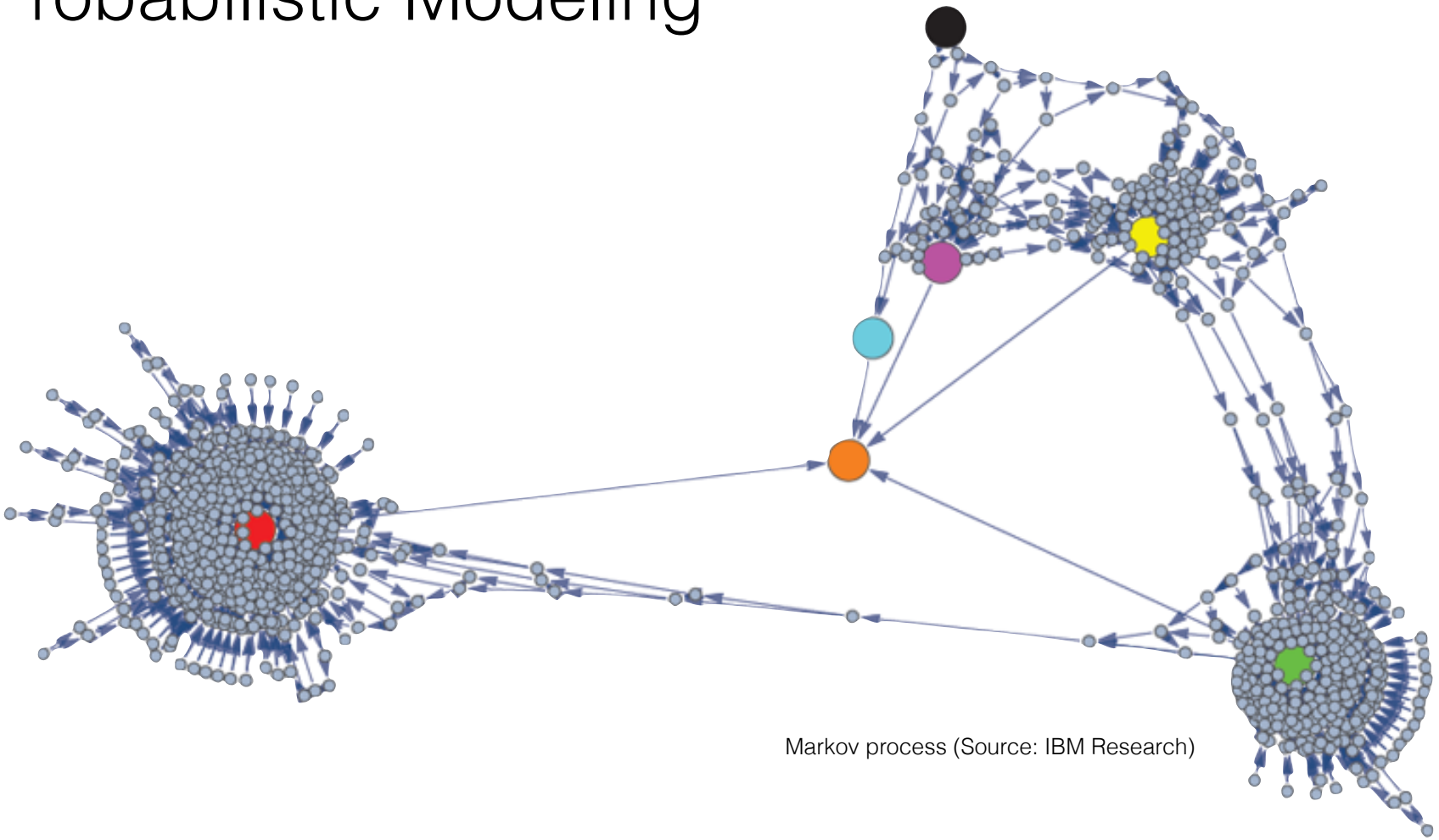
SECTION 2

Cyber Security Analytics:
Probabilistic Modeling



Markov process (Source: IBM Research)

Probability Theory

- the branch of mathematics concerned with probability
- there are several different **probability interpretations**
- axioms formalize probability in terms of a probability space[1], which
  - assigns a probability measure between 0 and 1
  - to a set of POSSIBLE OUTCOMES—or *sample points*—, called the sample space Ω
  - any subset of these outcomes is called an EVENT
- mathematical abstractions of uncertain (*stochastic*) processes:
  - discrete and continuous random variables
  - probability distributions P
- describes random events using the **law of large numbers** and the **central limit theorem**

Conceptual experiments: (1) Coin tossing, (2) Rolling dice, (3) Roulette wheel

As a mathematical foundation for statistics, probability theory is essential to quantitative analysis of data.

---

[1] Andrey Kolmogorov introduced the notion of probability space, together with other axioms of probability, in the 1930s.

Axioms of Probability

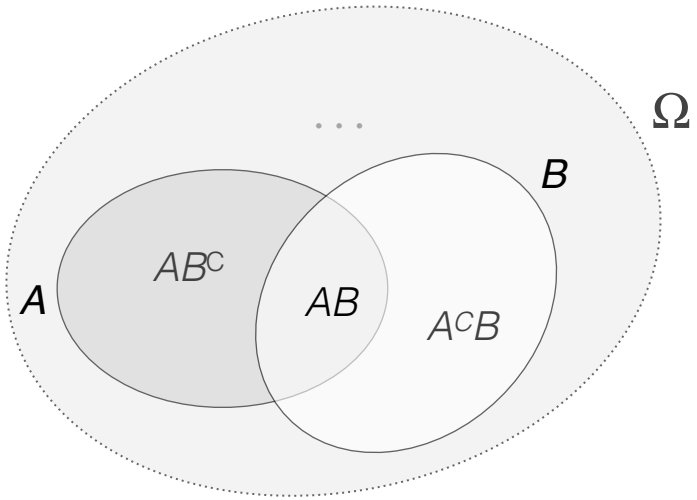
The axioms of probability are mathematical rules that **probability must satisfy**.

Let A and B be events and $P(A)$ denote the probability of the event A. The axioms of probability are these three conditions on the function P:

- For every event A, $P(A) \geq 0$.
- The probability of the entire sample space Ω is 100%.
- If two events A and B are disjoint, $P(A \cup B) = P(A) + P(B)$, the probability that either of the events happens is the sum of the probabilities that each happens.

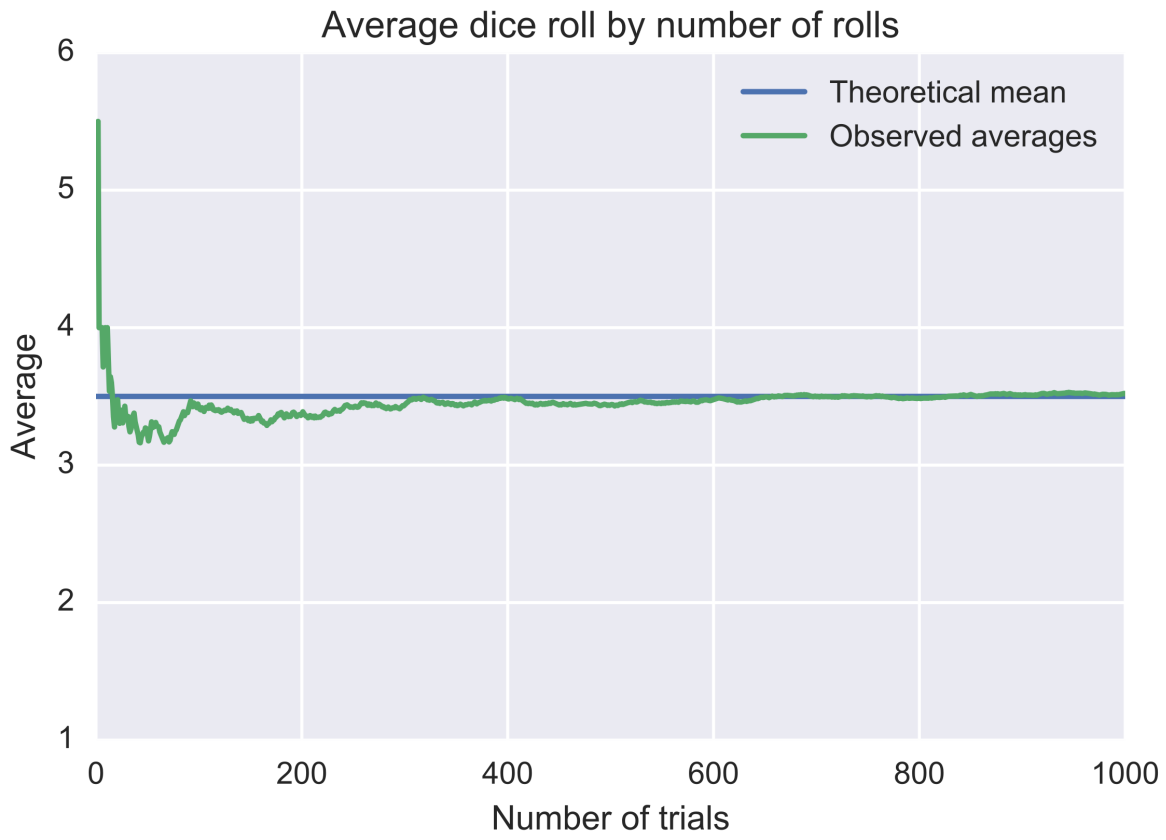
  If two events A and B are not disjoint, $P(A \cup B) = P(A) + P(B) - P(AB)$.

Note: $P(A) = P(AB^C \cup AB) = P(AB^C) + P(AB)$,
$P(B) = P(A^CB \cup AB) = P(A^CB) + P(AB)$



Law of Large Numbers

- describes the result of performing the same experiment a large number of times
- the average of the results obtained from a large number of trials should be close to the expected value, and tends to become closer as more trials are performed
- it guarantees **stable long-term results** for the averages of some random events



Central Limit Theorem

The central limit theorem states that

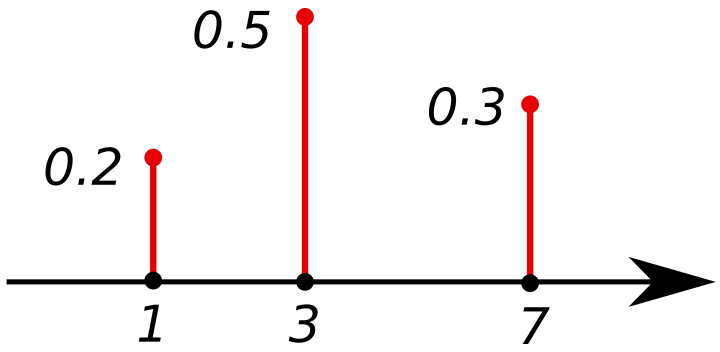
- if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement, the distribution of **the sample means** will be approx. normally distributed.

- This will hold true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually $n > 30$).

This means that we can use the **normal probability model** to quantify uncertainty when making inferences about a population mean based on the sample mean.

Probability Mass Function

a function that gives the probability that a **discrete random variable** is exactly equal to some value



- Probability mass function (PMF) of a discrete probability distribution defining the probability of observing “1, 3, or 7” is 0.2, 0.5, 0.3, respectively.

- All the values must be non-negative and sum up to 1.

Expected Value

Let X be a random variable with a finite number of possible outcomes $x_1, x_2, \ldots, x_k$ occurring with probabilities $p_1, p_2, \ldots, p_k$, respectively.

The expected value (a.k.a. average, expectation, mean value or **mean**) of X is defined as the probability-weighted average of X:

$$E[X] = \sum_{i=1}^{k} x_i p_i = x_1 p_1 + x_2 p_2 + \ldots + x_k p_k .$$

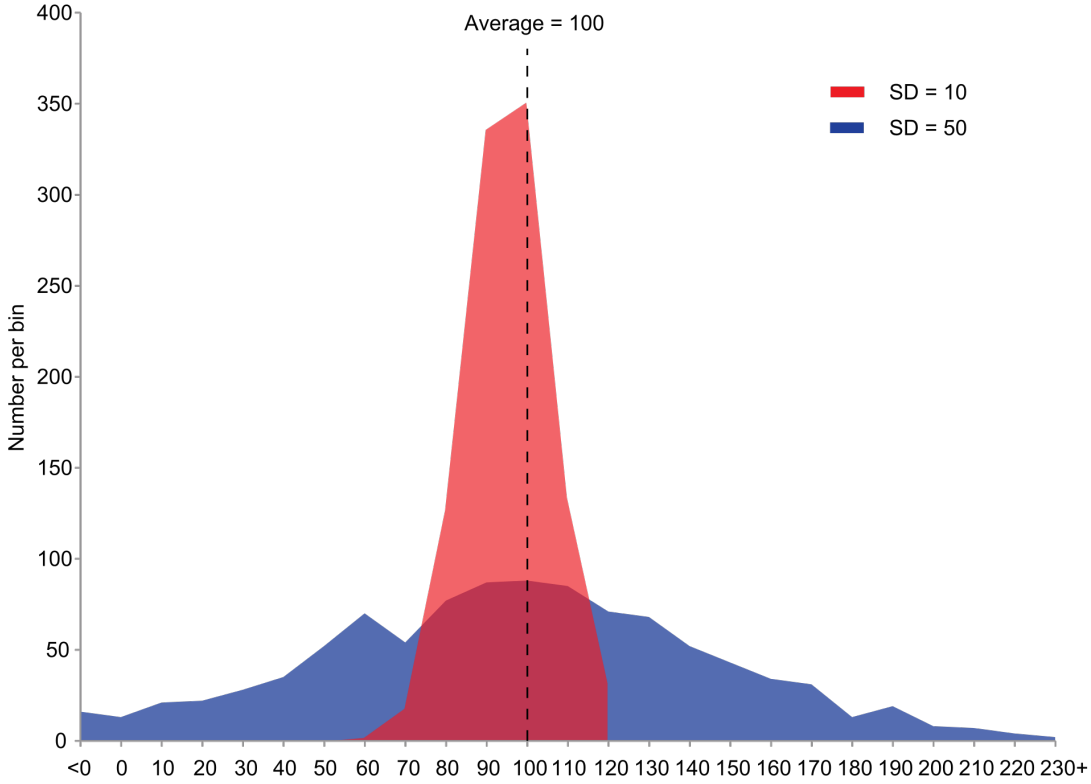
Variance

The variance of a random variable X is the **expected value** of the squared deviation from the mean of X, $\mu = E[X]$:

$$\text{Var}(X) = E[(X - \mu)^2]$$

Variance is typically also referred to as σ^2 (σ designates the **standard deviation**).

Two samples with the same mean and different variances. Red sample: mean 100 and variance 100; blue sample has mean 100 and variance 2,500. Each sample has 1000 values drawn at random from a gaussian distribution with the specified parameters. (Source: Wikipedia)



Standard Deviation

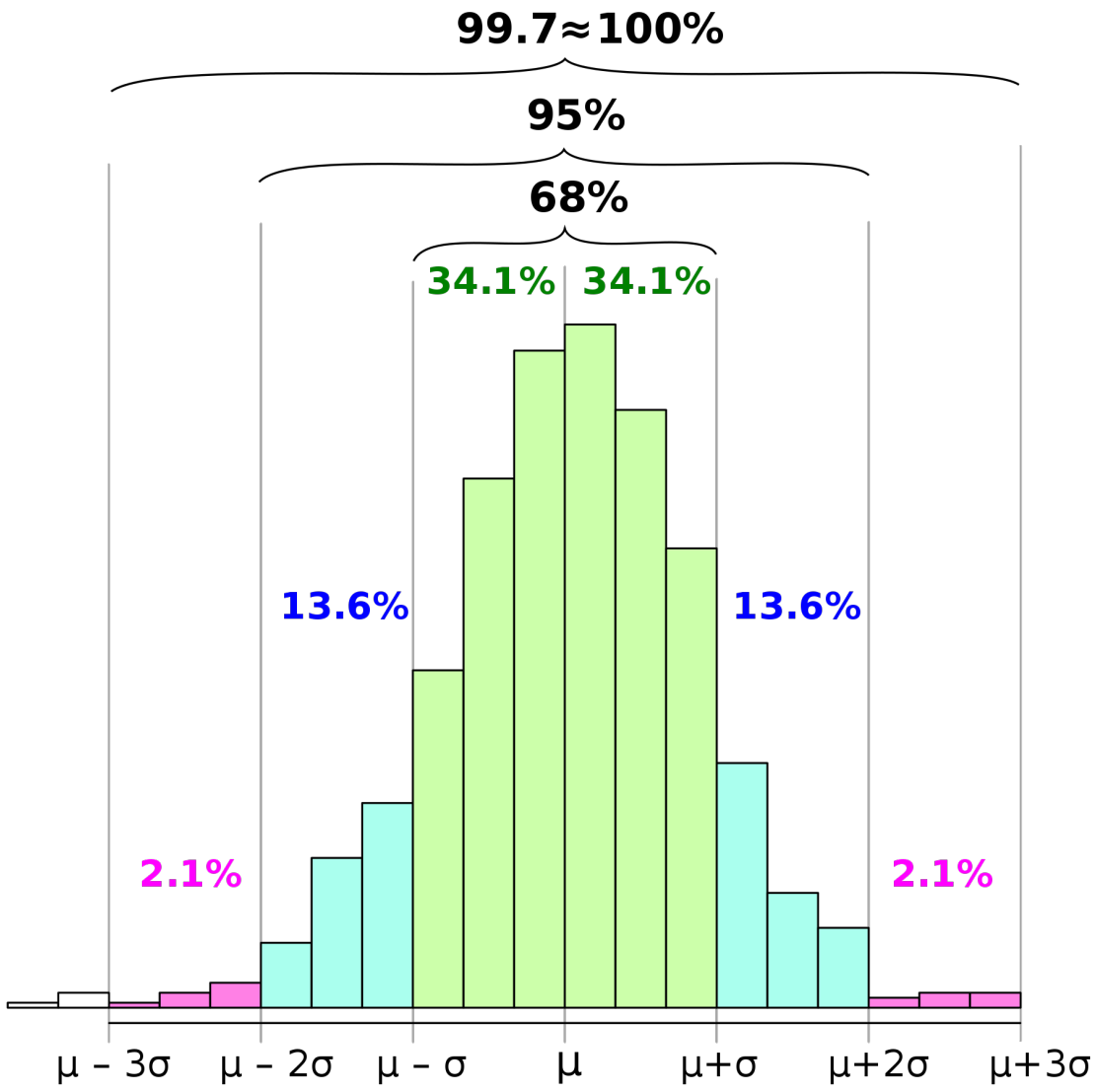
Let X be a random variable with **mean value** μ: $E[X] = \mu$.

Then the standard deviation of X is the quantity

$$\begin{aligned}
\sigma &= \sqrt{\mathrm{E}[(X-\mu)^2]} \\
&= \sqrt{\mathrm{E}[X^2] + \mathrm{E}[-2\mu X] + \mathrm{E}[\mu^2]} \\
&= \sqrt{\mathrm{E}[X^2] - 2\mu\,\mathrm{E}[X] + \mu^2} \\
&= \sqrt{\mathrm{E}[X^2] - 2\mu^2 + \mu^2} \\
&= \sqrt{\mathrm{E}[X^2] - \mu^2}
\end{aligned}$$

σ is the square root of the **variance** of X; i.e.,

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \quad \text{where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i .$$



Conditional Probability

William Feller. An Introduction to Probability Theory and Its Applications (Third Edition). John Wiley & Sons,1968.

Conditional probability is a measure of the probability of an event occurring given that another event has occurred. If the event of interest is *A* and the event *B* is known or assumed to have occurred, the conditional probability of *A* given *B*—the probability of *A* under the condition *B*—is usually written as $P(A \mid B)$, or sometimes $P_B(A)$.

Example: the probability that any given person has a cough on any given day may be only 5%. But if we know or assume that the person has a cold, then they are much more likely to be coughing. The conditional probability of coughing given that you have a cold might thus be a much higher than 5%.

For $P(B) > 0$, we have

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

the probability that both events A and B occur

Bayesian Probability

- also known as *epistemic probability* or **subjectivism**[2],
  views 'probability' as a subjective measure of the degree of belief of the INDIVIDUAL assessing the UNCERTAINTY of a particular situation

- probability is interpreted as **reasonable expectation** representing a state of knowledge or as quantification of a personal belief

- can be seen as an extension of propositional logic that enables reasoning with hypotheses: propositions whose truth or falsity is uncertain

- to evaluate the probability of a hypothesis, one specifies a prior probability which, in turn, is then updated to a posterior probability in the light of new evidence (relevant data) using procedures and formulae to perform this calculation

The term Bayesian derives from the 18th century mathematician and theologian **Thomas Bayes**, who provided the first mathematical treatment of a non-trivial problem of statistical data analysis using what is now known as **Bayesian inference**.

---

[2] What is the meaning of "probability" depending on the application context? There are two broad categories of probability interpretations which can be called **physical probabilities** and **evidential probabilities** (*subjectivism*).

Bayes' Theorem

In probability theory and statistics, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event (a.k.a. prior). For example, if the chance of having a disease is related to age, then, using Bayes' theorem, a person's age can be used to more accurately assess the probability that they have the disease, compared to the assessment of the probability of the disease made without knowledge of the person's age.

Bayes' theorem is stated mathematically as the following equation:

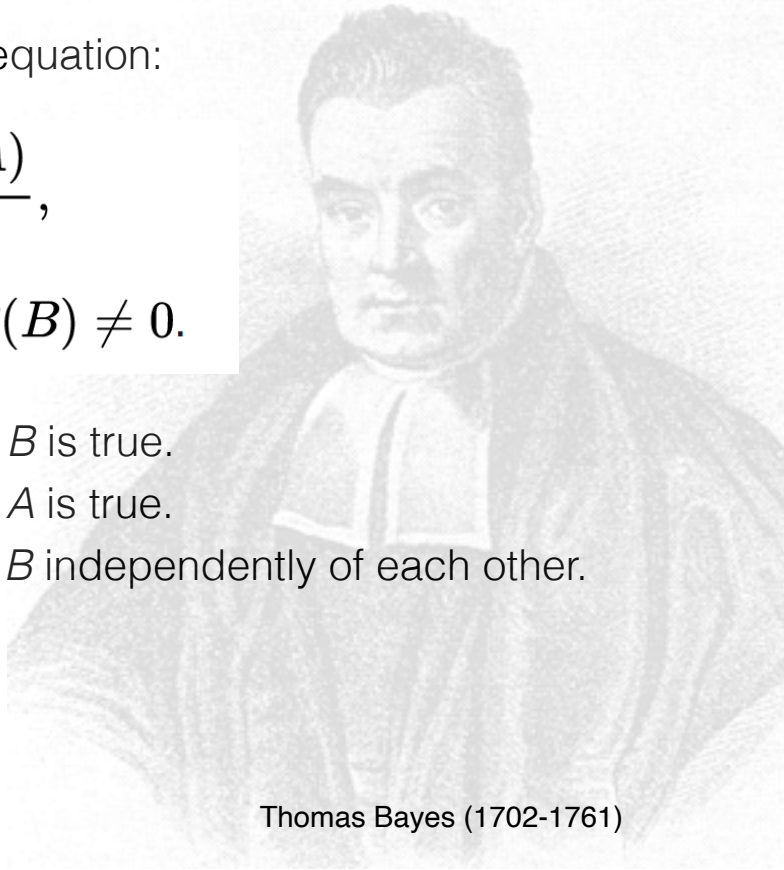
$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

where A and B are events and $P(B) \neq 0$.

- $P(A \mid B)$ is the likelihood of event *A* occurring given that *B* is true.
- $P(B \mid A)$ is the likelihood of event *B* occurring given that *A* is true.
- $P(A)$ and $P(B)$ are the probabilities of observing *A* and *B* independently of each other.

The Bayesian Trap

https://www.youtube.com/watch?v=R13BD8qKeTg



Thomas Bayes (1702-1761)

Bayesian interpretation

The interpretation of Bayes' theorem depends on "the meaning of probabilities" as associated with the terms $P(A \mid B)$, $P(A)$, $P(B)$.

In the Bayesian interpretation, probability measures a "degree of belief." Bayes' theorem then links the degree of belief in a proposition before and after accounting for evidence.

Example: suppose it is believed with 50% certainty that a coin is **twice as likely** to land heads than tails. If the coin is flipped a number of times and the outcomes observed, that degree of belief may rise, fall or remain the same depending on the results.

For proposition *A* and evidence *B*,

- $P(A)$, the prior, is the initial degree of belief in *A*.
- $P(A \mid B)$, the posterior, is the degree of belief having accounted for *B*.

$\frac{P(B \mid A)}{P(B)}$ represents the support *B* provides for *A*.

Statistical Learning

- Statistical learning focuses on supervised and unsupervised **modeling and prediction**.
- Progress in statistical learning has been marked by the increasing availability of powerful and "relatively user-friendly" software, such as the popular and freely available R system.

Prediction Models[3]

Input variables are called *independent variables, features, or predictors, while the* output variable is called *dependent variable* or *response.*

Suppose we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$, for a dependent variable Y. We assume that there is **some relationship** between X and Y.

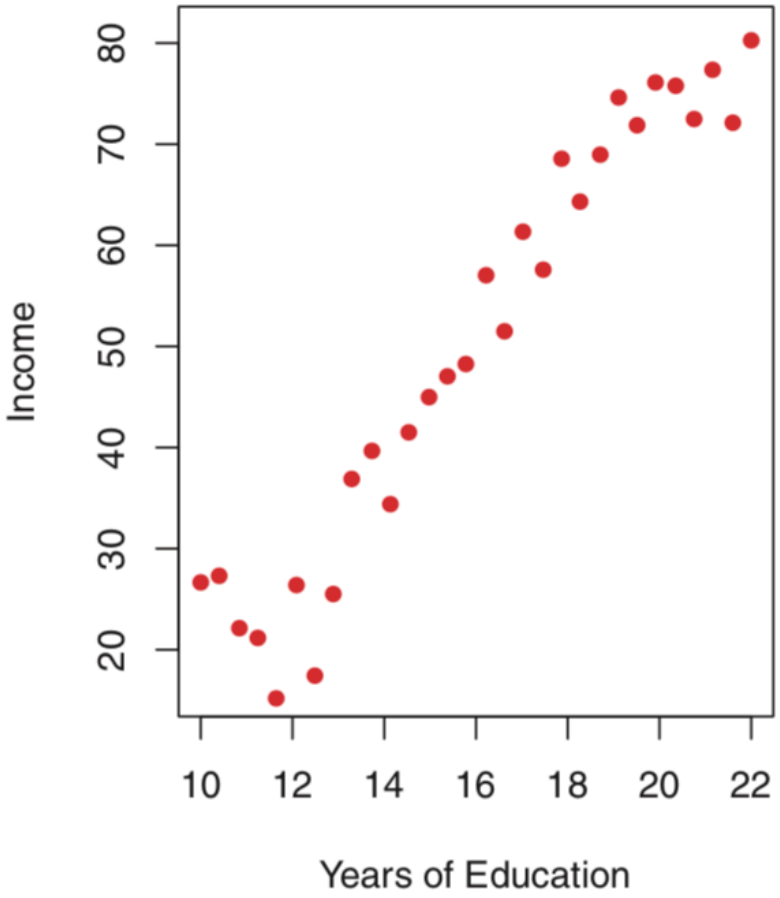
Generally, the relationship between Y and $X = (X_1, X_2, \ldots, X_p)$ can be written as

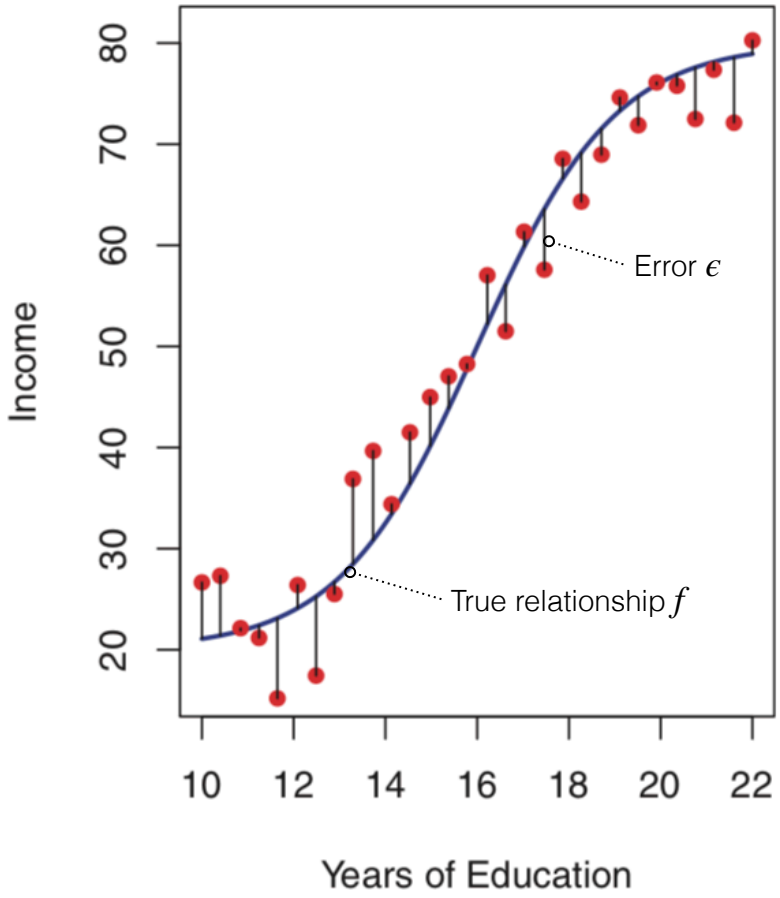
$$Y = f(X) + \epsilon$$

f is some function of X, and ϵ is a random error term,
ϵ is independent of X and has mean zero.

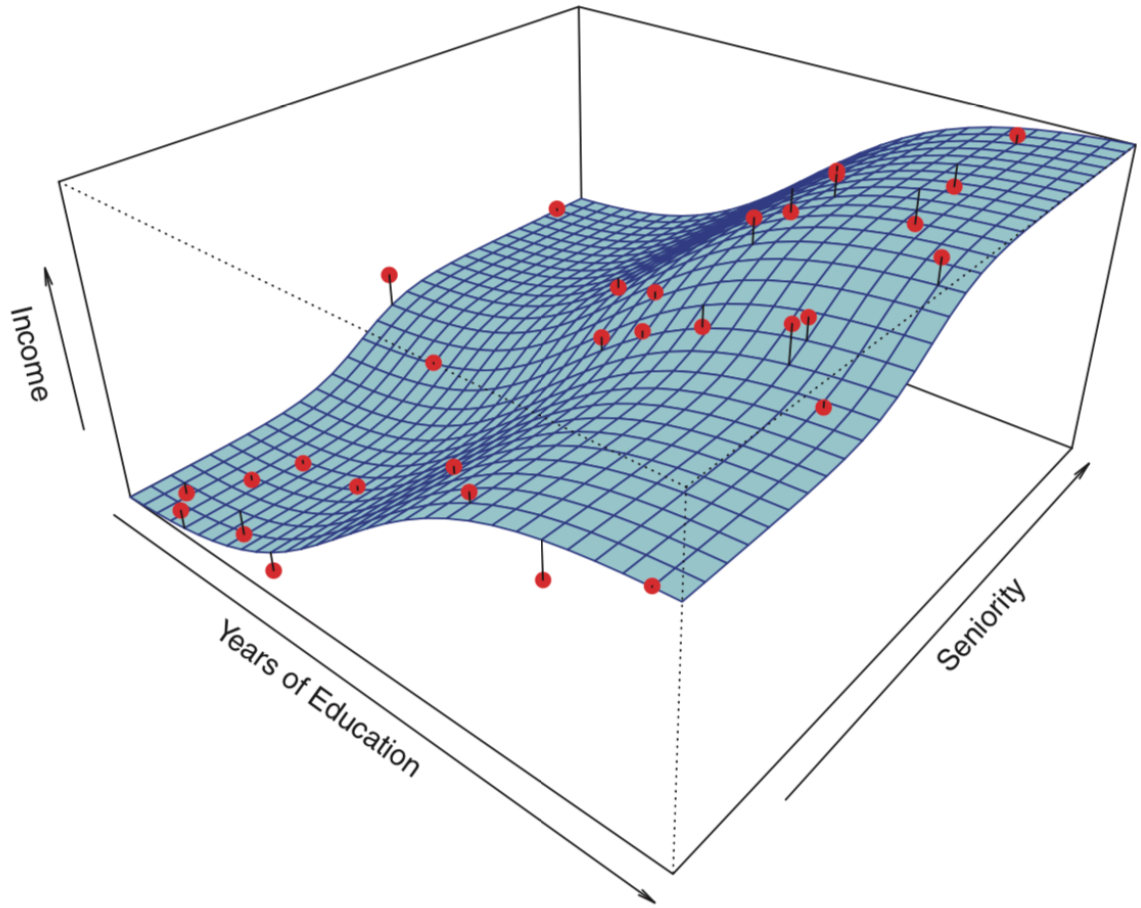
[3] G. James, D. Witten, T. Hastie, and R. Tibshirani. An Introduction to Statistical Learning: with Applications in R, Springer, 2017.

Example

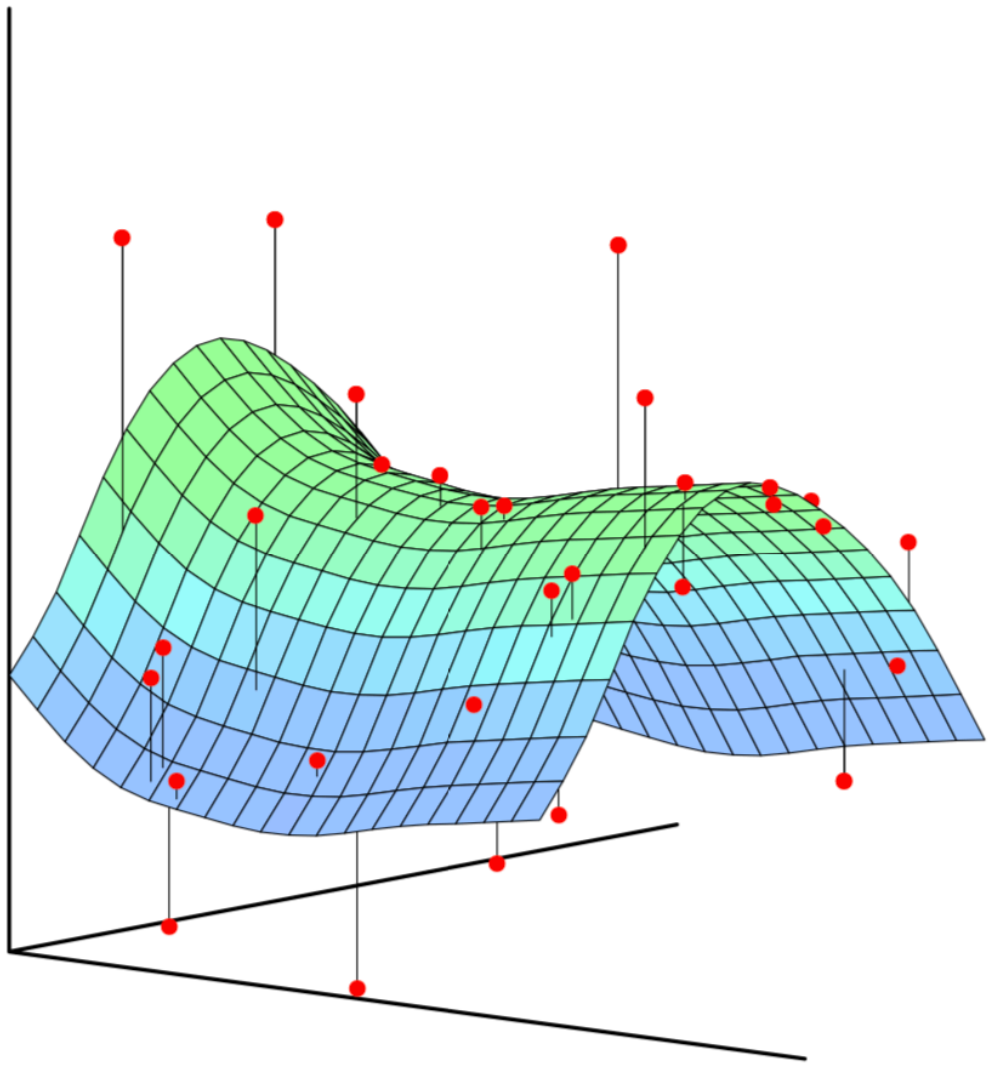




In general, f may involve more than one input variable. The following plot[4] displays **income** as a function of **years of education** and **seniority**, where the blue surface represents the true underlying relationship between income and years of education and seniority.



[4] Source: James et al., 2017.



The function f that connects the input variable to the output variable is in general unknown; thus, one can only **estimate** f based on the **observed points**.

> Statistical learning refers to a set of approaches for estimating f.

- We will explore some key theoretical concepts for estimating f as well as tools for evaluating the estimates obtained.
- We will however do all this with an applied orientation.

Why Estimate f ?

Two main reasons why we may wish to estimate f: ***prediction*** and ***inference***

Prediction

In many situations, a set of inputs X are readily available, but the output Y cannot be easily obtained. Since the error term averages to zero, we can predict Y using

$$\hat{Y} = \hat{f}(X).$$

$\hat{f}$ represents our **estimate**[5] for f and $\hat{Y}$ represents our **prediction** for Y.

Example: suppose $X_1, X_2, \ldots, X_p$ are characteristics of a patient's blood sample that can be easily measured in a lab, and Y is a variable encoding the patient's risk for a severe adverse reaction to a particular drug.

Predicting Y using X may avoid giving the drug to patients who are at high risk of an adverse reaction, i.e. the estimate of Y is high.

[5] We may view $\hat{f}$ as a **black box**, not being concerned with the exact form of $\hat{f}$, as long as it yields accurate predictions for Y.

Accuracy

The accuracy of $\hat{Y}$ as a prediction for Y depends on two quantities, the reducible error and the irreducible error.

- In general, $\hat{f}$ will **not be a perfect estimate** for f.
- This inaccuracy is potentially reducible by using a more appropriate statistical learning technique to estimate f.
- But even with a perfect estimate for f, meaning $\hat{Y} = f(X)$, our prediction would still have some error in it because Y is also a function of ϵ.
- Since ϵ cannot be predicted using X, we cannot reduce the error introduced by ϵ; this irreducible error is inescapable.
- Example: the risk of an adverse reaction **might vary** for a given patient on a given day, depending on manufacturing variation in the drug itself or the patient's general feeling of well-being on that day.

- **Our focus here** is on techniques for estimating f with the objective of minimizing the reducible error. Keep in mind that the irreducible error (typically unknown in practice) will always provide an upper bound on the accuracy of our prediction for Y.

Inference

Understanding the relationship between X and Y, or more specifically, how Y changes as a function of $X_1, X_2, \ldots, X_p$, means finding answers to the following questions[6]:

- Which predictors are associated with the response?
  - only a small fraction of the available predictors may be **substantially associated** with Y
  - identifying the important predictors among a large set of possible variables can be extremely useful

- What is the relationship between the response and each predictor?
  - some predictors may have a positive relationship with Y
  - other predictors may have the opposite relationship

- Can the relationship between Y and each predictor be adequately summarized using a **linear equation**, or is the relationship more complicated?
  - A linear model may be more desirable but not accurately describe the relationship between the input and output variables.

---

[6] **Note**: the goal here is to find an estimation for f, but not necessarily to predict Y.

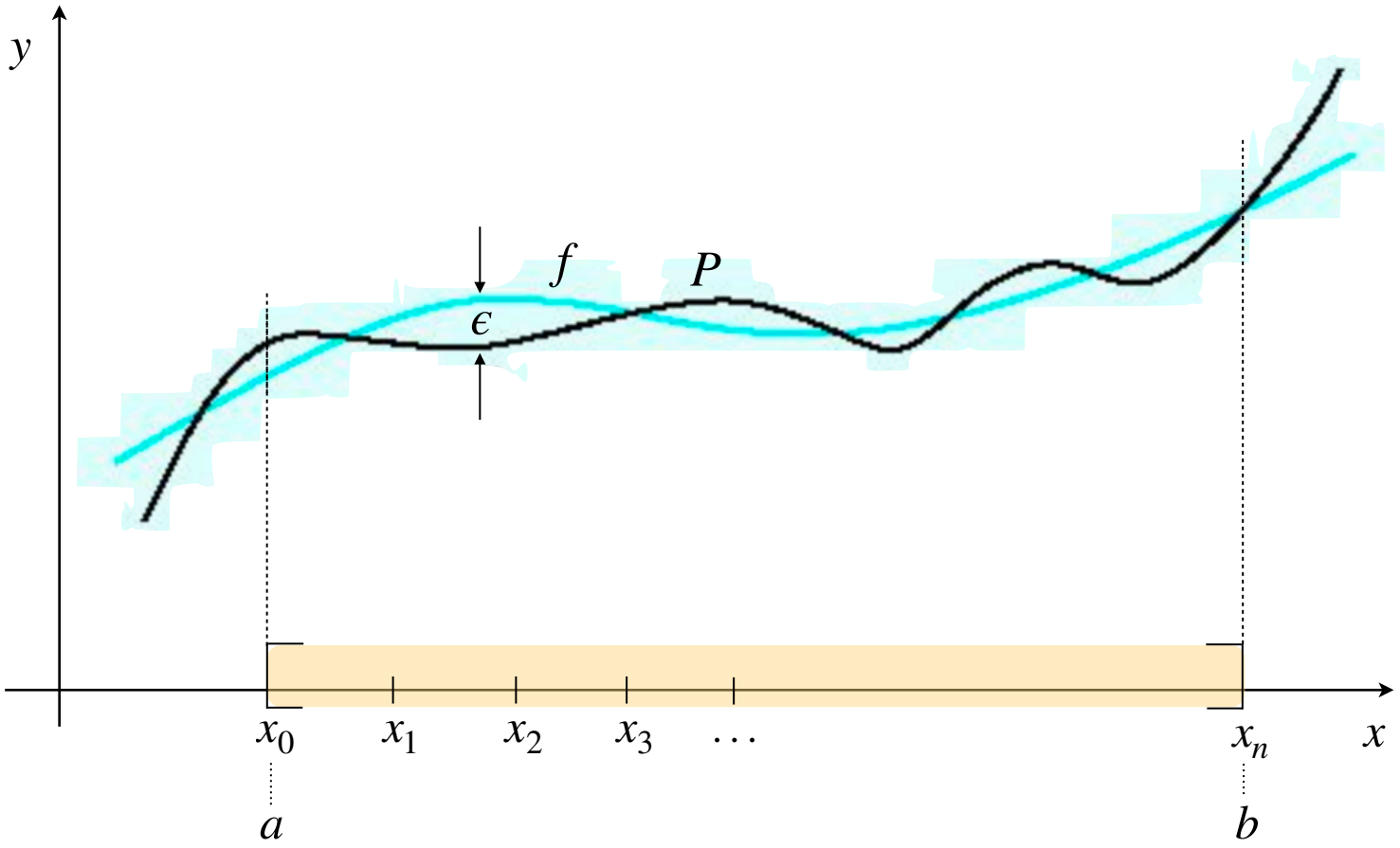
Trade-Off

Depending on our goal—prediction, inference, or a combination of the two—, different methods for estimating f may be appropriate.

- linear models allow for relatively simple and **interpretable inference**, but may not yield as accurate predictions as some other approaches
- highly non-linear approaches can potentially provide quite **accurate predictions** for Y, but this comes at the expense of a less interpretable model for which inference is more challenging
- consider the approximation theorem …

Weierstrass approximation theorem

Suppose f is a **continuous real-valued function** defined on the interval $[a, b]$.
For each $\epsilon > 0$, there exists a **polynomial function** P with real coefficients such that for all x in $[a, b]$, the property $|f(x) - P(x)| < \epsilon$ holds.



How Do We Estimate f ?

Henceforth, we always assume that we have observed a set of n **data points**.

These observations are called the training data because we use them to train a model for estimating the unknown function f. Our goal is to apply a statistical learning method to the training data for this purpose, i.e. we want to find a function $\hat{f}$ such that

$$Y \approx \hat{f}(X) \text{ for any observation } (X, Y).$$

Statistical learning methods for this task can normally be characterized as either parametric or non-parametric.

Parametric methods

Parametric methods involve a two-step model-based approach:

- First, we make an assumption about the functional form, or shape, of f. For example, one very simple assumption is that f is linear in X:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \,.$$

  Once we have assumed that f is linear, the estimation problem is greatly simplified. Instead of having to estimate an entirely arbitrary p-dimensional function $f(X)$, one only needs to estimate the $p + 1$ coefficients $\beta_0, \beta_1, \ldots, \beta_p$.

- After a model has been selected, we need a procedure that uses the training data to fit or train the model. For a linear model, we need to find parameter values such that

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \,.$$

  The most common approach to fitting the model is referred to as (ordinary) **least squares.**

Parametric methods (cont.)

The model-based approach just described is referred to as parametric; it reduces the problem of estimating f down to one of estimating a set of parameters.

- Assuming a parametric form for f simplifies the estimation problem because it is generally much easier to estimate a set of parameters, such as $\beta_0, \beta_1, \ldots, \beta_p$, than it is to fit an entirely arbitrary function f.

- The potential disadvantage of a parametric approach is that the model we choose will usually not match the true unknown form of f. If the chosen model is too far from the true f, then our estimate will be poor.

- One can try to address this problem by choosing **flexible models** that can fit many different possible functional forms for f.

- Fitting a more flexible model requires estimating a greater number of parameters. These more complex models can lead to a phenomenon known as overfitting the data, which essentially means they follow the errors, or noise, too closely.

Non-parametric Methods

Non-parametric methods do not make explicit assumptions about the functional form of f. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.

- Such approaches can have a **major advantage over parametric approaches**: by avoiding the assumption of a particular functional form for f, they have the potential to accurately fit a wider range of possible shapes for f.

- Non-parametric approaches do suffer though from **a major disadvantage**: since they do not reduce the problem of estimating f to a small number of parameters, a very large number of observations—far more than is typically needed for a parametric approach—is required in order to obtain an accurate estimate for f.

Quality of Fit

We evaluate the performance of a learning method on a given data set, by measuring how well its predictions actually match the observed data.

- Problem: How to quantify the extent to which the predicted response value for a given observation is close to the true response value for that observation

- Methods: Most commonly used is he *mean squared error* (MSE), given by

$$\text{MSE} = \frac{1}{n}\sum_{1}^{n}(y_i - f(x_i))^2$$

corrected!

where $f(x_i)$ is the prediction that f gives for the i-th observation.

The MSE will be small if most predicted responses are very close to the true responses, and will be large if for some of the observations, the predicted and true responses differ substantially.

Quality of Fit (cont.)

The MSE computed using the training data that was used to fit the model is referred to as the **training MSE**.

More interesting though is the **test MSE**: the accuracy of the predictions obtained when we apply this method to previously unseen test data.

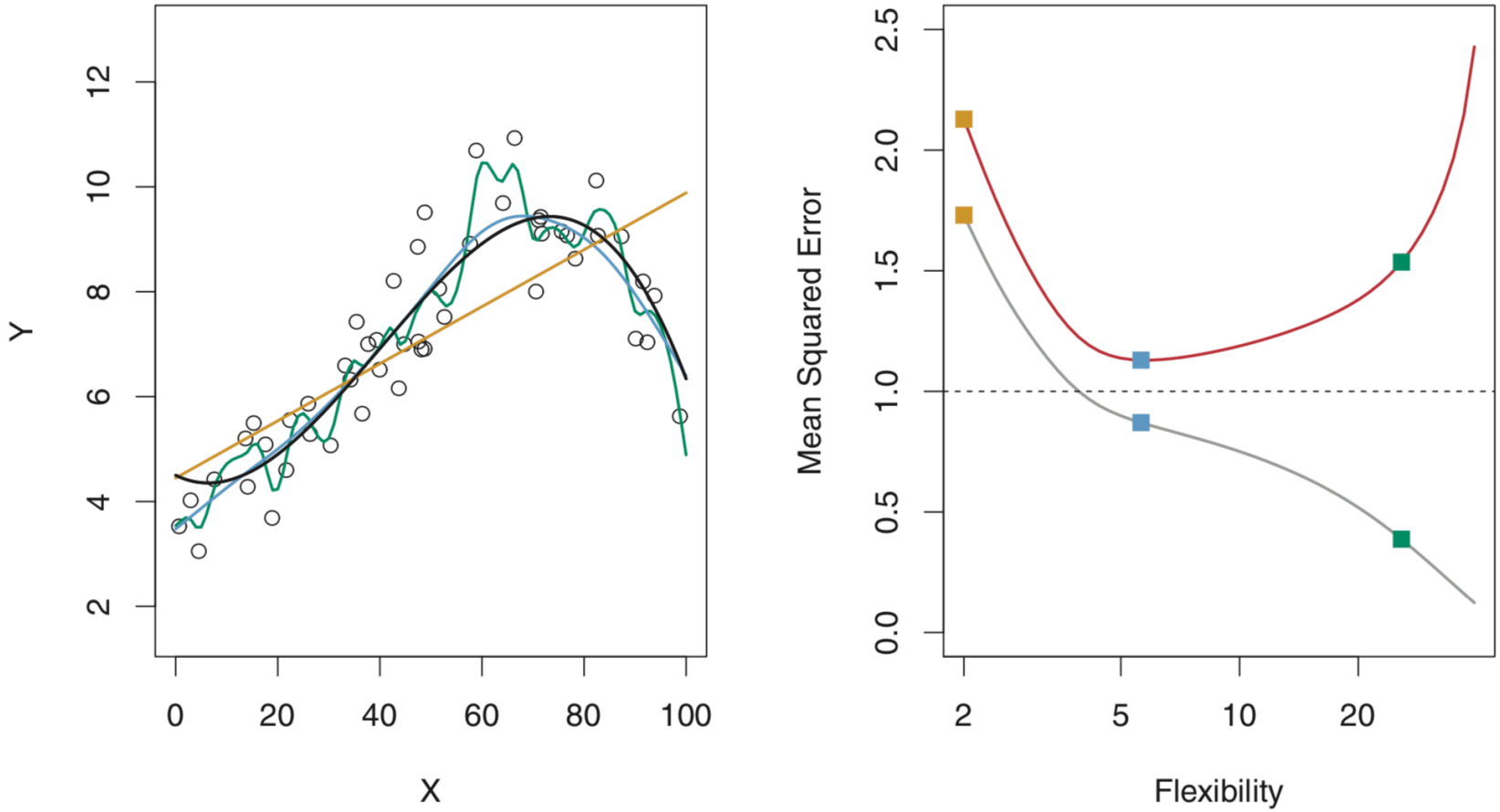
Why is this what we care about? Ultimately, we want to choose a method that gives the lowest test MSE, as opposed to the lowest training MSE.

Problem: What if no test observations are available?

- In that case, one might imagine simply selecting a statistical learning method that minimizes the training MSE. This seems like it might be a sensible approach, since the training MSE and the test MSE appear to be closely related.

- Unfortunately, there is a **fundamental problem** with this strategy: there is no guarantee that the method with the lowest training MSE will also have the lowest test MSE. Many statistical methods estimate coefficients so as to minimize the training set MSE. For these methods, the training set MSE can be quite small, but the test MSE is often much larger.

Quality of Fit (cont.)



Left: **Data simulated from f**, shown in black. Estimates of f: a linear regression line (orange), and two smoothing spline fits (blue and green). Right: Training MSE (grey), test MSE (red), 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.

Quality of Fit (cont.)

Left Panel:

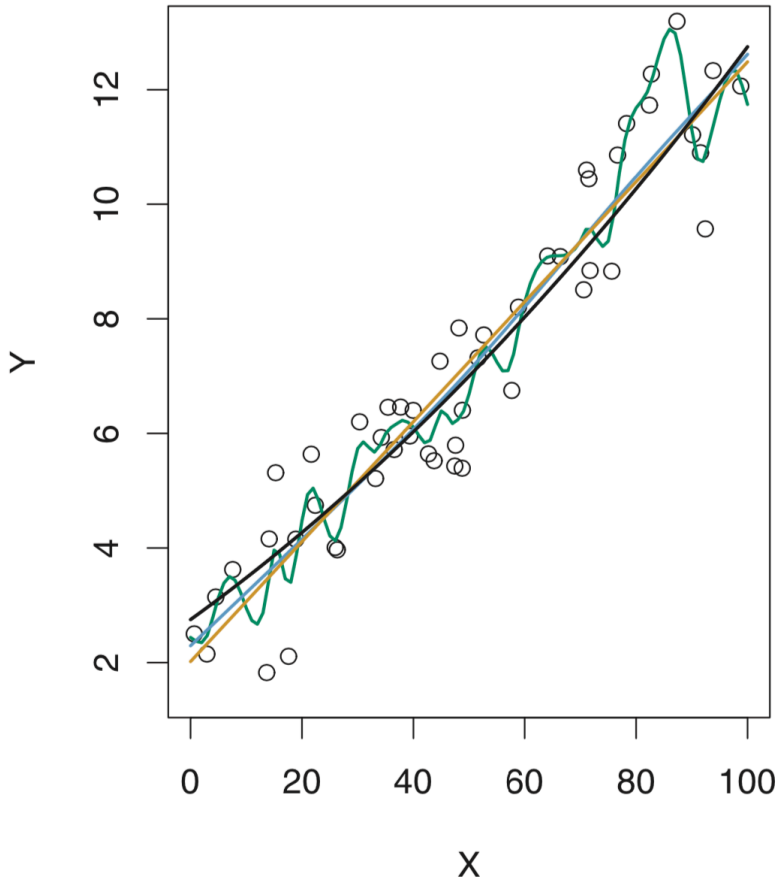
- The orange, blue and green curves illustrate **three possible estimates** for f obtained using methods with increasing levels of flexibility.
- The orange line is the linear regression fit, which is relatively inflexible.
- The blue and green curves were produced using *smoothing splines* with different levels of smoothness.
- It is clear that **as the level of flexibility increases**, the curves fit the observed data more closely. The training MSE declines monotonically as flexibility increases.
- The green curve is the most flexible and matches the data very well; we observe though that it fits the true f poorly because it is too wiggly.

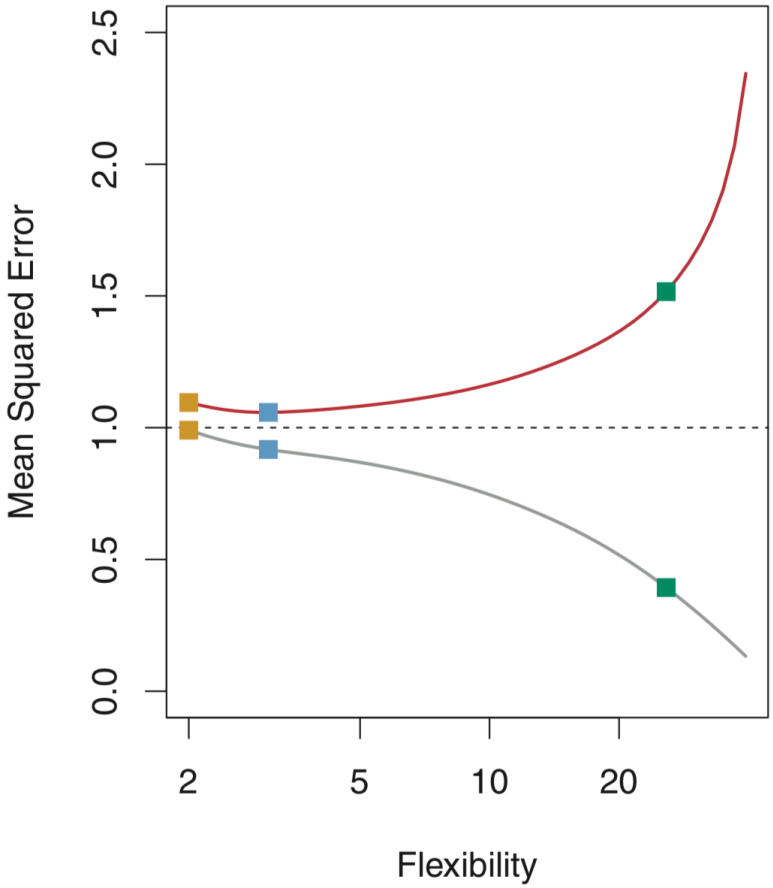
Quality of Fit (cont.)

Right Panel:

- The horizontal dashed line indicates $Var(\epsilon)$, the irreducible error, corresponding to the lowest achievable test MSE among all possible methods.
- The U-shape in the test MSE is a fundamental property of statistical learning that holds regardless of the particular data set at hand and regardless of the statistical method being used.
- As model flexibility increases, training MSE will decrease, but the test MSE may not. When a given method yields a small training MSE but a large test MSE, we are said to be **overfitting the data**.
- Overfitting happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f.
- When we overfit the training data, the test MSE will be very large because the supposed patterns that the method found in the training data simply don't exist in the test data.

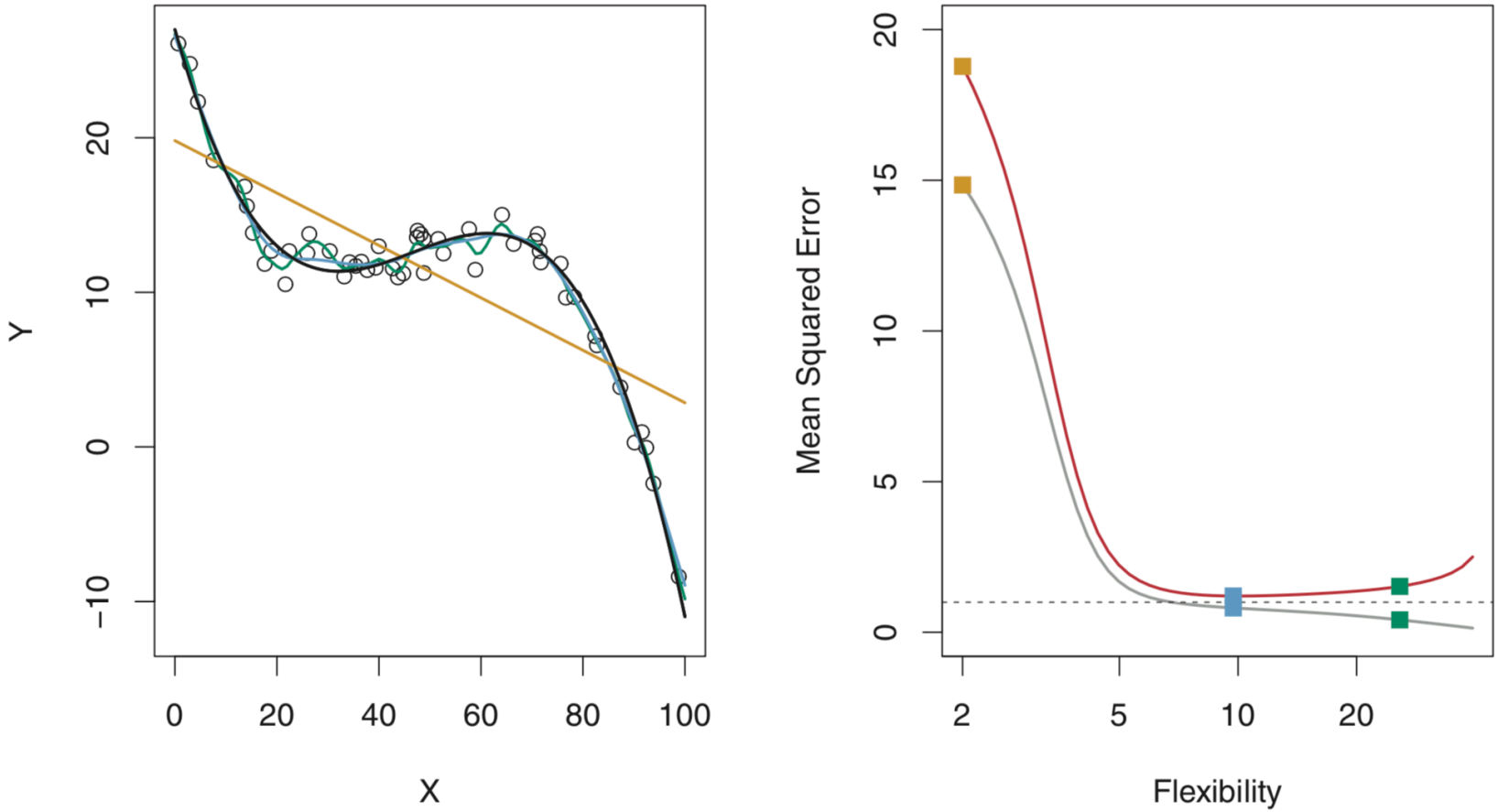
Quality of Fit (cont.)





Using a different true f that is **much closer to linear**. In this setting, linear regression provides a very good fit to the data.

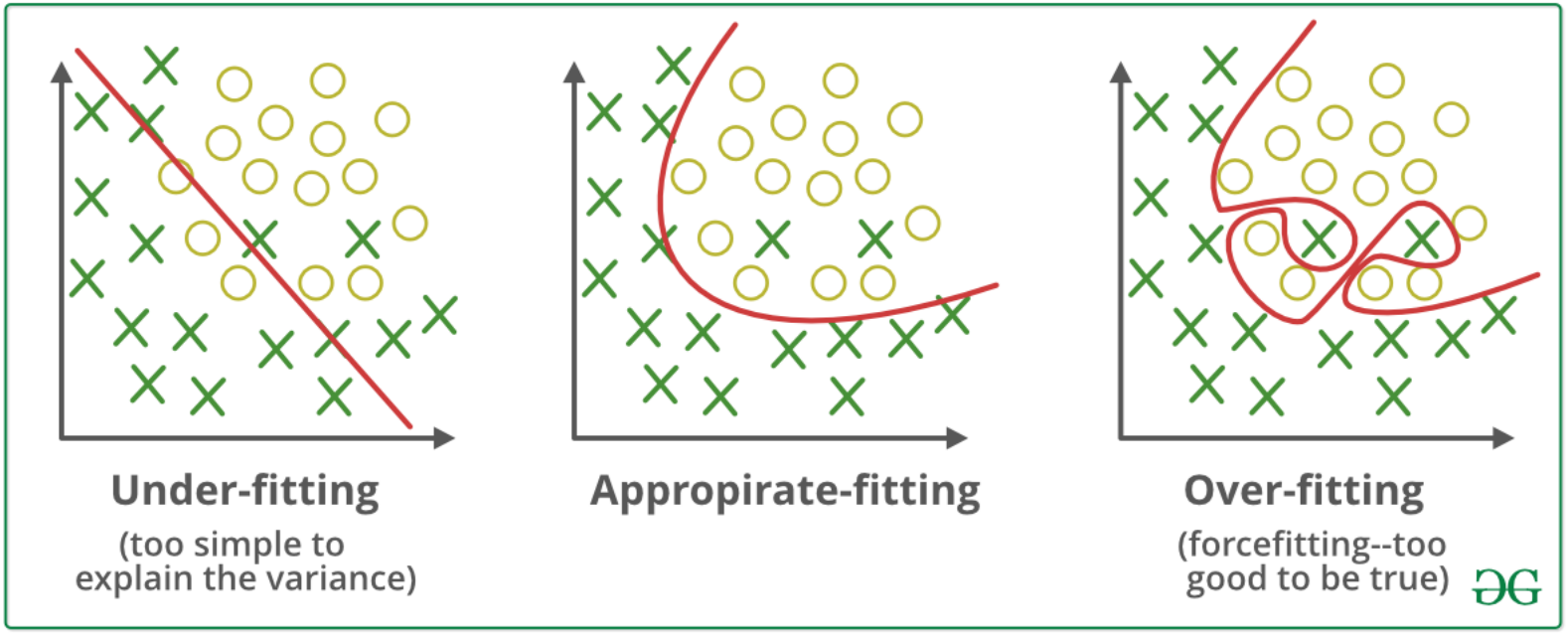
Quality of Fit (cont.)



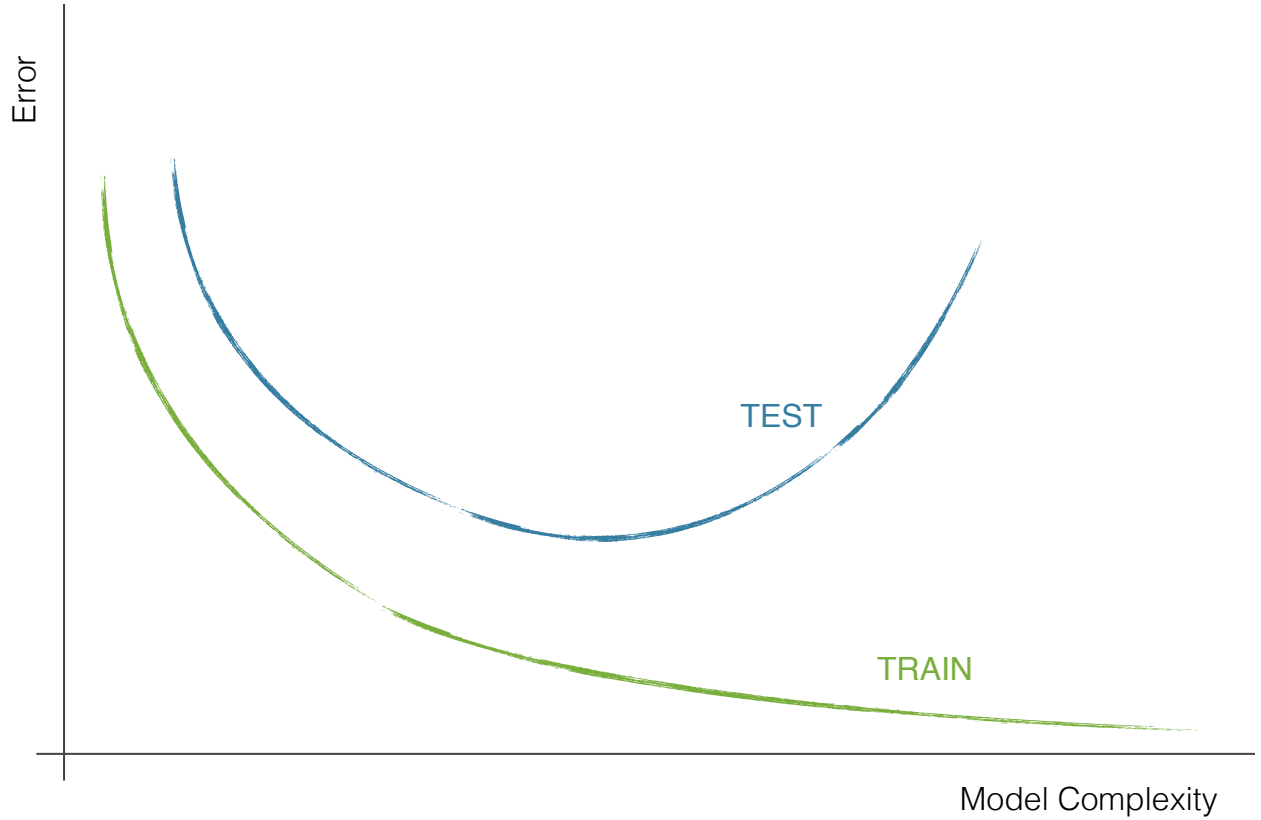
Using a different f that is **far from linear**. In this setting, linear regression provides a very poor fit to the data.

Overfitting vs. Underfitting

The cause of poor performance of a model in machine learning is either overfitting or underfitting the data.



Source: towardsdatascience.com



Overfitting

Linear Regression

- a useful tool for predicting a quantitative response
- appears dull compared to more modern statistical learning approaches
- but is still a widely used statistical learning method

Simple Linear Regression[7]

- a very straightforward approach for predicting a quantitative response Y on the basis of a single predictor variable X, assuming that there is **approximately** a linear relationship between X and Y:

$$Y \approx \beta_0 + \beta_1 X$$

Note: this assumption may, or **may not**, be adequate.

[7] **Multiple Linear Regression:** In practice X has often more than one predictor, i.e. $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$

Estimating the Coefficients

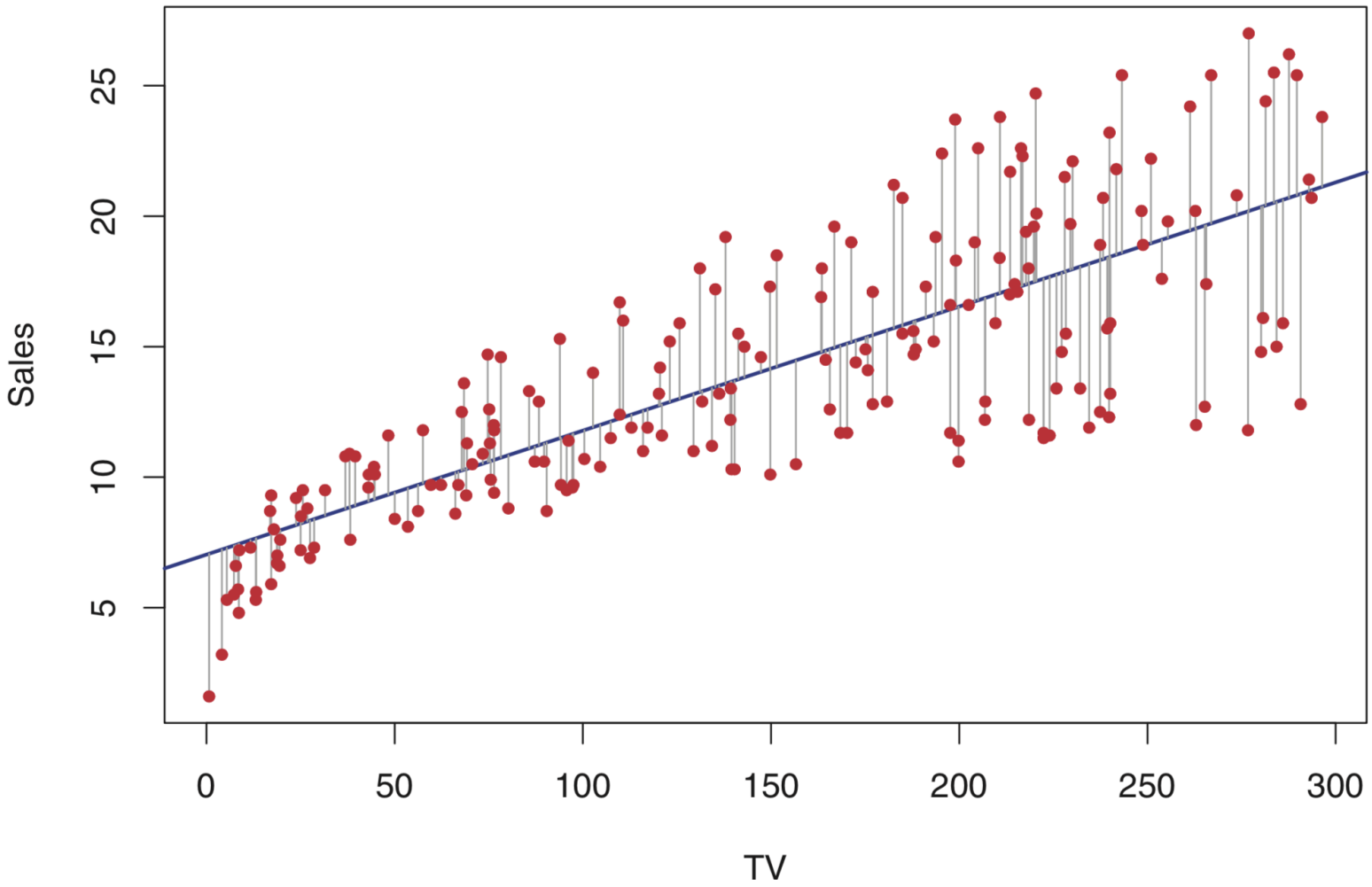
In practice, β_0 and β_1 are **unknown**, so we must use data to estimate the coefficients. Let

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

represent n **observation pairs**, each of which consists of a measurement of X and a measurement of Y.

- Our goal is to obtain coefficient estimates for the *intercept* $\hat{\beta}_0$ and the *slope* $\hat{\beta}_1$ such that the linear model **fits the available data closely**.
- There are a number of ways of measuring closeness. However, by far the most common approach involves minimizing the *least squares* criterion.

Example



Residual Sum of Squares

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y based on the i-th value of X. Then

$$e_i = y_i - \hat{y}_i$$

represents the i-th residual, i.e. the difference between the i-th observed response value and the i-th response value that is predicted by our linear model.

We define the *residual sum of squares* (RSS) as

$$\text{RSS} = e_1^2 + e_2^2 + \ldots + e_n^2,$$

or equivalently as

$$RSS = (y_1 - \hat{\beta}_0 + \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 + \hat{\beta}_1 x_2)^2 + \ldots + (y_n - \hat{\beta}_0 + \hat{\beta}_1 x_n)^2.$$

The least squares regression coefficient estimates characterize the *least squares line* $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$. The LS method chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ so as to **minimize the RSS**.

If f is to be approximated by a linear function, then we can write this relationship as

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

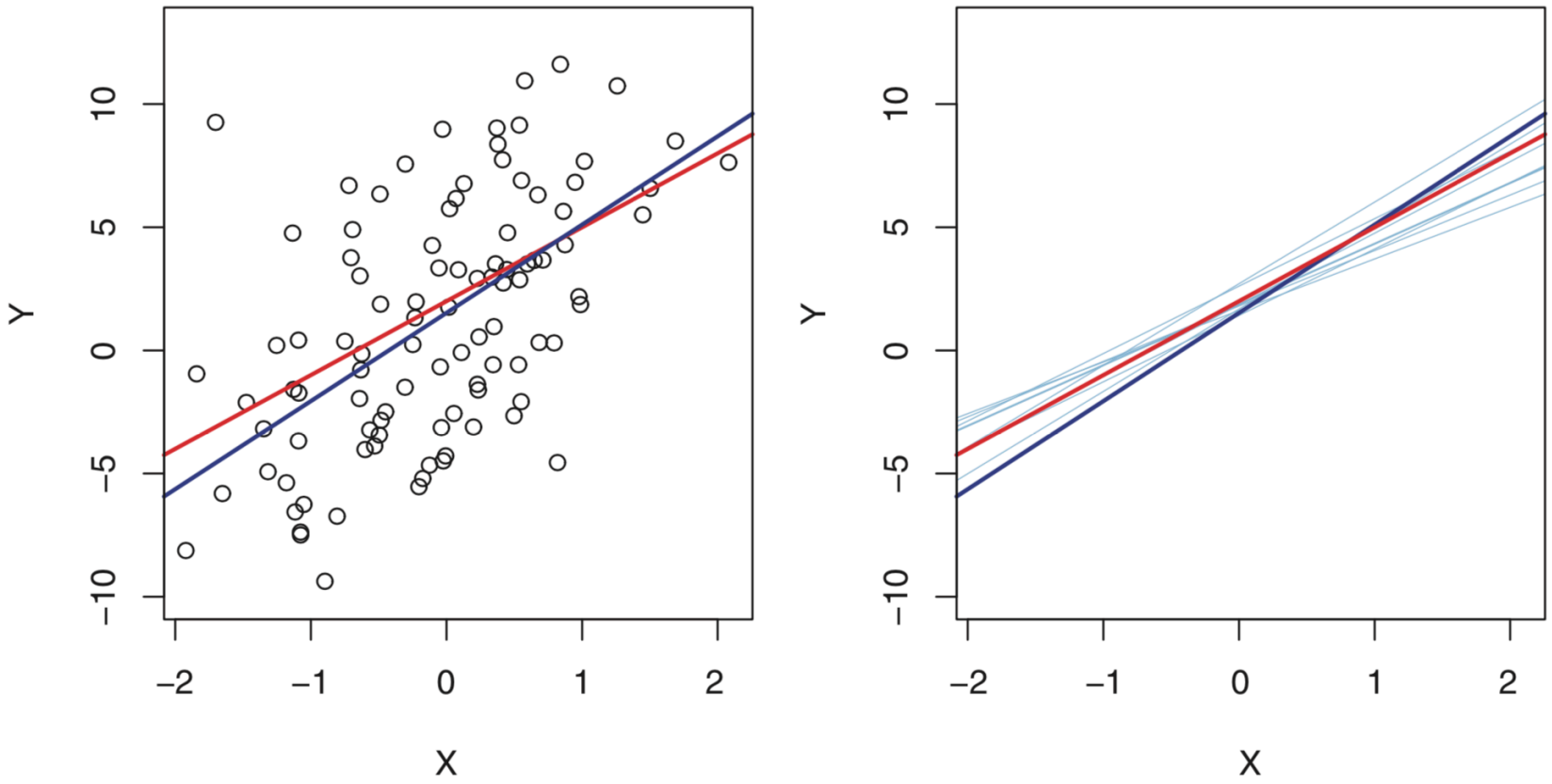
- The random error term with mean zero **is a catch-all** for what we miss with this simple model: the true relationship is probably not linear, there may be other variables that cause variation in Y, and there may be measurement error.

  We typically assume that ϵ is independent of X.

- This model defines the *population regression line*, which is the best linear approximation to the true relationship between X and Y.[8]

[8] The assumption of linearity is often a useful **working model**. However, we seldom believe that the true relationship is linear.

Example



A simulated data set. Left: The red line represents the true relationship, $f(X) = 2 + 3X$, the population regression line. The blue line is the least squares line based on the observed data, shown in black. **Right:** In light blue, 10 least squares lines are shown, each computed on the basis of a **separate** random set of observations. Each least squares line is different, but on average, the least squares lines are quite close to the population regression line.

Why this is important:

- In real applications, the true relationship is generally not known, but the least squares line can always be computed using the coefficient estimates.
- In other words, we have access to a set of observations from which we can compute the least squares line; however, **the population regression line is unobserved**.

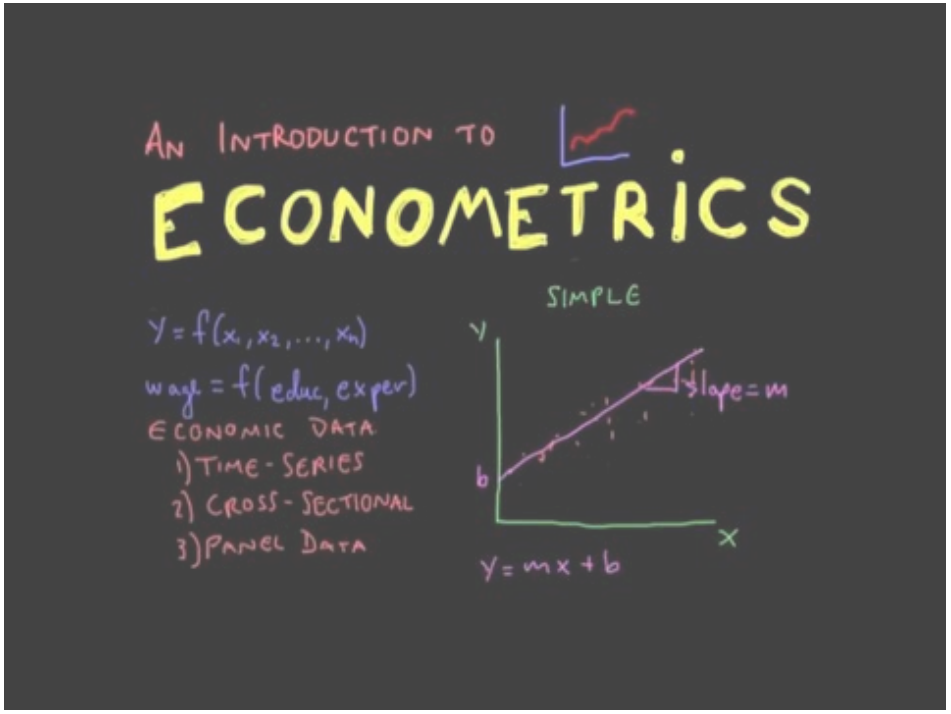
Fundamentally, the concept of these two lines is a natural extension of the standard statistical approach of using information from a sample to estimate characteristics of a large population.

For example, suppose that we are interested in knowing the population mean μ of some random variable Y. Unfortunately, μ **is unknown**, but we do have access to n observations from Y, which we can write as $y_1, \ldots, y_n$, and which we can use to estimate μ.

A reasonable estimate is $\hat{\mu} = \bar{y}$, where $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ is the sample mean.

The sample mean and the population mean are different, but in general the sample mean will provide a good estimate of the population mean.

Using Monte Carlo simulation

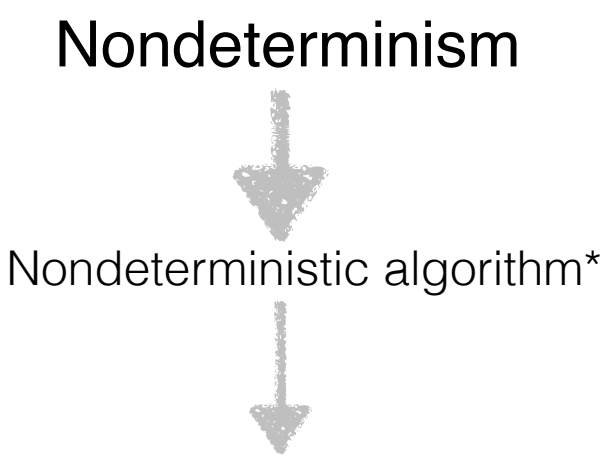


https://www.youtube.com/watch?v=5nM5e2_10Q0

... the branch of economics concerned with the use of mathematical methods (especially statistics) in describing economic systems

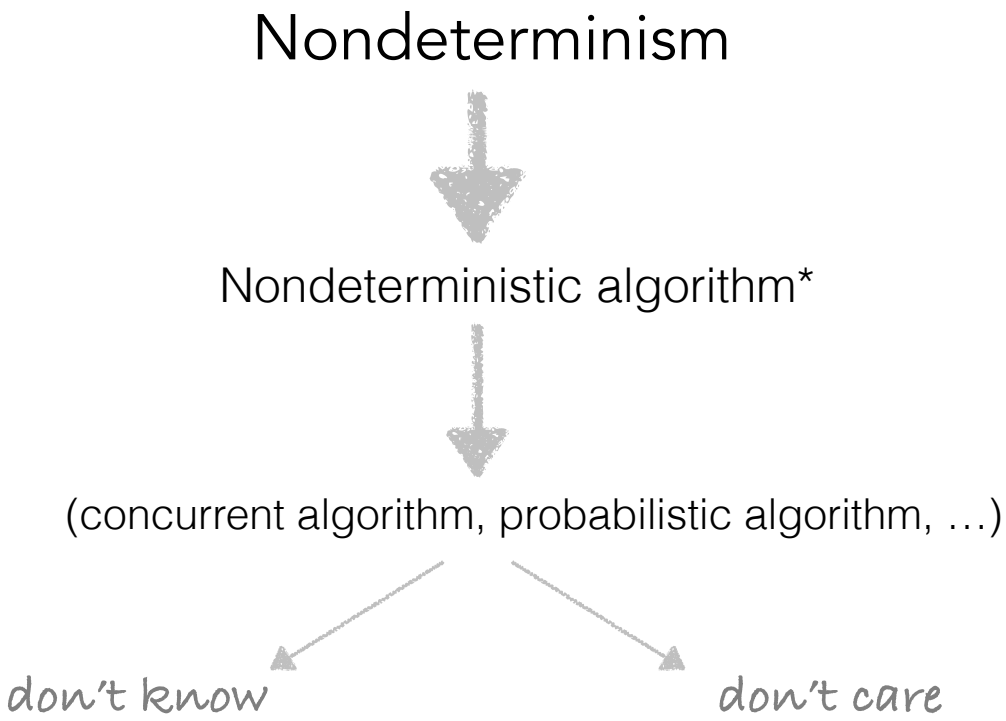
Randomness

Abstraction principles and best practices



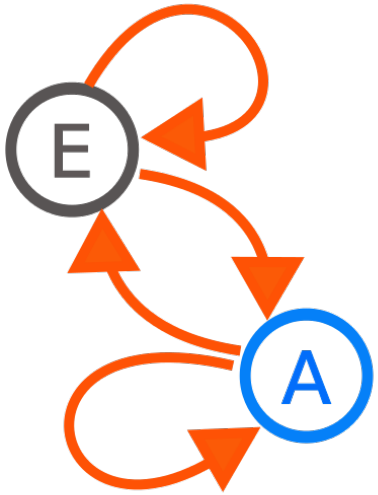
Randomness

Abstraction principles and best practices



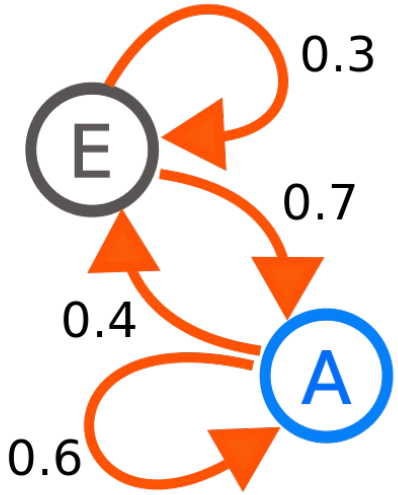
*In computer science, a nondeterministic algorithm is an algorithm that, even for the same input, can exhibit different behaviors on different runs, as opposed to a deterministic algorithm.

A simple example



Assume some computational process with two states, A and E, and state transitions as illustrated here. What is the observable behavior of the process?

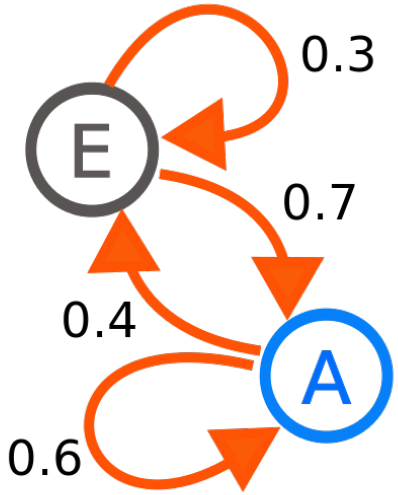
A simple example



Consistent labelling!

Each number represents the probability of the process changing from one state to another state, with the direction indicated by an arrow. For example, if the process is in state A, then the probability it changes to state E is 0.4, while the probability it remains in state A is 0.6.

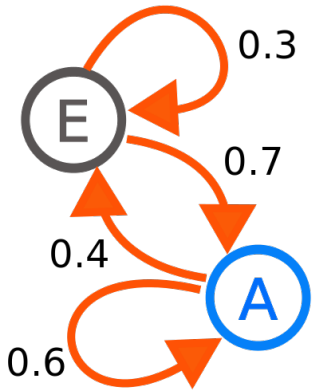
A simple example



Assume in any given run the choice of the next state does only depend on the current state (and the state transition probabilities) but neither on any previous state(s) nor on the history of computation.

A simple example

With this assumption, the below diagram can be associated with the state transition behavior of a two-state Markov process, with the states labelled **E** and **A**.



What is missing here ...?

Markov Models

A Markov process is a random process in which the future is **independent** of the past, given the presence. Named after the inventor of Markov analysis, the Russian mathematician Andrei A. Markov (1856-1922).

In probability theory, a Markov model is a **stochastic model** used to model randomly changing systems where it is assumed that future states depend only on the current state not on any events that occurred in a prior state, i.e., it assumes the Markov property.

Markov Models

A Markov process is a random process in which the future is **independent** of the past, given the presence. Named after the inventor of Markov analysis, the Russian mathematician Andrei A. Markov (1856-1922).

In probability theory, a Markov model is a **stochastic model** used to model randomly changing systems where it is assumed that future states depend only on the current state not on any events that occurred in a prior state, i.e., it assumes the Markov property.

The term Markov property refers to the memoryless property of a stochastic process, that is the conditional probability distribution of future states of the process depends only on the **present state**, not on the sequence of events that preceded it.

- Generally, this assumption enables reasoning and computation with Markov models that would otherwise be intractable. For this reason, in the fields of predictive modeling and probabilistic forecasting—a.k.a. **predictive analytics**—, it is desirable for a model to exhibit the Markov property.

Markov chain

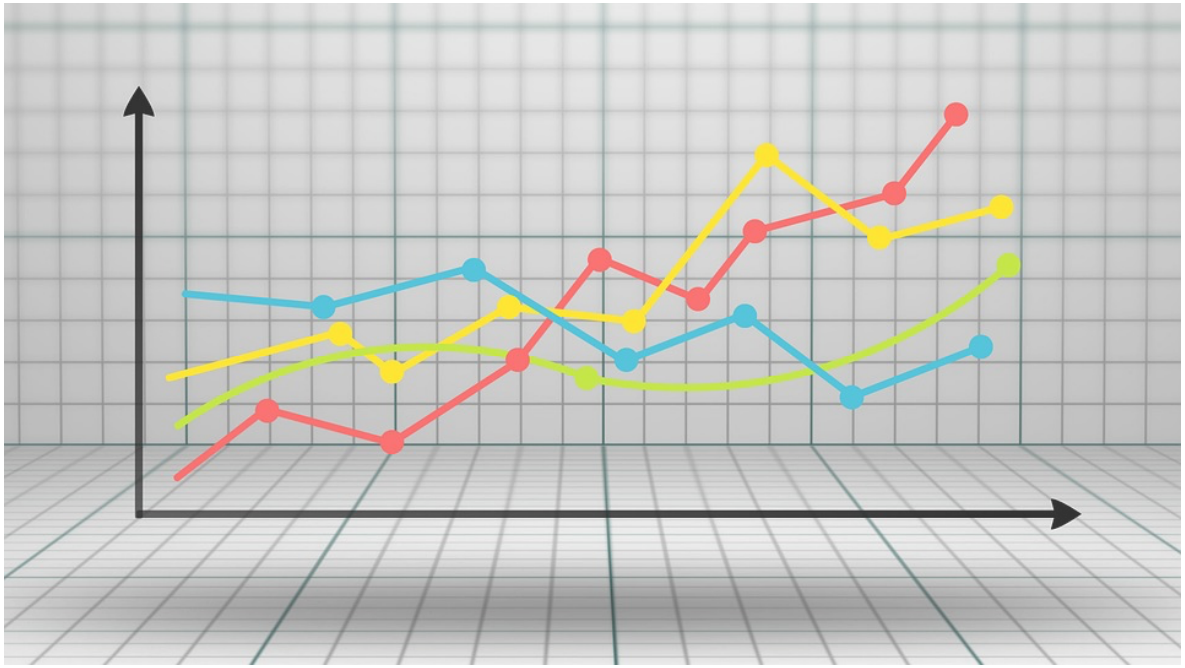
A **discrete-time stochastic process** (in contrast to continuous-time stochastic process) satisfying the Markov property is known as a Markov chain.

The term "Markov chain" refers to the sequence of random variables such a process moves through, with the Markov property defining serial dependence only between adjacent periods (as in a “chain").

- Markov chains describe systems that follow a chain of linked events, where what happens next depends only on the current state of the system.

Time Series

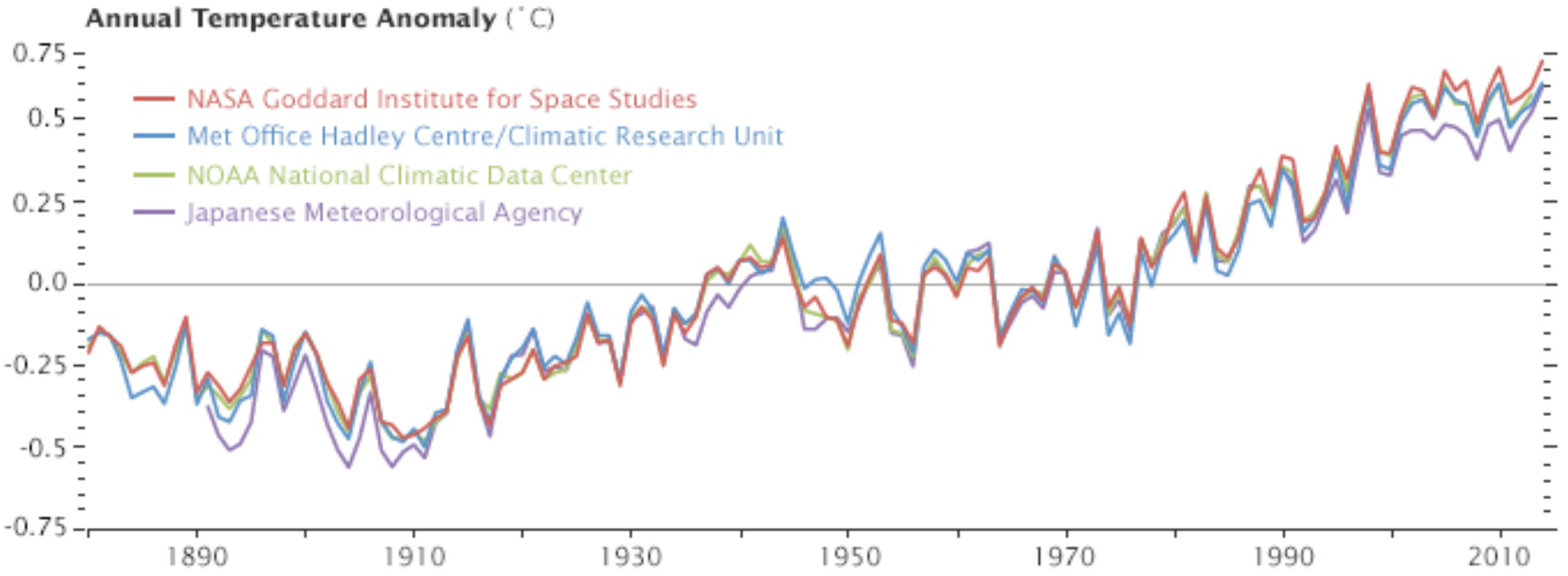
- a series of **data points ordered in time** (discrete-time data)
- most commonly, taken at successive equally spaced points in time
- *Examples*: ocean tides and the daily closing value of the Dow Jones Industrial Avg.



- Time series data often arise when monitoring industrial processes or tracking corporate business metrics.

Yearly temperature anomalies from 1880 to 2014

- How much various regions of the world have warmed or cooled when compared with a base period of 1951-1980
- The period of 1951-1980 was chosen to define “normal” or average temperature.



Univariate vs. Multivariate

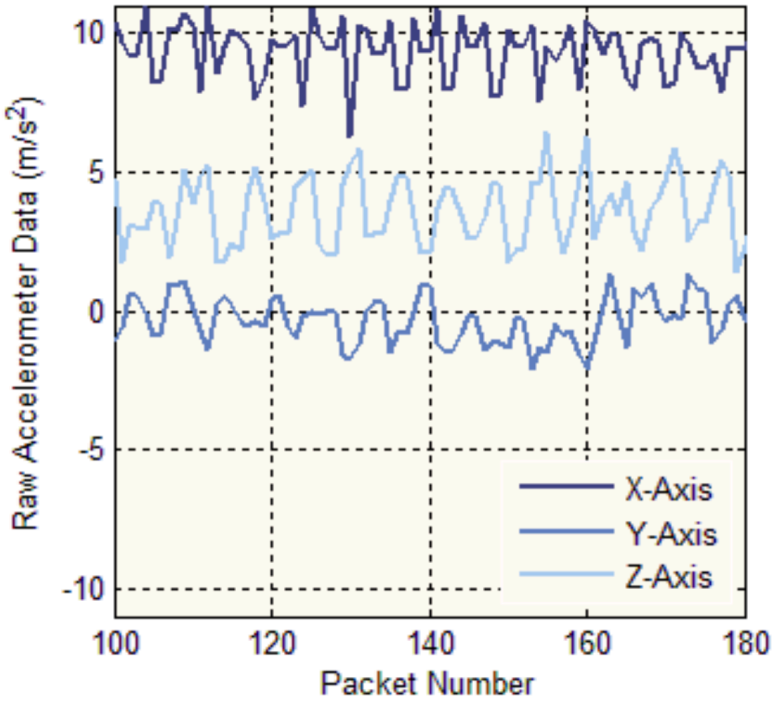
Univariate: **one variable** is varying over time

- *example:* data collected from a sensor measuring the (one-dimensional) temperature value of a room every second

Multivariate: **multiple variables** are varying over time

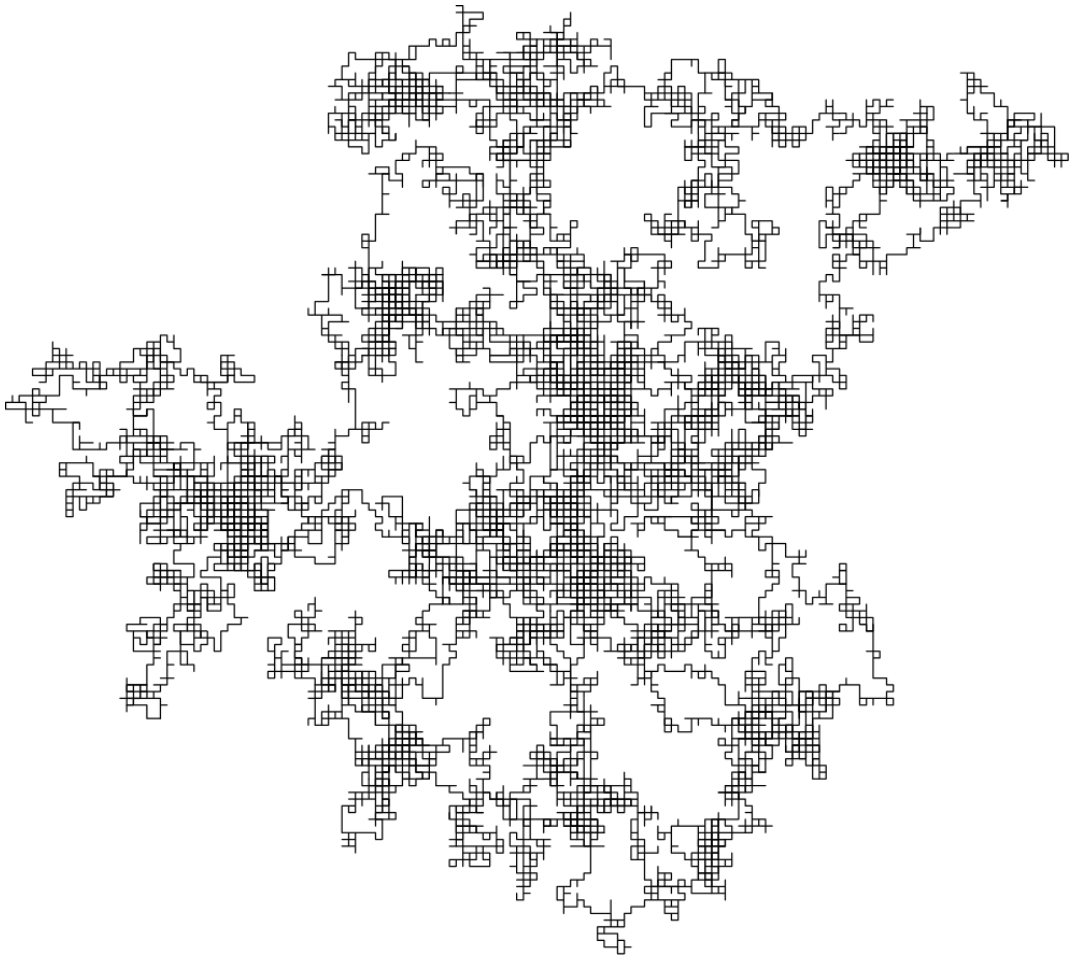
- *example:* a triaxial accelerometer measuring three accelerations (*X,Y,Z*), which vary simultaneously over time

Triaxial accelerometer attached to wrist while walking (Source: Wikipedia)



Example 1

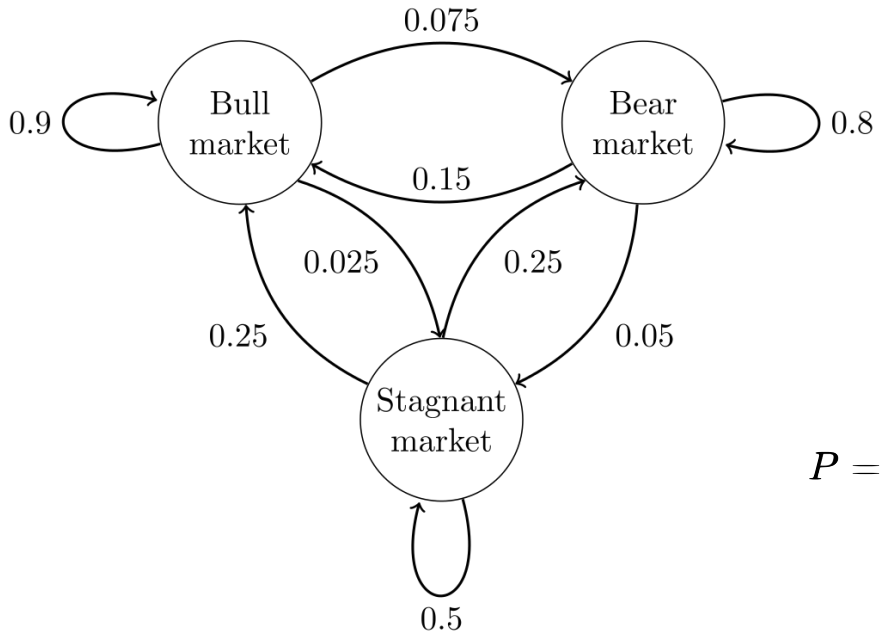
Random walk on a grid where the decision **how to continue from a given grid point** is independent of any grid points visited prior to the current one.



Random walk in two dimensions with **25 thousand steps**. (Source: Wikipedia)

Example 2

A state diagram for a simple example is using a directed graph to picture the state transitions. The states represent whether a hypothetical stock market is exhibiting a bull market, bear market, or stagnant market trend during a given week.



$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

Explanation: A bull week is followed by another bull week 90% of the time, a bear week 7.5% of the time, and a stagnant week the other 2.5% of the time. Labelling the state space {1 = bull, 2 = bear, 3 = stagnant} the transition matrix for this example is shown separately.

Example 3

Weather forecast using Markov chains?
https://www.youtube.com/watch?v=4XqWadvEj2k

Modeling Markov Processes

Source: Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77(2):257–286, 1989.

Consider real-world processes **producing observable outputs** that can be abstractly characterized as signals.

Distinct characteristic properties result in various types of signals:

- discrete or continuous (symbol alphabet — temperature measurements)
- stationary or non-stationary signal source (statistical properties vary with time)
- pure or noisy (one source only — additional sources causing distortion, noise etc.)

Modeling Markov Processes

Source: Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77(2):257–286, 1989.

Consider real-world processes **producing observable outputs** that can be abstractly characterized as signals.

Distinct characteristic properties result in various types of signals:

- discrete or continuous (symbol alphabet — temperature measurements)
- stationary or non-stationary signal source (statistical properties vary with time)
- pure or noisy (one source only — additional sources causing distortion, noise etc.)

Challenging problem: How can one **characterize real signals in terms of signal models**?

Rational for using signal models:

- provide a basis for a theoretical description of signal processing systems
- enhance understanding of the signal source even if the source is unavailable (through simulation of the real-world process)
- enable building prediction systems, recognition systems, identification systems etc.

Approaches to signal modeling

1. Deterministic models: use known specific properties of a signal (amplitude, frequency)
2. Statistical models: characterize only statistical signal properties (Gaussian, Poisson)

Statistical modeling is based on the assumption that the signal can be well characterized as a parametric random process—like a Gaussian process, Poisson process, Markov process etc.—and that the parameters of this stochastic process can be determined (estimated) in a precise, well-defined manner.

⚐ In the following we will focus exclusively on statistical approaches to signal modeling, and restrict the scope to real-world phenomena that can be viewed as Markov processes. Specifically, we use Hidden Markov models as a compact representation of process behavior for the purpose of predicting "normal" behavior in terms of sequences of observations.

Three fundamental problems in Hidden Markov model (HMM) design and analysis:

- evaluation of the probability (or likelihood) of a sequence of observations generated by a given HMM
- determination of a "best" sequence of model states
- adjustment of model parameters so as to best account for the observed signal

Discrete Markov Processes

Consider a system with N distinct states, $S_1, S_2, \ldots, S_N$. At regularly spaced discrete times t, where t = 1, 2, …, the system performs a state transition from **a given state** q_t to **the next state** q_{t+1} (possibly back to the same state) according to a set of transition probabilities associated with each state.

A complete probabilistic description of such a system generally requires specification of the current state at time t, as well as the sequence of predecessor states. For a discrete (first order) **Markov chain**, the general description can be truncated as it requires only the current and the predecessor state:

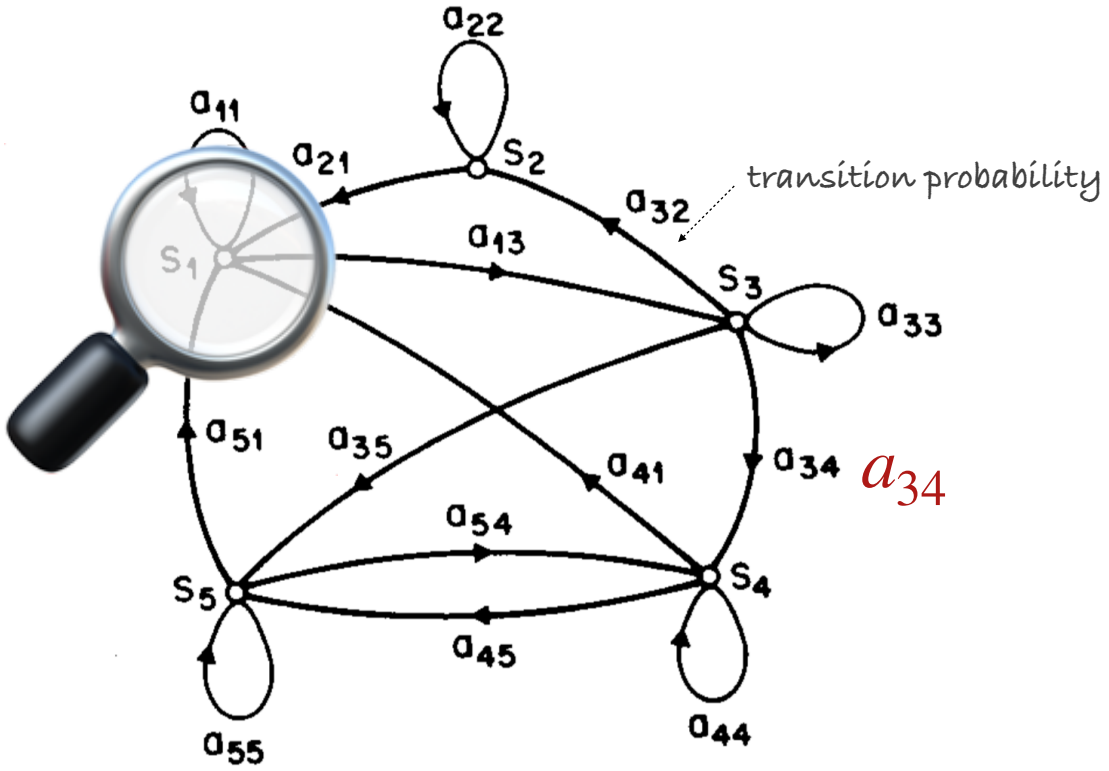
$$\begin{aligned} P[q_t = S_j \,|\, q_{t-1} = S_i, q_{t-2} = S_k, \ldots] \\ = P[q_t = S_j \,|\, q_{t-1} = S_i]\,. \end{aligned}$$

For the Markov chain, the right-hand side of the above equation does not depend on time, other than the present state, leading to a set of **state transition probabilities** a_{ij} of the form

$$a_{ij} = P[q_t = S_j \,|\, q_{t-1} = S_i], \quad 1 \le i, j \le N$$

with the additional properties on state transition coefficients: $a_{ij} \ge 0$, $\sum_{j=1}^{N} a_{ij} = 1$

Assume the **state space** of a stochastic process as illustrated below is observable and each state identifies the occurrence of a physical event. Then the output generated by the process is a sequence of observable events (states) **ordered in time**.



a_{34}

Fig. 1. A Markov chain with 5 states (labeled S_1 to S_5) with selected state transitions.

Consider a **3-state Markov chain** of "whatever" with the state transition probability matrix A and initial state probabilities $\pi_i = P[q_1 = S_i],\ 1 \le i \le N,$ with $P[q_1 = S_3] = 1$.

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

For a given **observation sequence** O with $O = S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3$, we calculate the probability that O is generated by the model in 8 consecutive steps $t = 1, 2, \ldots, 7, 8$.

3-state Markov chain

$$\begin{aligned} P(O|\text{Model}) &= P[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|\text{Model}] \\ &= P[S_3] \cdot P[S_3|S_3] \cdot P[S_3|S_3] \cdot P[S_1|S_3] \\ &\quad \cdot P[S_1|S_1] \cdot P[S_3|S_1] \cdot P[S_2|S_3] \cdot P[S_3|S_2] \\ &= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\ &= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

Given the model is in a known state, what is the probability that it stays in that state for exactly d steps? For the observation sequence

$$O = \{\underset{1}{S_i}, \underset{2}{S_i}, \underset{3}{S_i}, \ldots, \underset{d}{S_i}, \underset{d+1}{S_j \neq S_i}\}$$

the probability that this sequence occurs can be evaluated as

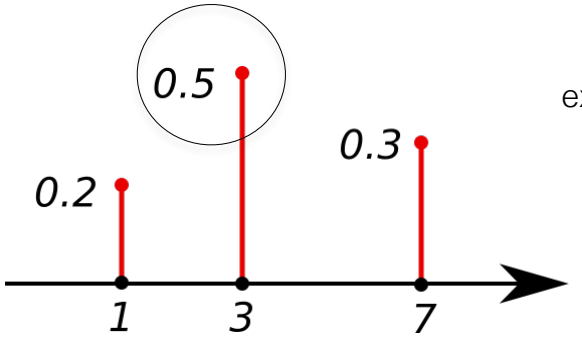
$$P(O \,|\, \text{Model}, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{ii}) = p_i(d)\,.$$

The quantity $p_i(d)$ is the discrete probability density function—a.k.a. probability mass function—of duration d in state i. This exponential duration density is **characteristic of the state duration in a Markov chain**.

Probability mass function

- A probability mass function states the probability that a discrete random variable is exactly equal to some value.
- The value of the random variable having the largest probability mass is called the mode; it is the value at which its probability mass function takes its maximum value (i.e. the value that is most likely to be sampled).

Probability mass function for a random variable with sample space {1,3,7}



expected value = 3.8

All the values of this function must be non-negative and sum up to 1.

Source: Wikipedia

Given the model is in a known state, what is the probability that it stays in that state for exactly d steps? For the observation sequence

$$O = \{\underset{1}{S_i}, \underset{2}{S_i}, \underset{3}{S_i}, \ldots, \underset{d}{S_i}, \underset{d+1}{S_j \neq S_i}\}$$

the probability that this sequence occurs can be evaluated as

$$P(O \mid \text{Model}, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{ii}) = p_i(d)\,.$$

The quantity $p_i(d)$ is the discrete probability density function—a.k.a. probability mass function—of duration d in state i. This exponential duration density **is characteristic of the state duration in a Markov chain**.

Based on $p_i(d)$, we calculate the expected durations in a state, conditioned on starting in that state as

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d)$$

$$= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) = \frac{1}{1 - a_{ii}}.$$

Given the model is in a known state, what is the probability that it stays in that state for exactly d steps? For the observation sequence

$$O = \{\underset{1}{S_i}, \underset{2}{S_i}, \underset{3}{S_i}, \ldots, \underset{d}{S_i}, \underset{d+1}{S_j \neq S_i}\}$$

the probability that this sequence occurs can be evaluated as

$$P(O \mid \text{Model}, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{ii}) = p_i(d)\,.$$

The quantity $p_i(d)$ is the discrete probability density function—a.k.a. probability mass function—of duration d in state i. This exponential duration density is characteristic of the state duration in a Markov chain.

Based on $p_i(d)$, we calculate the expected durations in a state, conditioned on starting in that state as

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d)$$

The expected number of consecutive states for S3 is 1 / (1 - 0.8) = 5, for S2 is 2.5, and for S1 is 1.67.

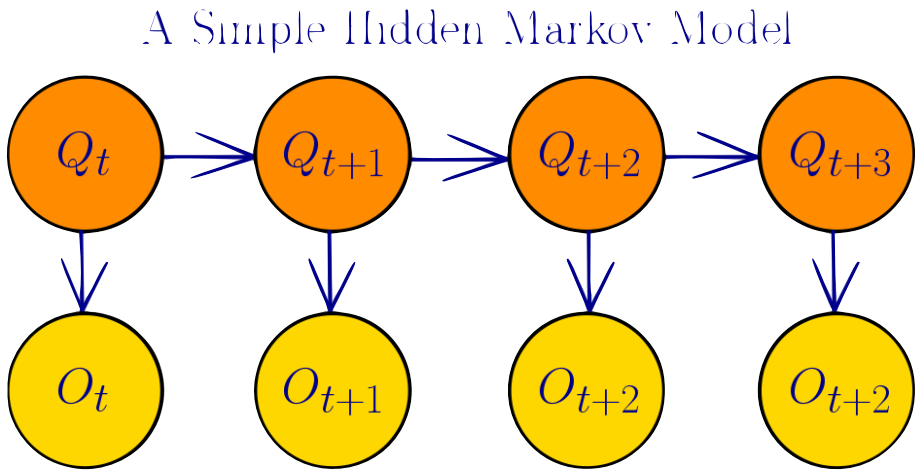
$$= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) = \frac{1}{1 - a_{ii}}.$$

Hidden Markov model

A hidden Markov model (HMM) is a Markov model in which the system being modeled is assumed to be a Markov process with **unobserved (hidden) states**.

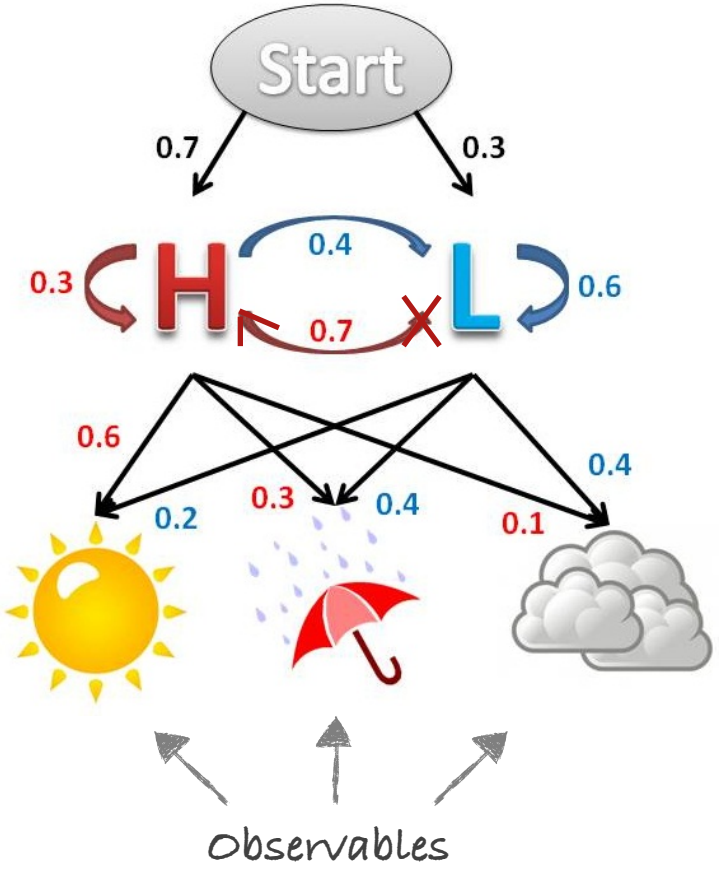
In simpler Markov models (like a Markov chain) states are **directly observable**, whereas in an HMM the state sequence through which the model passes remains hidden.

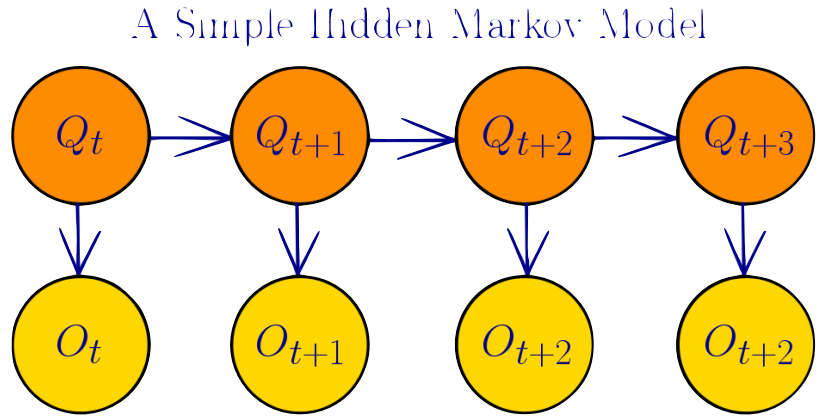
In an HMM, the observation is a probabilistic function of the state. The resulting model is a **doubly embedded stochastic process** that is not directly observable, but can be observed indirectly through another set of stochastic processes that produce a sequence of observations.



Modeling Markov Processes

Representing a weather system's behavior as a hidden Markov model





Probabilistic functions of Markov chains

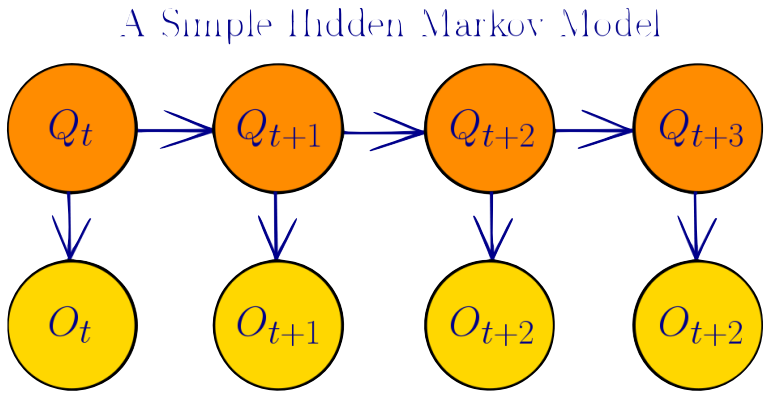
What's the problem here?

Hidden Markov model

A hidden Markov model (HMM) is a Markov model in which the system being modeled is assumed to be a Markov process with **unobserved (hidden) states**.

In simpler Markov models (like a Markov chain) states are **directly observable**, whereas in an HMM the state sequence through which the model passes remains hidden.

In an HMM, the observation is a probabilistic function of the state. The resulting model is a **doubly embedded stochastic process** that is not directly observable, but can be observed indirectly through another set of stochastic processes that produce a sequence of observations.



Each state has a **probability distribution** over the possible output tokens. Thus, the sequence of observations ("tokens" generated by a HMM) gives some information about the underlying sequence of states associated with the **observed token sequence**.

Hidden Markov models are especially known for applications in **temporal pattern recognition** in signal processing such as

- speech
- handwriting
- gesture recognition
- part-of-speech tagging
- musical score following
- partial discharges
- bioinformatics
- …

the process of automatically listening to a live music performance and tracking the position in the score; an active area of research at the intersection of artificial intelligence, pattern recognition, signal processing, and musicology.

Thus, it is not surprising that HMMs can also be used for **detecting anomalous patterns** in time series data as will be explained.

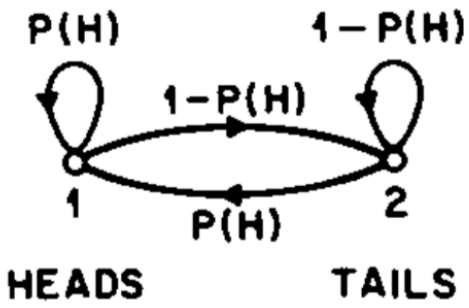
A Tibetan musical score from the 19th century.

Coin Tossing Scenarios

Someone performs a series of coin tosses using one or more coins. We cannot observe the coins but only learn about the outcomes as observation sequence of HEADS or TAILS.

How can we build an HMM that models the observed sequence of heads or tails?

- There is more than one way of doing this. Choices we make depend on what the states of the model represent and how many states we intend to use.
- Let's start with a simple **2-state model** where one state represents HEAD and the other TAIL. That is, we assume only a single (biased) coin is being tossed. So we only need to decide on the bias, say the probability of HEADS.

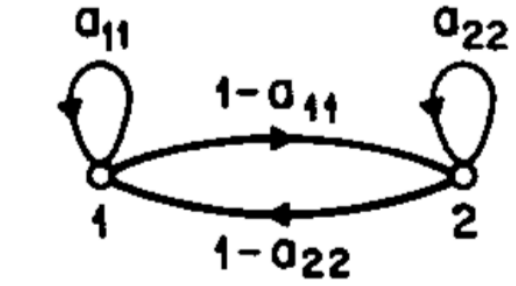


O = H H T T H T H H T T H ...
S = 1 1 2 2 1 2 1 1 2 2 1 ...

Not really a Hidden Markov model

- Alternatively, one could in principle also use a **1-state model** and encode the bias in the probabilistic observability function of the state (resulting in a degenerate model).

- A **2-state model** for representing **two different coins** with different biases, where each state is characterized by its probability distribution of HEADS and TAILS as well as its state transition probabilities:



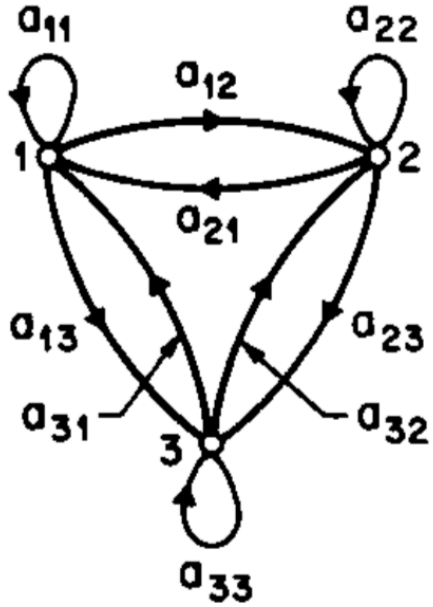
$P(H) = P_1$ $P(H) = P_2$

$P(T) = 1-P_1$ $P(T) = 1-P_2$

O = H H T T H T H H T T H ...

S = 2 1 1 2 2 2 1 2 2 1 2 ...

- Finally, a model for representing three different biased coins:



O = H H T T H T H H T T H ...
S = 3 1 2 3 3 1 1 2 3 1 3 ...

	STATE 1	2	3
P(H)	P_1	P_2	P_3
P(T)	$1-P_1$	$1-P_2$	$1-P_3$

Given the choices, which of the three models (excluding the degenerate one) matches best the actual observations of HEADS and TAILS?

Model complexity:

- The 1-coin model has only one unknown parameter.
- The 2-coin model has four unknown parameters.
- The 3-coin model has nine unknown parameters.

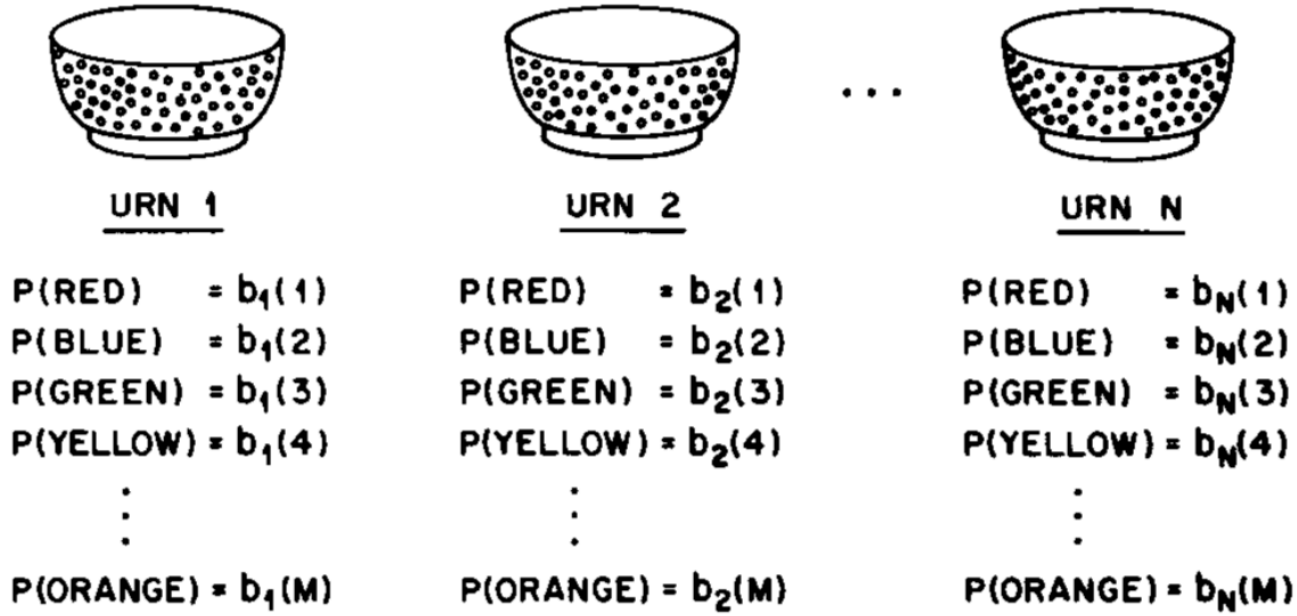
With the extra degrees of freedom, the larger HMM may somehow seem more capable of modeling a series of coin tossing experiments; however, there are additional considerations to be taken into account, which impose limitations on the size of the model for practical reasons.

Specifically, if the actual experiments use only a single coin, the 3-state model would be highly inappropriate as the model would not correspond to the actual physical event.

⚐ **A good solution** is a model with **the least complexity** necessary to reflect the observable behavior of the Markov process with sufficient accuracy in a **direct and intuitive way**. Generally, there may be (and usually is) more than one such model.

The Urn and Ball Process

In its discrete form, a hidden Markov process can be visualized as a generalization of the urn problem with replacement (where each item taken from the urn is returned to the original urn before the next step).



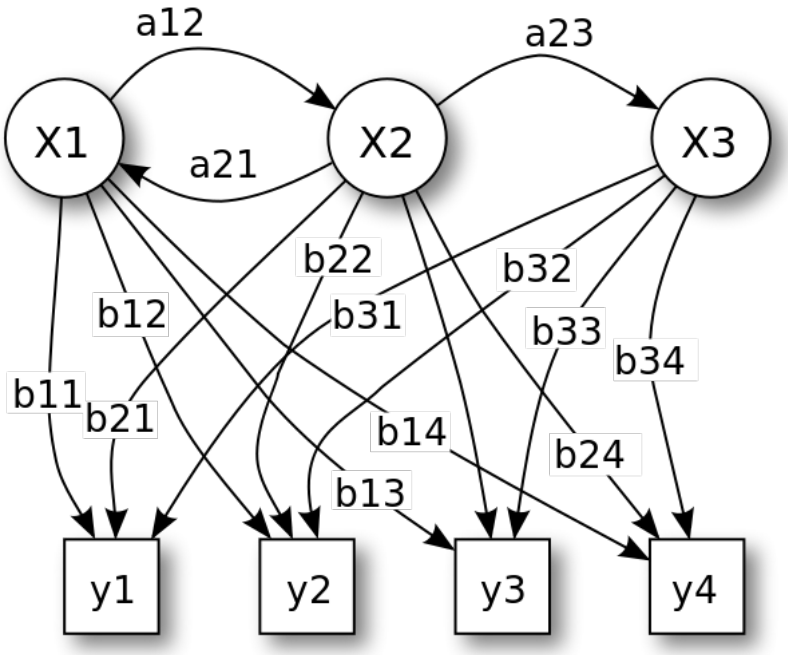
N-state urn and ball model to illustrate the general case of a discrete symbol HMM

Problem definition

- In a room, not visible to an observer, there is a “genie”.
- Connected to the room is a (moving) conveyor belt visible to the observer.
- The room contains urns X1, X2, X3, …, each of which contains (a large number of) balls labeled y1, y2, y3, …, **with a known mix of balls**.
- The genie chooses an urn, using some random selection process (1), and randomly draws a ball from that urn (2).
- It then puts the identical label onto a conveyer belt, where the observer can observe the sequence of ball labels but not the sequence of urns from which they were drawn.

Problem definition (cont.)

- The procedure to choose the urn for the n-th ball depends only upon the choice of the urn for the $(n-1)$-th ball, but does not directly depend on the urns chosen before this single previous urn; therefore, this is called a (first order) Markov process.



Scenario with 3 urns, **4 different types of balls**, where the probability that a certain ball type is drawn from a given urn is illustrated by the labels **bij**.

The Markov process itself cannot be observed, only the sequence of labeled balls, thus this arrangement is called a "hidden Markov process".

One can see that balls y1, y2, y3, y4 can be drawn at each state. Even if the observer knows the composition of the urns and has just observed a sequence of three balls, e.g. y1, y2 and y3 on the conveyor belt, the observer still cannot be certain which urn the genie has drawn the third ball from. One can reason though that the starting state was either X1 or X2.

Still, the observer can work out information, such as the likelihood that the third ball came from each of the urns.

Probabilistic parameters of a HMM:

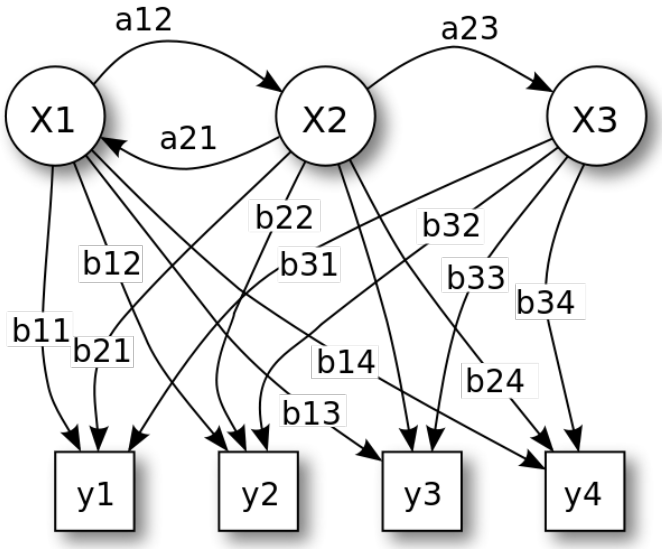
X — states

y — possible observations

a — state transition probabilities

b — output probabilities

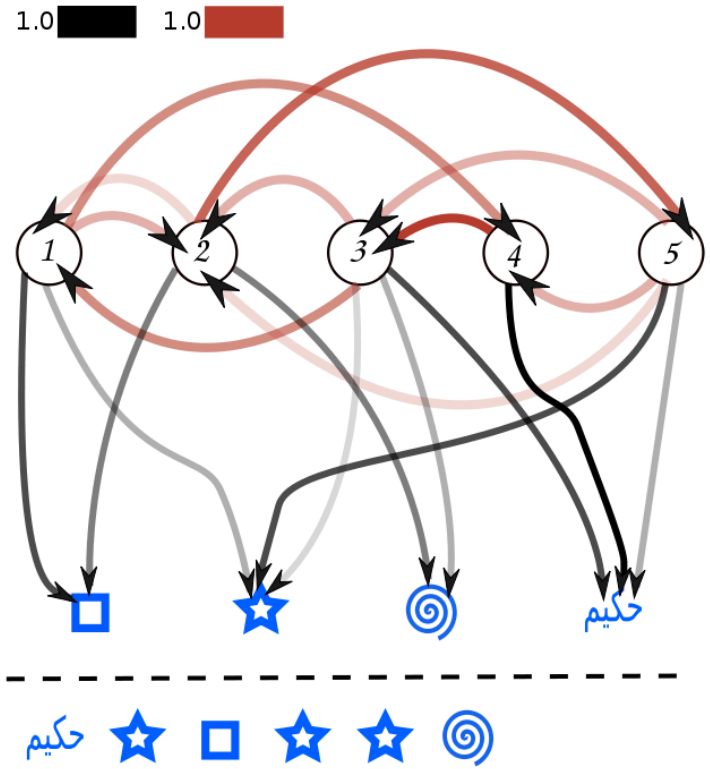
…



An illustrative example

The state transition and output probabilities of an HMM are indicated by the **line opacity** in the upper part of the diagram. Given the observed output sequence in the lower part of the diagram, we may be interested in the most likely sequence of states that could have produced it. Based on the arrows that are present in the diagram, the following state sequences are candidates:

5 3 2 5 3 2
4 3 2 5 3 2
3 1 2 5 3 2



Source: Wikipedia

We can find the most likely sequence by evaluating the joint probability of both the state sequence and the observations for each case (simply by **multiplying the probability values**, which correspond to the opacities of the arrows involved).

The simplest HMM that corresponds to the urn and ball process is one in which

- each state corresponds to a specific urn, and for which
- a (ball) **label probability** is defined for each state;
- the choice of urns is dictated by the **state transition matrix** of the HMM.

Real-life example

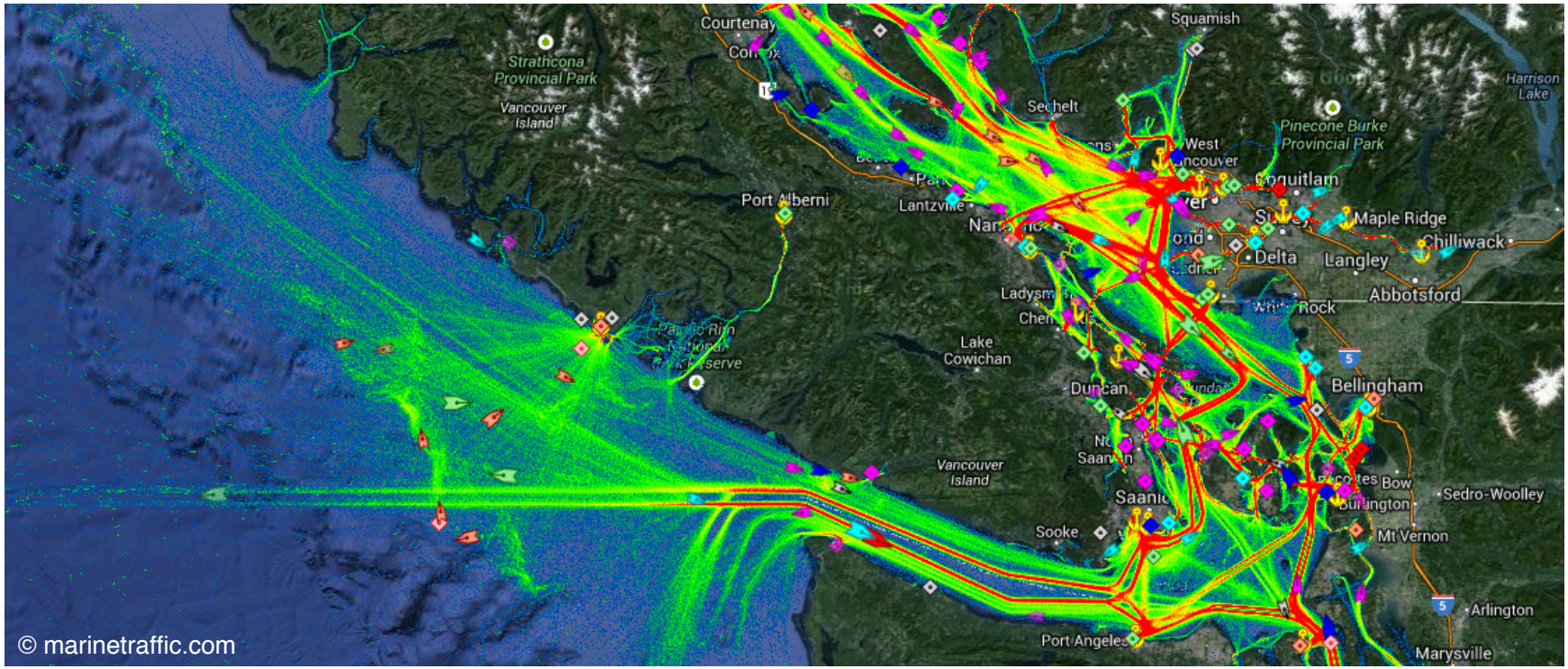
Maritime Situational Awareness

Marine vessel monitoring and tracking with Automated Identification System - AIS

https://www.marinetraffic.com

AIS data heat map:

Tracking marine traffic on the West Coast near the ports of Seattle and Vancouver

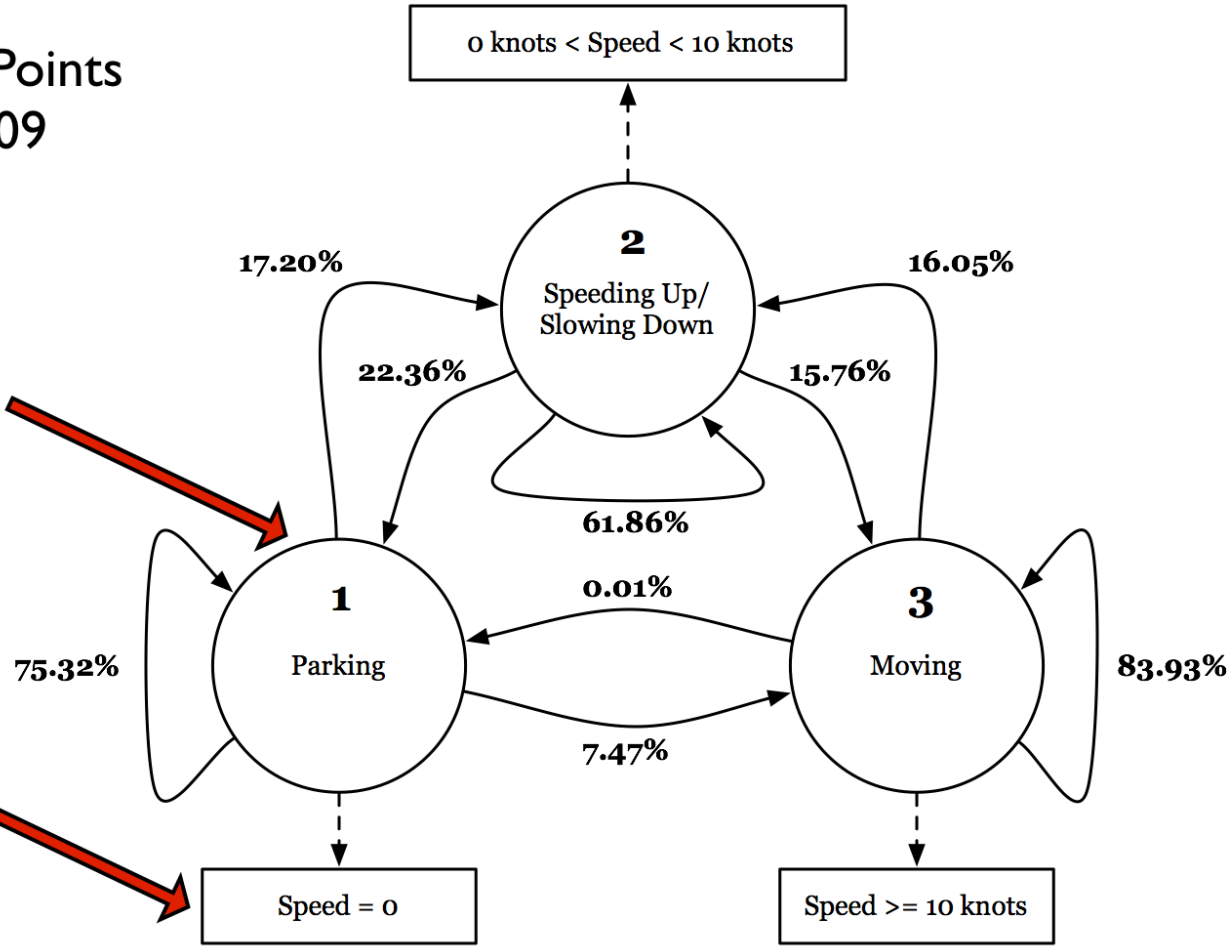


Seabus movement pattern

57729 AIS Data Points
1-31October 2009

Hidden States

Observations

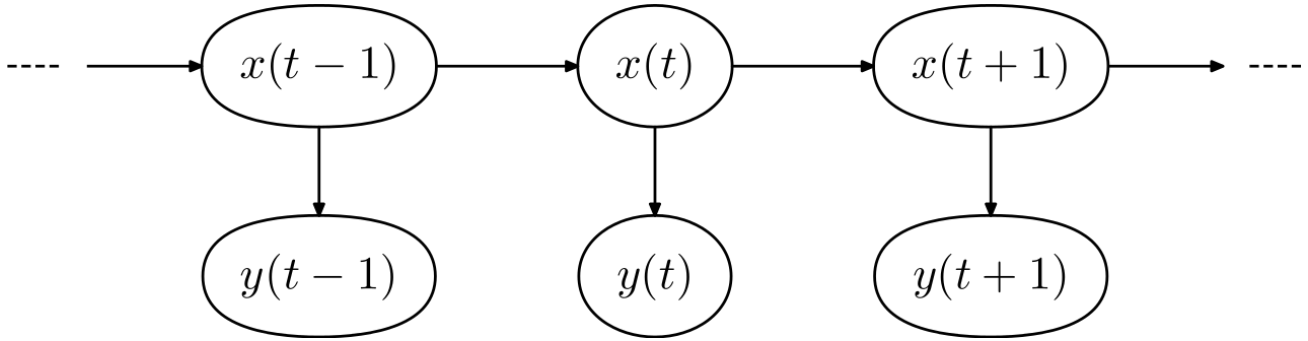


HMM Architecture

The diagram below shows the **generic architecture of an instantiated HMM**:

- each oval represents a random variable that can adopt any of a number of values;
- the random variable $x(t)$ is the hidden state at time t;
- the random variable $y(t)$ is the observation at time t.

Arrows in the diagram denote conditional dependencies.



HMM Abstract Characterization

In the standard type of hidden Markov model, the state space of the hidden variables is discrete, while the observation values can either be discrete or continuous. The **parameters of a hidden Markov model** are of two types: (i) TRANSITION PROBABILITIES and INITIAL STATE PROBABILITIES, and (ii) OUTPUT PROBABILITIES. The transition probabilities control the way the hidden state at time t is chosen given the hidden state at time $t-1$.

- The hidden state space is assumed to consist of one of N possible values, modeled as a categorical distribution (a discrete distribution characterized by a probability mass function). This means that for each of the N possible states a hidden variable at time t can be in, there is a transition probability from this state to each of the N possible states of the hidden variable at time $t+1$, for a total of N^2 TRANSITION PROBABILITIES.

- Note that the set of transition probabilities for transitions from any given state must sum to 1. Thus, the NxN-matrix of transition probabilities is a Markov matrix.

- In addition, for each of the N possible states, there **is a set of** OUTPUT PROBABILITIES governing the distribution of the observed variable values at a particular time given the state of the hidden variable at that time. The size of this set depends on the nature of the observed variable.

Hidden Markov Model - Definition

A HMM is characterized in terms of five aspects (Rabiner, 1989):

1. N, the **number of states** in the model; S, the states $S = \{S_1, S_2, \ldots, S_N\}$, and q_t, the state at time t.

   M, the number of **distinct observation symbols** per state—the size of the discrete alphabet—and the individual symbols as $V = \{v_1, v_2, \ldots, v_M\}$.

   Observation symbols correspond to the PHYSICAL OUTPUT of the system being modeled.

2. The **state transition probability distribution** $A = \{a_{ij}\}$, where

$$a_{ij} = P[q_{t+1} = S_j \,|\, q_t = S_i], \quad 1 \leq i, j \leq N.$$

   If each state can reach every other state in a single step, we have $a_{ij} > 0$ for all i, j.

3. The **observation symbol probability distribution** in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P[v_k \text{ at } t \,|\, q_t = S_j], \; for \; 1 \leq j \leq N \text{ and } 1 \leq k \leq M.$$

4. The **initial state distribution** $\pi = \{\pi_i\}$, where $\pi_i = P[q_1 = S_i]$, for $1 \leq i \leq N$.

HMM Operation

Given appropriate values for N, M, A, B, and π, the HMM can be used as a generator for an observation sequence, $O = O_1, O_2, \ldots, O_T$, where each observation O_t is a symbol from V, and T is the number of observations in the sequence as follows:

1. **Choose** an initial state $q_1 = S_i$ according to the initial state distribution π;
2. **Set** $t = 1$;
3. **Choose** $O_t = v_k$ according to the observation symbol probability distribution in state S_i as determined by $b_j(k)$;
4. **Transition** to state $q_{t+1} = S_j$ according to the state transition probability distribution for state S_i as determined by $\{a_{ij}\}$;
5. **Set** $t = t + 1$;
6. **Return** to Step 3, if $t < T$; otherwise, **Terminate**.

Note: The above procedure can be used as both a generator of observations, as well as a model to explain how a given observation sequence was generated by an HMM.

Model Parameters

In short, a complete specification of a HMM requires specification of

- two model parameters (N, M),
- the observation symbols (discrete or continuous), and
- the specification of three probability measures $A, B,$ and π.

For convenience, the **compact notation** $\lambda = (A, B, \pi)$ is used to indicate the complete set of probability measures for a Hidden Markov model.

Three basic problems

Given the above form of HMM, there are three basic problems of interest that must be solved for a model to be used in real-world applications (Rabiner, 1989):

Problem 1: Given the observation sequence $O = O_1, O_2, \ldots, O_T$ and a model $\lambda = (A, B, \pi)$, how do we **efficiently** compute $P(O \mid \lambda)$, the probability of the observation sequence given the model?

This problem can also be viewed as one of scoring how well a given model matches a **given observation sequence**.

Three basic problems

Given the above form of HMM, there are three basic problems of interest that must be solved for a model to be used in real-world applications (Rabiner, 1989):

Problem 2: Given the observation sequence $O = O_1, O_2, \ldots, O_T$, and a model $\lambda = (A, B, \pi)$, how do we choose a corresponding state sequence $Q = q_1\, q_2\, \ldots\, q_T$ that is optimal in some meaningful sense (i.e., best explains the observations)?

This problem can be viewed as attempting to uncover the hidden part of a model.

Since there is no "correct" state sequence to be found, for practical situations, one usually uses an optimality criterion to solve the problem as best as possible. There can be several reasonable optimality criteria that can be imposed, depending on the intended use of the uncovered state sequence.

Three basic problems

Given the above form of HMM, there are three basic problems of interest that must be solved for a model to be used in real-world applications (Rabiner, 1989):

Problem 3: Given the observation sequence $O = O_1, O_2, \ldots, O_T$, and a model $\lambda = (A, B, \pi)$, how do we adjust the model parameters A, B, and π to maximize $P(O \mid \lambda)$?

One attempts to optimize the model parameters so as to best describe how a given observation sequence comes about.

The observation sequence used to adjust the model parameters is called a training sequence since it is used to "train" the HMM.

The **training problem** is the crucial one for most applications of HMMs, since it allows us to fit the model to the observed training data[9], creating the best model for a real phenomenon of interest.

[9] Besides the choice of the number of states and the observation symbols

Solutions to the Problems

Given a model $\lambda = (A, B, \pi)$ and an observation sequence $O = O_1, O_2, \ldots, O_T$, how can one compute the probability $P(O \mid \lambda)$ — the probability of O given the model.

The most straightforward (brute force) way of computing $P(O \mid \lambda)$ is by enumerating each and every possible state sequence of length T. This could take a while though.

Consider one such state sequence $Q = q_1 q_2 \ \ldots \ q_T$, where q_1 is the initial state. The probability of O for the state sequence Q is

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda)$$

Solutions to the Problems

Given a model $\lambda = (A, B, \pi)$ and an observation sequence $O = O_1, O_2, \ldots, O_T$, how can one compute the probability $P(O \mid \lambda)$ — the probability of O given the model.

The most straightforward (brute force) way of computing $P(O \mid \lambda)$ is by enumerating each and every possible state sequence of length T. This could take a while though.

Consider one such state sequence $Q = q_1 q_2 \ \ldots \ q_T$, where q_1 is the initial state. The probability of O for the state sequence Q is

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda)$$

Assuming statistical independence of observations, we get

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T).$$

We also know that the probability of the state sequence Q can be written as

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}.$$

The joint probability of O and Q (both events occurring simultaneously) is the product off the above two terms,

$$P(O, Q|\lambda) = P(O|Q, \lambda) \cdot P(Q| \lambda).$$

Typo in the paper!

The probability of O (given the model) is obtained by summing up this joint probability over all possible state sequences Q, that is

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda)\, P(Q|\lambda)$$

$$= \sum_{q_1, q_2, \cdots, q_T} \pi_{q_1} b_{q_1}(O_1)\, a_{q_1 q_2} b_{q_2}(O_2)$$

$$\cdots\, a_{q_{T-1} q_T} b_{q_T}(Q_T).$$

Interpretation

The interpretation of the computation assumed in the above equation for any given state sequence $Q = q_1\, q_2\, \ldots\, q_T$ is as follows.

At time $t = 1$, the process starts in state q_1 with probability π_{q_1} and generates symbol O_1 with probability $b_{q_1}(O_1)$. When the time advances to $t = 2$, a transition from q_1 to state q_2 occurs with probability $a_{q_1 q_2}$, and symbol O_2 is generated in q_2 with probability $b_{q_2}(O_2)$.

This process continues for $T - 1$ steps, when it reaches state q_T from q_{T-1} at time T with probability $a_{q_{T-1} q_T}$, and generates O_T with probability $b_{q_T}(O_T)$.

Computational Complexity

The calculation of $P(O \mid \lambda)$ involves on the order of $2T \times N^T$ calculations, since at every $t = 1, 2, \ldots, T$, there are N possible states that can be reached (hence, there are N^T possible state sequences), and for each such state sequence about $2T$ calculations are required for each term in the sum of the above equation for $P(O \mid \lambda)$.

This complexity means that the calculation is computationally infeasible, even for small values of N and T; for instance, for $N = 5$ (states) and $T = 100$ (observations), there are on the order of $2 \times 100 \times 5^{100} \approx 10^{72}$ computations.

Computational Complexity

The calculation of $P(O \mid \lambda)$ involves on the order of $2T \times N^T$ calculations, since at every $t = 1, 2, \ldots, T$, there are N possible states that can be reached (hence, there are N^T possible state sequences), and for each such state sequence about $2T$ calculations are required for each term in the sum of the above equation for $P(O \mid \lambda)$.

This complexity means that the calculation is computationally infeasible, even for small values of N and T; for instance, for $N = 5$ (states) and $T = 100$ (observations), there are on the order of $2 \times 100 \times 5^{100} \approx 10^{72}$ computations.

A more efficient procedure is required to solve Problem 1.

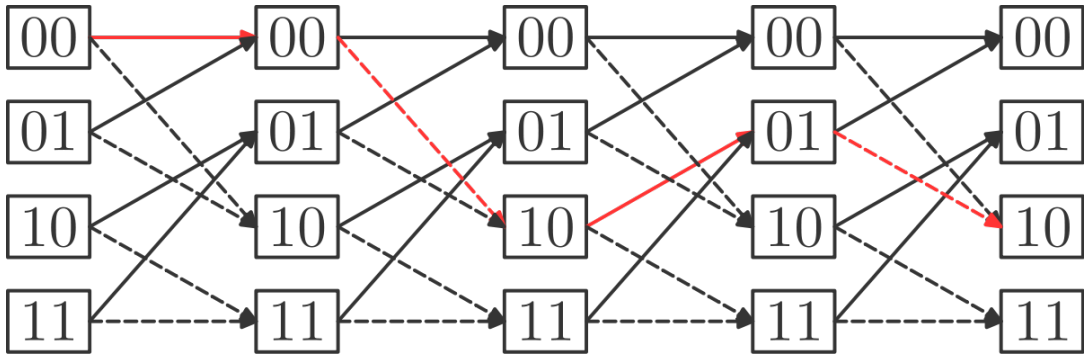
Applying the principle of dynamic programming, this problem and related problems can be handled efficiently using dedicated algorithms such as the **Forward-Backward** algorithm, the **Viberti** algorithm (*Viterbi path*: most likely hidden state sequence), and the **Baum–Welch** algorithm (find the unknown parameters λ).

Efficient Solutions to the three HMM problems

Problem 1: Likelihood computation

The Forward algorithm is an algorithm with $O(N^2T)$ complexity that

- uses a table to store intermediate values as it gradually builds up the probability of the observation sequence;
- computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence, but it does so
- efficiently by implicitly folding each of these paths into a single forward trellis[10].



Convolutional code trellis diagram (Source: Wikipedia)

[10] A trellis is a graph whose nodes are ordered into vertical slices (time), and with each node at each time connected to at least one node at an earlier and at least one node at a later time. The earliest and latest times in the trellis have only one node. Trellises are used in **encoders and decoders for communication theory and encryption**. They are also the central datatype used in Baum–Welch algorithm or the Viterbi Algorithm for Hidden Markov Models.

Problem 2: Decoding

- For any model (such as an HMM) that contains hidden variables, the task of determining which sequence of variables is ‘’the underlying source’’ of the sequence of observations is called the **decoding task**. The most common decoding algorithm for HMMs is the Viterbi algorithm.
- This algorithm finds the most likely sequence of hidden states—called the *Viterbi path*—, which results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models.
- Like the Forward algorithm, Viterbi is a dynamic programming algorithm that uses dynamic programming trellis.

Problem 3: HMM Training (find the unknown parameters)

The Forward-Backward algorithm is an inference algorithm for hidden Markov models which computes the posterior marginals of all hidden state variables given a sequence of observations $O_{1:T} = O_1, \ldots, O_T$. That is, the algorithm computes, for all hidden state variables $X_t \in X_1, \ldots, X_T$, the distribution $P(X_t \mid O_{1:T})$.

The algorithm makes use of dynamic programming to efficiently compute the values that are required to obtain these distributions in two passes. The first pass goes forward in time, while the second goes backward in time; hence the name forward–backward algorithm.

- In the first pass, the forward–backward algorithm computes a set of forward probabilities which provide, for all $t \in \{1, \ldots, T\}$, the probability of **ending up in any particular state**, given the first t observations in the sequence, i.e. $P(X_t \mid O_{1:t})$.
- In the second pass, the algorithm computes a set of backward probabilities which provide the probability of **observing the remaining observations**, given any starting point t, i.e. $P(O_{t+1:T} \mid X_t)$.
- These two sets of probability distributions can then be combined to obtain the distribution over states at any specific point in time, given the entire observation sequence.

Baum-Welch algorithm

In electrical engineering, computer science, statistical computing and bioinformatics, the Baum–Welch algorithm is a special case of the EM algorithm[11] used to find the unknown parameters of a hidden Markov model (HMM). It makes use of the forward-backward algorithm to compute the statistics for the expectation step.

[11] In statistics, latent variables (as opposed to observable variables) are **variables that are not directly observed** but are rather inferred (through a mathematical model) from other variables that are observed (directly measured). Mathematical models that aim to explain observed variables in terms of latent variables are called latent variable models.

Sources

Feller, 1968
William Feller. An Introduction to Probability Theory and Its Applications. John Wiley & Sons, 1968.

James et al., 2017
G. James, D. Witten, T. Hastie, and R. Tibshirani. An Introduction to Statistical Learning: with Applications in R, Springer, 2017.

Rabiner, 1989
Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77(2):257–286, 1989.