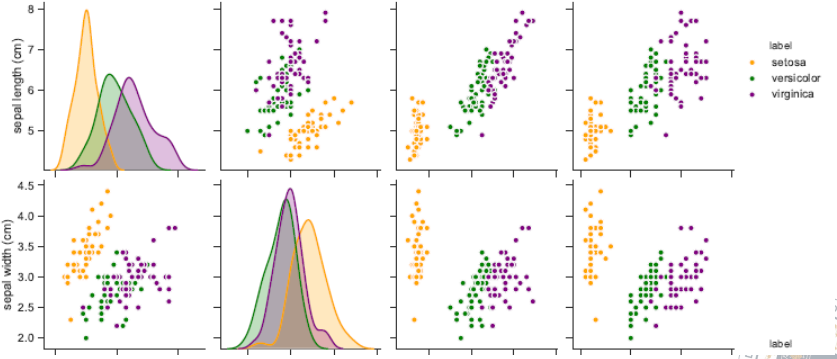
It is common to store them in an $N \times D$ matrix, in which each row represents an example, and each column represents a feature. This is known as a design matrix;



index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
	...				
50	7.0	3.2	4.7	1.4	versicolor
	...				
149	5.9	3.0	5.1	1.8	virginica

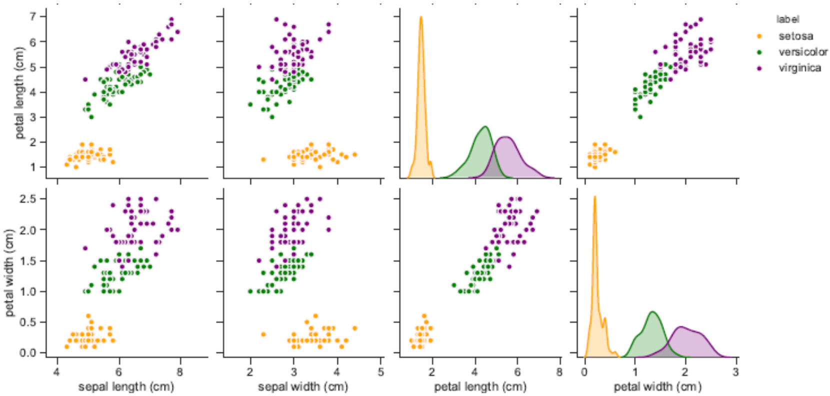
Classification

Exploratory data analysis. Before tackling a problem with ML, it is usually a good idea to perform exploratory data analysis.



Classification

Exploratory data analysis. Before tackling a problem with ML, it is usually a good idea to perform exploratory data analysis.



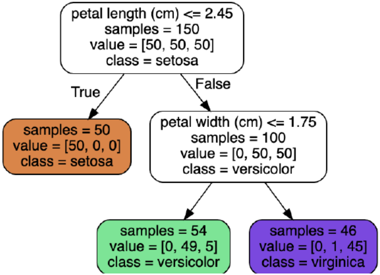
Classification

We can see that the setosa class is easy to distinguish from the other two classes. For example, suppose we create the following decision rule:

$$f(\mathrm{x};\boldsymbol{\theta}) = \begin{cases} \text{Setosa if petal length} < 2.45 \\ \text{Versicolor or Virginica otherwise} \end{cases}$$

Classification

We can arrange these nested rules in to a tree structure, **called a decision tree**.



petal length (cm) <= 2.45
samples = 150
value = [50, 50, 50]
class = setosa

True

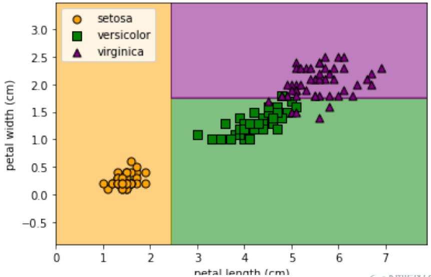
False

samples = 50
value = [50, 0, 0]
class = setosa

petal width (cm) <= 1.75
samples = 100
value = [0, 50, 50]
class = versicolor

samples = 54
value = [0, 49, 5]
class = versicolor

samples = 46
value = [0, 1, 45]
class = virginica



Model fitting

The goal of supervised learning is to automatically come up with classification models. A common way to measure performance on this task is in terms of the misclassification rate on the training set:

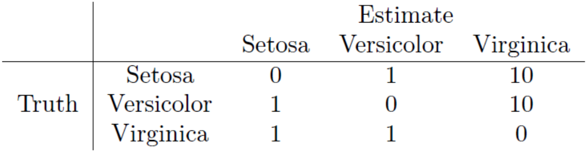
$$\mathcal{L}(\theta) \triangleq \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(y_n \neq f(x_n; \theta))$$

where $\mathbb{I}(e)$ is the binary indicator function,

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

Model fitting

For example, suppose we are foraging in the wilderness and we find some iris flowers. Furthermore, suppose that setosa and versicolor are tasty, but virginica is poisonous. In this case, we might use the asymmetric loss function



		Estimate		
		Setosa	Versicolor	Virginica
Truth	Setosa	0	1	10
	Versicolor	1	0	10
	Virginica	1	1	0

Model fitting

We can then define empirical risk to be the average loss of the predictor on the training set:

$$\mathcal{L}(\theta) \triangleq \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, f(x_n; \theta))$$

the empirical risk is equal to misclassification rate when we use zero-one loss for comparing the true label with the prediction

$$\ell_{01}(y, \hat{y}) = \mathbb{I}(y_n \neq \hat{y})$$

Model fitting

One way to define the problem of model fitting or training is to find a setting of the parameters that minimizes the empirical risk on the training set:

$$\hat{\theta} = \arg \min_{\theta} \mathcal{L}(\theta) = \arg \min_{\theta} \frac{1}{N} \sum_{n=1}^{N} l(y_n, f(x_n; \theta))$$

This is called empirical risk minimization.

Uncertainty

In many cases, we will not be able to perfectly predict the exact output given the input, due to lack of knowledge of the input-output mapping (this is called epistemic uncertainty or model uncertainty), and/or due to intrinisic (irreducible) stochasticity in the mapping (this is called aleatoric uncertainty or data uncertainty).We can capture our uncertainty using the following conditional probability distribution:

$$p(y = c|x; \theta) = f_c(x; \theta)$$

where $f : X \to [0; 1]^C$ maps inputs to a probability distribution over the C possible output labels.

Uncertainty

When fitting probabilistic models, it is common to use the negative log probability as our loss function:

$$\ell(y, f(x;\theta)) = -logp(y|f(x;\theta))$$

The reasons for this are explained after, but the intuition is that a good model (with low loss) is one that assigns a high probability to the true output y for each corresponding input x. The average negative log probability of the training set is given by

$$NNL(\theta) = \frac{1}{N}\sum_{n=1}^{N} logp(y|f(x;\theta))$$

This is called the **negative log likelihood**.

Regression

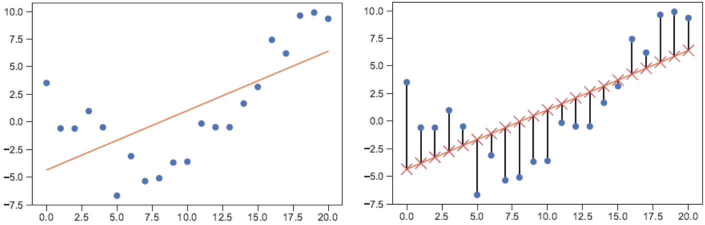
Now suppose that we want to predict a real-valued quantity $y \in \Re$ instead of a class label $y \in \{1, \dots, C\}$ this is known as regression. For example, in the case of Iris flowers, y might be the degree of toxicity if the flower is eaten, or the average height of the plant. Regression is very similar to classification. However, since the output is real-valued, we need to use a different loss function. For regression, the most common choice is to use quadratic loss, or l_2 loss:

$$\ell_2(y, \hat{y}) = (y - \hat{y})^2$$

Linear regression

We can fit this data using a simple linear regression model of the form

$$f(x; \theta) = b + wx$$



where w is the slope, b is the offset, and $\theta = (w, b)$ are all the parameters of the model.

Linear regression

By adjusting, we can minimize the sum of squared errors until we find the least squares solution

$$\hat{\theta} = \arg \underset{\theta}{\text{mín}} \, MSE(\theta)$$

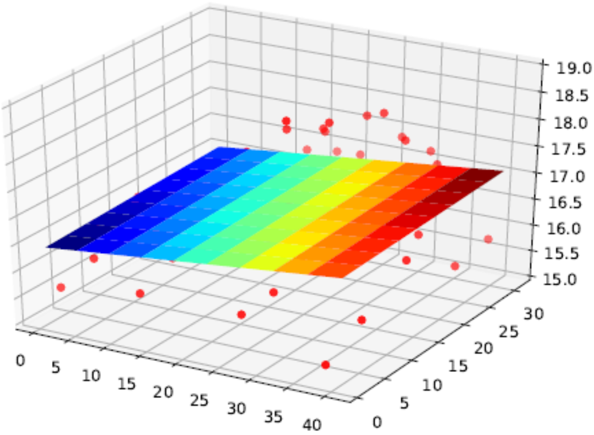
If we have multiple input features, we can write

$$f(x; \theta) = b + w_1 x_1 + \cdots + w_D x_D = b + \boldsymbol{w}^T \boldsymbol{x}$$

where$\boldsymbol{\theta} = (w; b)$. This is called **multiple linear regression**.

Linear regression

The fitted plane has the form $\hat{f}(\boldsymbol{x}) = w_0 + w_1 x_1 + w_2 x_2$.



Linear regression

We can extend this model to use $D > 2$ input features (such as time of day), but then it becomes harder to visualize. To reduce notational clutter, it is standard to absorb the bias term b into the weights $\boldsymbol{w}$ by defining $\tilde{w} = [b, w_1, \dots, wD]$ and defining $\tilde{x} = [1, x_1, \dots, x_D]$, so that

$$\tilde{w}^T \tilde{x} = \boldsymbol{w}^T \boldsymbol{x} + b$$

This converts the affine function into a linear function. In statistics, the $\boldsymbol{w}$ parameters are usually called regression coefficients (and are typically denoted by $\boldsymbol{\beta}$
) and b is called the intercept. In ML, the parameters $\boldsymbol{w}$ are called the **weights** and b is called the **bias**.

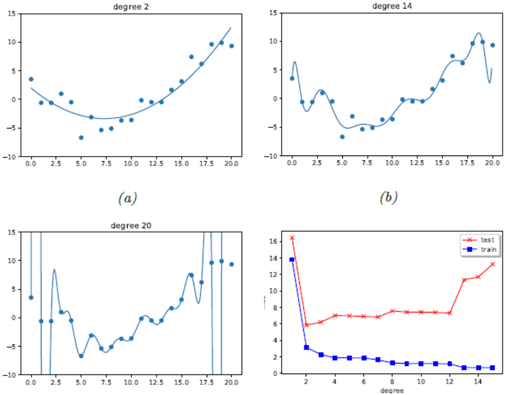
Polynomial regression

We can improve the fit by using a **polynomial regression** model of degree D. This has the form $f(x; \boldsymbol{w}) = \boldsymbol{w}^T \phi(x)$, where $\phi(x)$ is a feature vector derived from the input, which has the following form:

$$\phi(x) = [1, x, x^2, \ldots, x^D]$$

we see that using $D = 2$ results in a much better fit. We can keep increasing D, and hence the number of parameters in the model, until $D = N - 1$; in this case, we have one parameter per data point, so we can perfectly interpolate the data. The resulting model will have 0 MSE. However, intuitively the resulting function will not be a good predictor for future inputs, since it is too "wiggly".

Polynomial regression

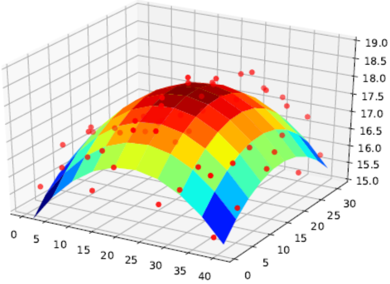


(a)

(b)

Polynomial regression

We can also apply polynomial regression to multi-dimensional inputs.For example, next figure plots the predictions for the temperature model after performing a quadratic expansion of the inputs



We can also add cross terms, such as $x_1 x_2$, to capture interaction effects.

Deep neural networks

We can create much more powerful models by learning to do such nonlinear feature extraction automatically. If we let $\phi(x)$ have its own set of parameters, say $\boldsymbol{V}$, then the overall model has the form

$$f(x; w, \boldsymbol{V}) = w^T \phi(x; \boldsymbol{V})$$

The resulting model then becomes a stack of L nested functions:

$$f(x; \boldsymbol{\theta}) = f_L(f_{L-1}(\cdots(f_1(x))\cdots))$$

where $f_\ell(x) = f(x; \theta_\ell)$ is the function at layer ℓ. The final layer is linear and has the form
$f_L(x) = w^T f_{1:L-1}(x)$, where $f_{1:L-1}(x)$ is the learned feature extractor. This is the key idea behind **deep neural networks** or **DNNs**.

Overfitting and generalization

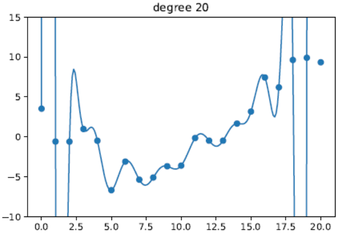
We can rewrite the empirical risk in the following equivalent way:

$$\mathcal{L}(\theta; \mathcal{D}_{train}) = \frac{1}{|\mathcal{D}_{train}|} \sum_{(x_n, y_n) \in \mathcal{D}_{train}} \ell(y_n, f(x_n); \boldsymbol{\theta})$$

where $N = |Dtrain|$ is the size of the training set $\mathcal{D}_{train}$. This formulation is useful because it makes explicit which dataset the loss is being evaluated on.

A model that perfectly fits the training data, but which is too complex, is said to suffer from **overfitting**.

Overfitting and generalization



To detect if a model is overfitting, let us assume (for now) that we have access to the true (but unknown) distribution $p*(x;y)$ used to generate the training set. Then, instead of computing the empirical risk we compute the theoretical expected loss or **population risk**

$$\mathcal{L}(\boldsymbol{\theta}; P) \triangleq \mathbf{E}_{p*(x,y)}[\ell(y, f(x;\boldsymbol{\theta}))] \tag{1}$$

Overfitting and generalization

In practice we don't know $p*$. However, we can partition the data we do have into two subsets, known as the training set and the **test set**. Then we can approximate the population risk using the **test risk**

$$\mathcal{L}(\theta; \mathcal{D}_{test}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{(x_n, y_n) \in \mathcal{D}_{test}} \ell(y_n, f(x_n))$$

In practice, we need to partition the data into three sets, namely the training set, the test set and a validation set; the latter is used for model selection, and we just use the test set to estimate future performance (the population risk), i.e., the test set is not used for model fitting or model selection.

No free lunch theorem

No free lunch theorem

There is no single best model that works optimally for all kinds of problems — this is sometimes called the no free lunch theorem [1].

For this reason, it is important to have many models and algorithmic techniques in one's toolbox to choose from.

[1] D. Wolpert. "The lack of a priori distinctions between learning algorithms". In: Neural Computation 8.7 (1996), pp. 1341–1390 (page 12).

Unsupervised learning

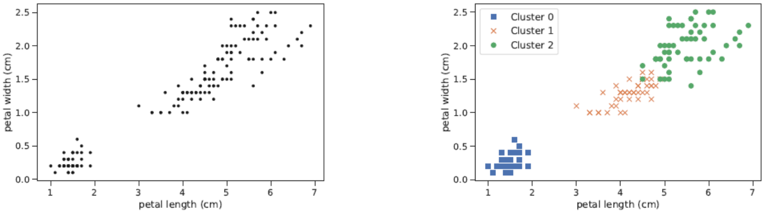
An arguably much more interesting task is to try to “make sense of” data, as opposed to just learning a mapping. That is, we just get observed “inputs” $\mathcal{D} = \{x_n : n = 1 : N\}$ without any corresponding “outputs” y_n. This is called **unsupervised learning**.
From a probabilistic perspective, we can view the task of unsupervised learning as fitting an unconditional model of the form $p(x)$.

Unsupervised learning

- Unsupervised learning avoids the need to collect large labeled datasets for training
- Unsupervised learning also avoids the need to learn how to partition the world into often arbitrary categories.
- Finally, unsupervised learning forces the model to “explain” the high dimensional inputs

Clustering

A simple example of unsupervised learning is the problem of finding clusters in data. The goal is to partition the input into regions that contain “similar” points.



Self-supervised learning

A recently popular approach to unsupervised learning is known as self-supervised learning. In this approach, we create proxy supervised tasks from unlabeled data.For example, we might try to learn to predict a color image from a gray scale image, or to mask out words in a sentence and then try to predict them given the surrounding context. The hope is that the resulting predictor $\hat{x}_1 = f(x_2; \theta)$, where x_2 is the observed input and x_1 is the predicted output, will learn useful features from the data, that can then be used in standard, downstream supervised tasks.

Evaluating unsupervised learning

it is very hard to evaluate the quality of the output of an unsupervised learning method, because there is no ground truth to compare. A common method for evaluating unsupervised models is to measure the probability assigned by the model to unseen test examples. We can do this by computing the (unconditional) negative log likelihood of the data:

$$\mathcal{L}(\theta;\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{(x\in\mathcal{D})} log\ p(x|\theta)$$

This treats the problem of unsupervised learning as one of density estimation.

Reinforcement learning

addition to supervised and unsupervised learning, there is a third kind of ML known as **reinforcement learning** (RL). In this class of problems, the system or agent has to learn how to interact with its environment. This can be encoded by means of a policy $a = \pi(x)$, which specifies which action to take in response to each possible input x (derived from the environment state).

Definición

El **puntaje z muestral** es una medida de posición relativa definida por

$$\text{puntaje } z = \frac{x - \bar{x}}{s}$$

Un puntaje z mide la distancia entre una observación y la media, medidas en unidades de desviación estándar.

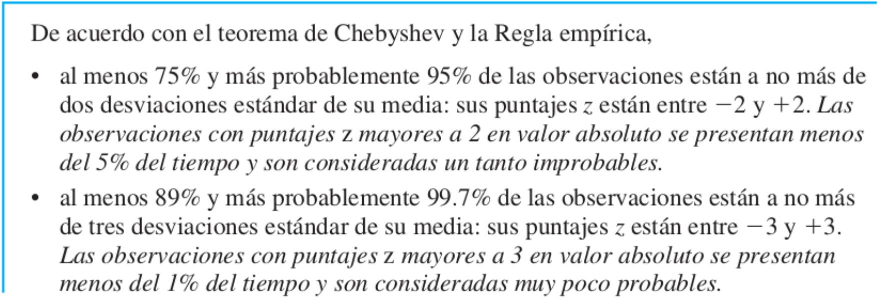
Mediciones de posición relativa

Por ejemplo, suponga que la media y desviación estándar de los puntajes de examen (basados en un total de 35 puntos) son 25 y 4, respectivamente. El puntaje z para una calificación de 30 se calcula como sigue:

$$\text{puntaje } z = \frac{x - \bar{x}}{s} = \frac{30 - 25}{4} = 1.25$$

El puntaje de 30 está a 1.25 desviaciones estándar arriba de la media ($30 = \bar{x} + 1.25s$).

Mediciones de posición relativa



De acuerdo con el teorema de Chebyshev y la Regla empírica,

- al menos 75% y más probablemente 95% de las observaciones están a no más de dos desviaciones estándar de su media: sus puntajes z están entre -2 y $+2$. *Las observaciones con puntajes z mayores a 2 en valor absoluto se presentan menos del 5% del tiempo y son consideradas un tanto improbables.*
- al menos 89% y más probablemente 99.7% de las observaciones están a no más de tres desviaciones estándar de su media: sus puntajes z están entre -3 y $+3$. *Las observaciones con puntajes z mayores a 3 en valor absoluto se presentan menos del 1% del tiempo y son consideradas muy poco probables.*

Percentil

Un percentil es otra medida de posición relativa y se usa con más frecuencia para conjuntos grandes de datos. (Los percentiles no son muy útiles para conjuntos pequeños de datos).

Definición

Un conjunto de n mediciones de la variable x se ha reacomodado en orden de magnitud. El p-ésimo percentil es el valor de x que es mayor a $p\,\%$ de las mediciones y es menor que el restante $(100 - p)\,\%$.

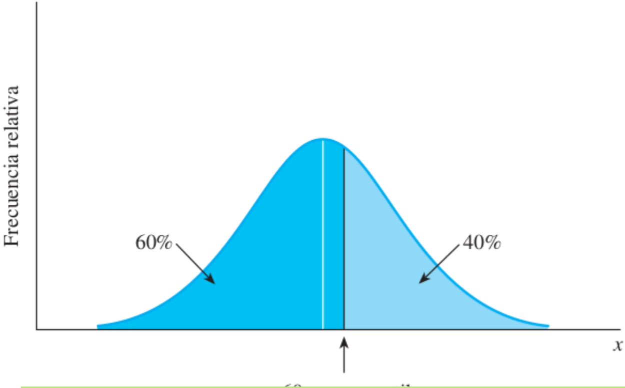
Percentil

Supongamos que usted ha sido notificado que su calificación de 610, en el Examen verbal de graduación, lo ha colocado en el 60avo percentil en la distribución de calificaciones. ¿Dónde está su calificación de 610 en relación a las calificaciones de los otros que tomaron el examen?

Solución Calificar en el 60avo percentil significa que 60 % de todas las calificaciones de examen fueron más bajas que la calificación de usted y 40 % fueron más altas.

Percentil

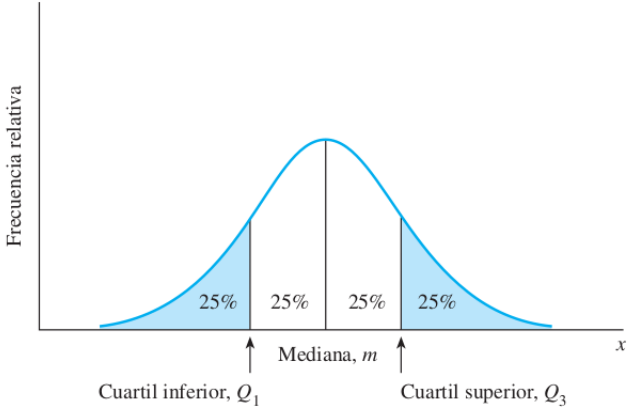
En general, el 60avo percentil para la variable x es un punto en el eje horizontal de la distribución de datos que es mayor a 60 % de las mediciones y menor que las otras.



Percentil-Cuartil

- La mediana es igual que el 50avo percentil.
- Los percentiles 25avo y 75avo, llamados cuartiles inferior y superior
- Veinticinco por ciento de las mediciones serán menores que el cuartil inferior (primero).
- 50 % serán menores que la mediana (el segundo cuartil).
- 75 % serán menores que el cuartil superior (tercero).

Cuartil

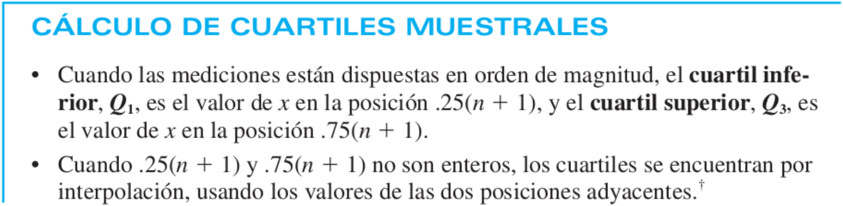

Frecuencia relativa
25%
25%
25%
25%
x
Mediana, m
Cuartil inferior, Q_1
Cuartil superior, Q_3

Cuartil

Definición

Un conjunto de n mediciones en la variable x se ha acomodado en orden de magnitud. El **cuartil inferior (primer cuartil),** Q_1 , es el valor de x que es mayor a un cuarto de las mediciones y es menor que los restantes tres cuartos. El **segundo cuartil** es la mediana. El **cuartil superior (tercer cuartil),** Q_3 , es el valor de x que es mayor a tres cuartos de las mediciones y es menor que el restante un cuarto.

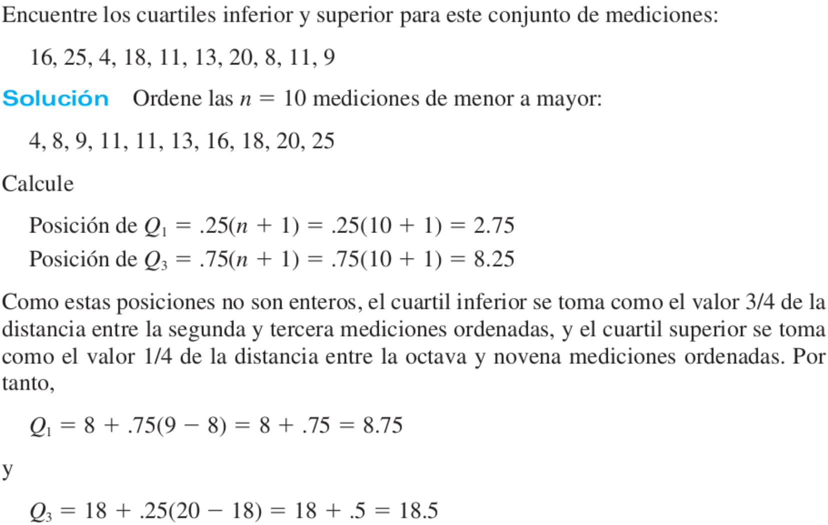
Cuartil



CÁLCULO DE CUARTILES MUESTRALES

- Cuando las mediciones están dispuestas en orden de magnitud, el **cuartil inferior**, $\boldsymbol{Q_1}$, es el valor de x en la posición $.25(n + 1)$, y el **cuartil superior**, $\boldsymbol{Q_3}$, es el valor de x en la posición $.75(n + 1)$.
- Cuando $.25(n + 1)$ y $.75(n + 1)$ no son enteros, los cuartiles se encuentran por interpolación, usando los valores de las dos posiciones adyacentes.†

Cuartil



Encuentre los cuartiles inferior y superior para este conjunto de mediciones:

16, 25, 4, 18, 11, 13, 20, 8, 11, 9

Solución Ordene las $n = 10$ mediciones de menor a mayor:

4, 8, 9, 11, 11, 13, 16, 18, 20, 25

Calcule

$$\text{Posición de } Q_1 = .25(n + 1) = .25(10 + 1) = 2.75$$
$$\text{Posición de } Q_3 = .75(n + 1) = .75(10 + 1) = 8.25$$

Como estas posiciones no son enteros, el cuartil inferior se toma como el valor 3/4 de la distancia entre la segunda y tercera mediciones ordenadas, y el cuartil superior se toma como el valor 1/4 de la distancia entre la octava y novena mediciones ordenadas. Por tanto,

$$Q_1 = 8 + .75(9 - 8) = 8 + .75 = 8.75$$

y

$$Q_3 = 18 + .25(20 - 18) = 18 + .5 = 18.5$$

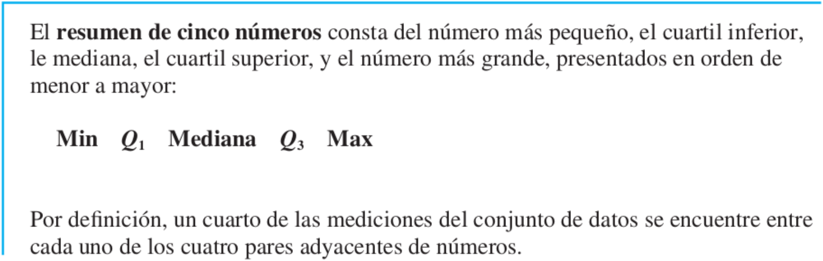
Intercuartil

Como la mediana y los cuartiles dividen la distribución de datos en cuatro partes, cada una de ellas conteniendo alrededor de 25 % de las mediciones, Q_1 y Q_3 son las fronteras superior e inferior para el 50 % central de la distribución. Podemos medir el rango de este “50 % central” de la distribución usando una medida numérica llamada rango intercuartil.

Definición

El rango intercuartil (IQR) para un conjunto de mediciones es la diferencia entre los cuartiles superior e inferior; esto es,
$IQR = Q_3 - Q_1$.

El resumen de cinco números y la gráfica de caja



El **resumen de cinco números** consta del número más pequeño, el cuartil inferior, le mediana, el cuartil superior, y el número más grande, presentados en orden de menor a mayor:

Min $\quad Q_1 \quad$ **Mediana** $\quad Q_3 \quad$ **Max**

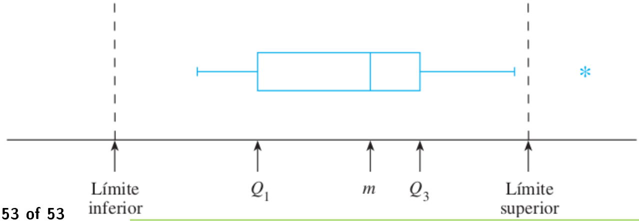
Por definición, un cuarto de las mediciones del conjunto de datos se encuentre entre cada uno de los cuatro pares adyacentes de números.

El resumen de cinco números y la gráfica de caja

PARA CONSTRUIR UNA GRÁFICA DE CAJA

- Calcule la mediana, los cuartiles superior e inferior y el IQR para el conjunto de datos.
- Trace una recta horizontal que represente la escala de medición. Forme una caja un poco arriba de la recta horizontal con los extremos derecho e izquierdo en Q_1 y Q_3. Trace una recta vertical que pase por la caja en la ubicación de la mediana.

Una gráfica de caja se muestra en la figura 2.17.


*
Límite inferior
Q_1
m
Q_3
Límite superior