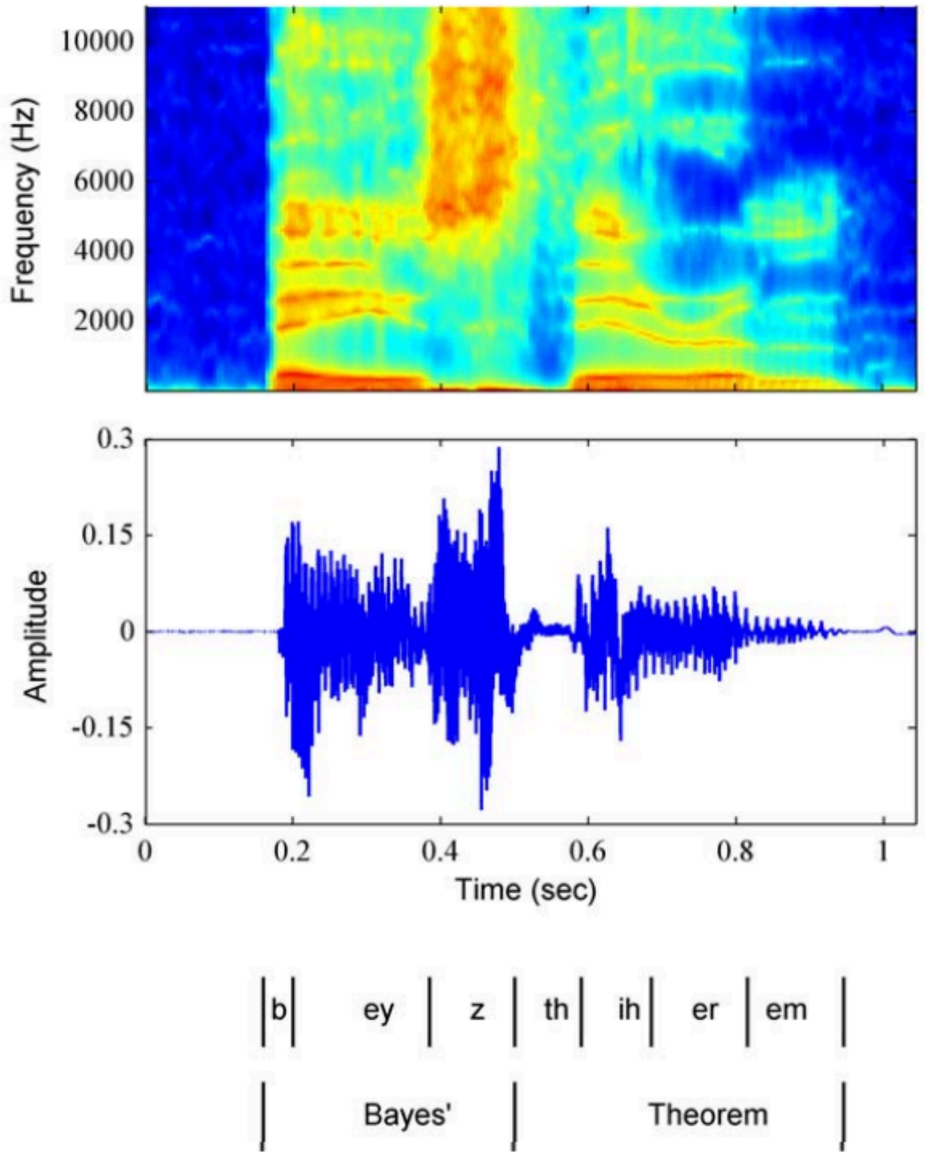
Project Objective

Identify hidden market states regimes from financial time-series data using state space machine learning models

Figure 13.1 Example of a spectrogram of the spoken words "Bayes' theorem" showing a plot of the intensity of the spectral coefficients versus time index.



Continous Data

Financial Time Series

Stationary and Nonstationary

Ref. Bishop

Financial Time Series

- We wish to be able to predict the next value in a time series (given observations of the previous values)
- Recent observations are likely to be more informative than more historical observations in predicting future values
- Impractical to consider a general dependence of future observations on all previous observations because the complexity of such a model would grow without limit as the number of observations increases
- Markov Models/ Chain : Future predictions are independent of all but the most recent observations.
- IID Assumption :

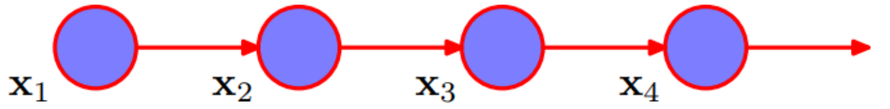
$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i).$$

- For markov chains, let us relax this assumption

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}).$$

$$p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

A first-order Markov chain of observations $\{\mathbf{x}_n\}$ in which the distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ of a particular observation $\mathbf{x}_n$ is conditioned on the value of the previous observation $\mathbf{x}_{n-1}$.



Ref. Bishop

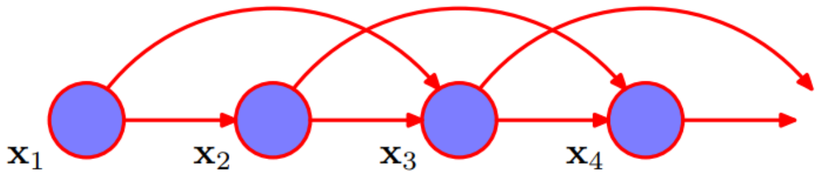
Distribution of predictions will depend only on the value of the immediately preceding observation and will be independent of all earlier observations.

$$p(X_{1:T}) = p(X_1)p(X_2|X_1)p(X_3|X_2)\ldots = p(X_1)\prod_{t=2}^{T} p(X_t|X_{t-1})$$

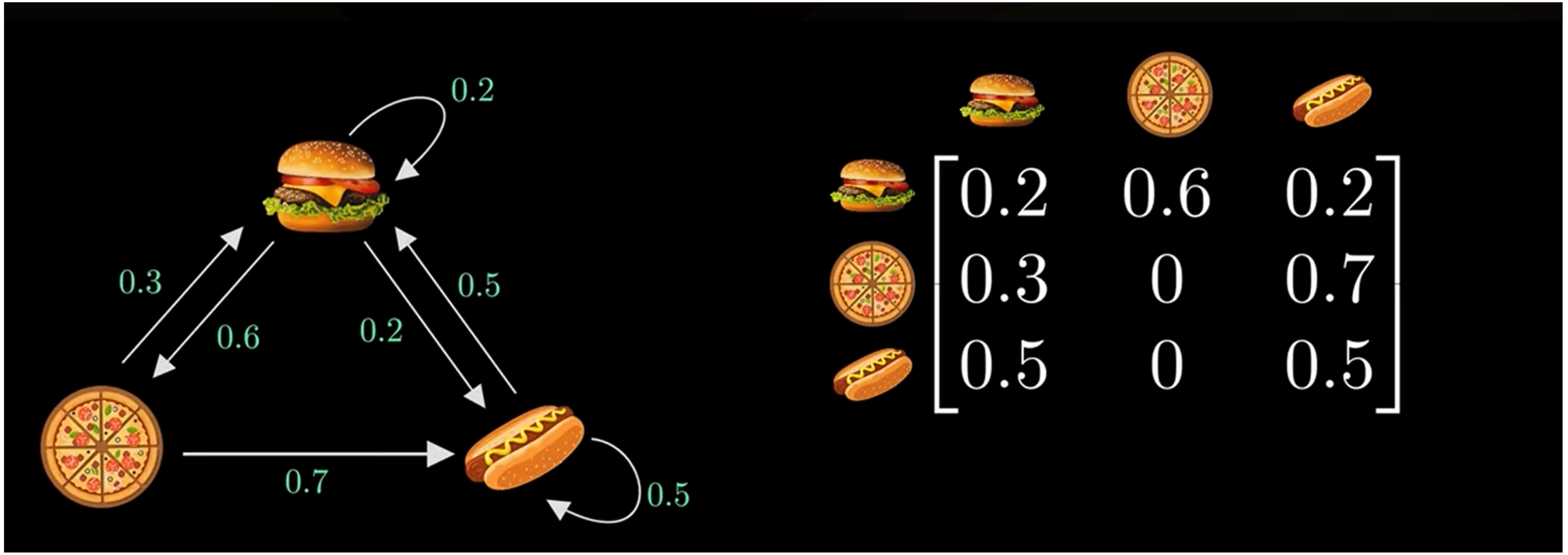
This is called a **Markov chain** or **Markov model**. first order

Ref. Murphy

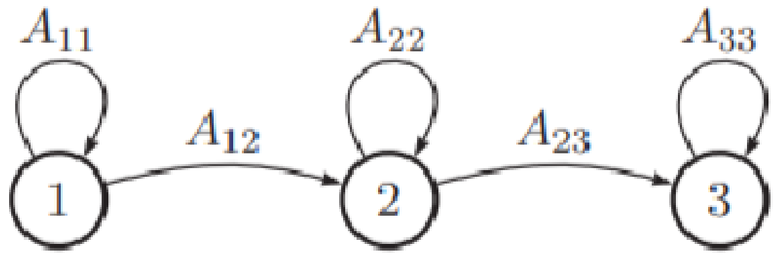
Figure 13.4 A second-order Markov chain, in which the conditional distribution of a particular observation $\mathbf{x}_n$ depends on the values of the two previous observations $\mathbf{x}_{n-1}$ and $\mathbf{x}_{n-2}$.



Model is described by directed graphs having a tree structure (no loops) for which inference can be performed efficiently using the sum-product algorithm.



Ref . Normalized Nerd Youtube Channel



$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & A_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

Ref. Murphy

The A_{ij} element of the transition matrix specifies the probability of getting from i to j in one step. The n-step transition matrix $\mathbf{A}(n)$ is defined as

$$A_{ij}(n) \triangleq p(X_{t+n} = j | X_t = i) \tag{17.4}$$

$$\mathbf{A}(n) = \mathbf{A}\,\mathbf{A}(n-1) = \mathbf{A}\,\mathbf{A}\,\mathbf{A}(n-2) = \cdots = \mathbf{A}^n$$

Thus we can simulate multiple steps of a Markov chain by "powering up" the transition matrix.

Theorem 17.2.1. *Every irreducible (singly connected), aperiodic finite state Markov chain has a limiting distribution, which is equal to $\boldsymbol{\pi}$, its unique stationary distribution.*

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{A}$$

To find the stationary distribution, we can just solve the eigenvector equation $\mathbf{A}^T\mathbf{v} = \mathbf{v}$, and then to set $\boldsymbol{\pi} = \mathbf{v}^T$, where $\mathbf{v}$ is an eigenvector with eigenvalue 1. (We can be sure such an

Application: Language modeling

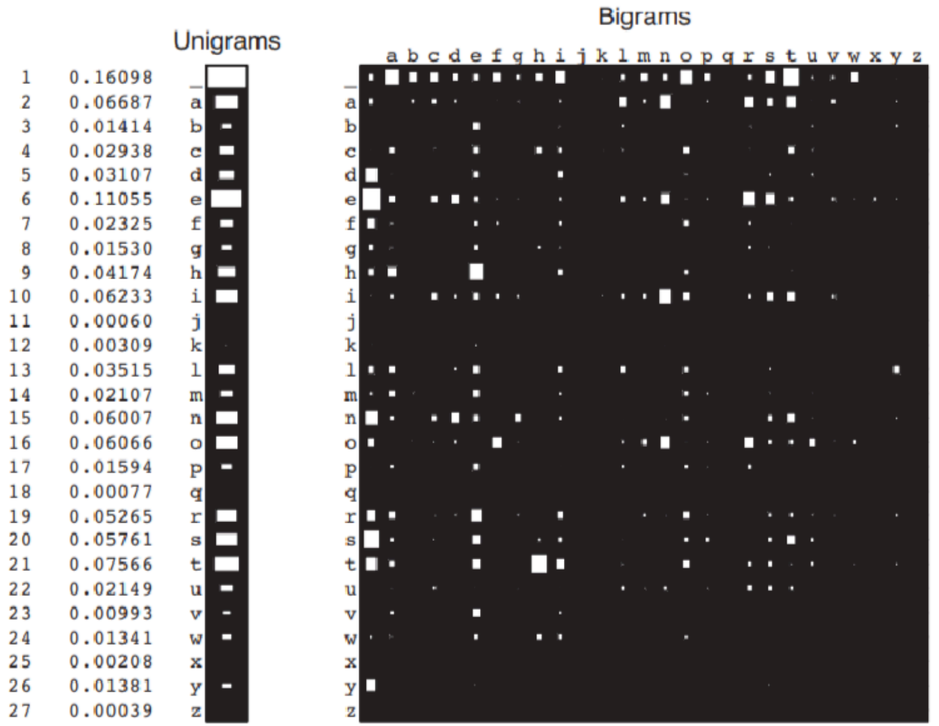
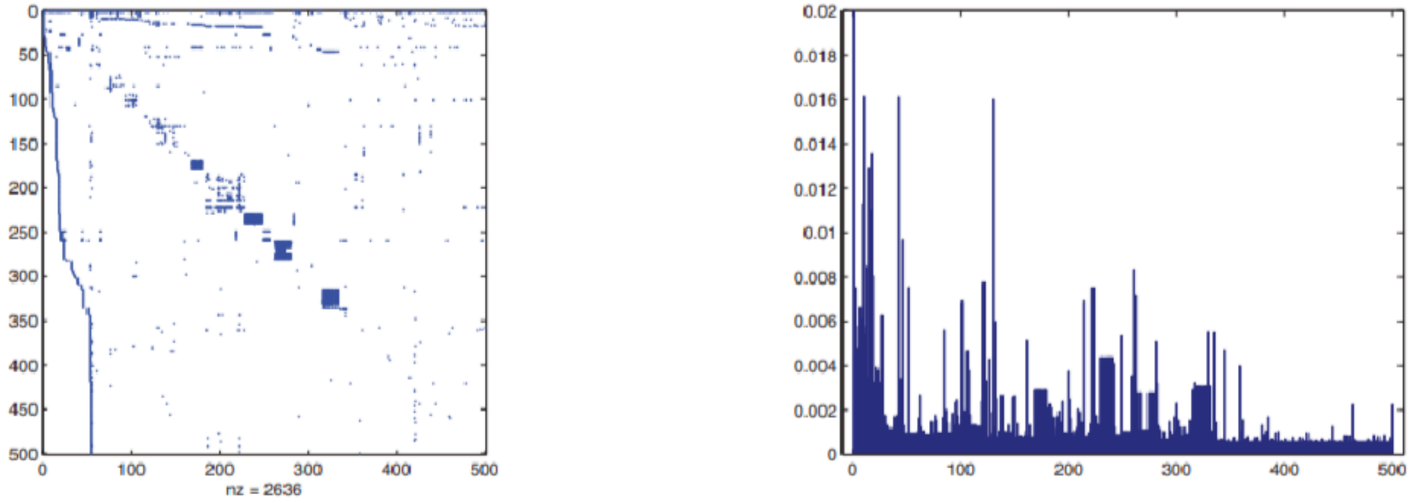


Figure 17.2 Unigram and bigram counts from Darwin's *On The Origin Of Species*. The 2D picture on the right is a Hinton diagram of the joint distribution. The size of the white squares is proportional to the value of the entry in the corresponding vector/ matrix. Based on (MacKay 2003, p22). Figure generated by `ngramPlot`.

Application: Google's PageRank algorithm for web page ranking *

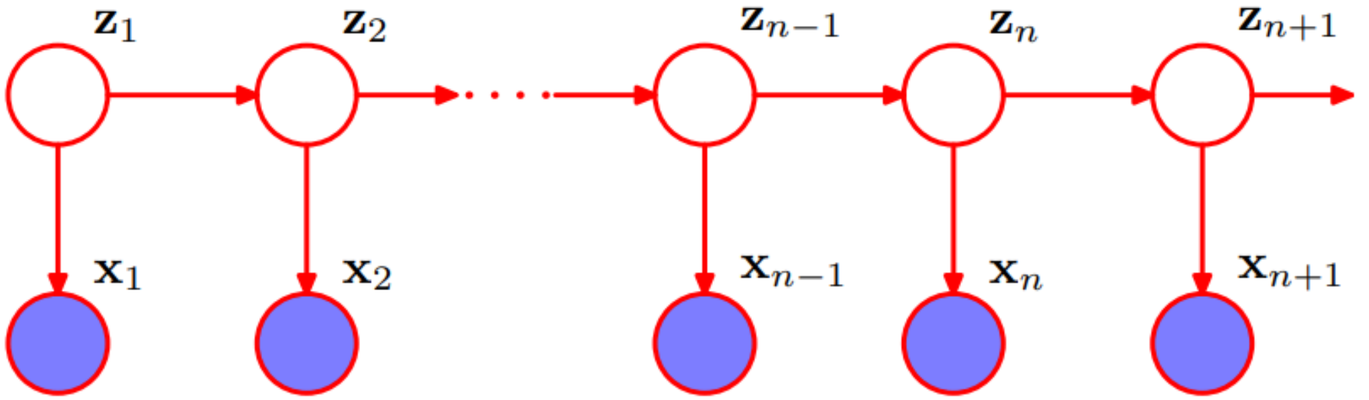


(a) (b)

Figure 17.6 (a) Web graph of 500 sites rooted at `www.harvard.edu`. (b) Corresponding page rank vector. Figure generated by `pagerankDemoPmtk`, Based on code by Cleve Moler (Moler 2004).

Hidden Markov Models

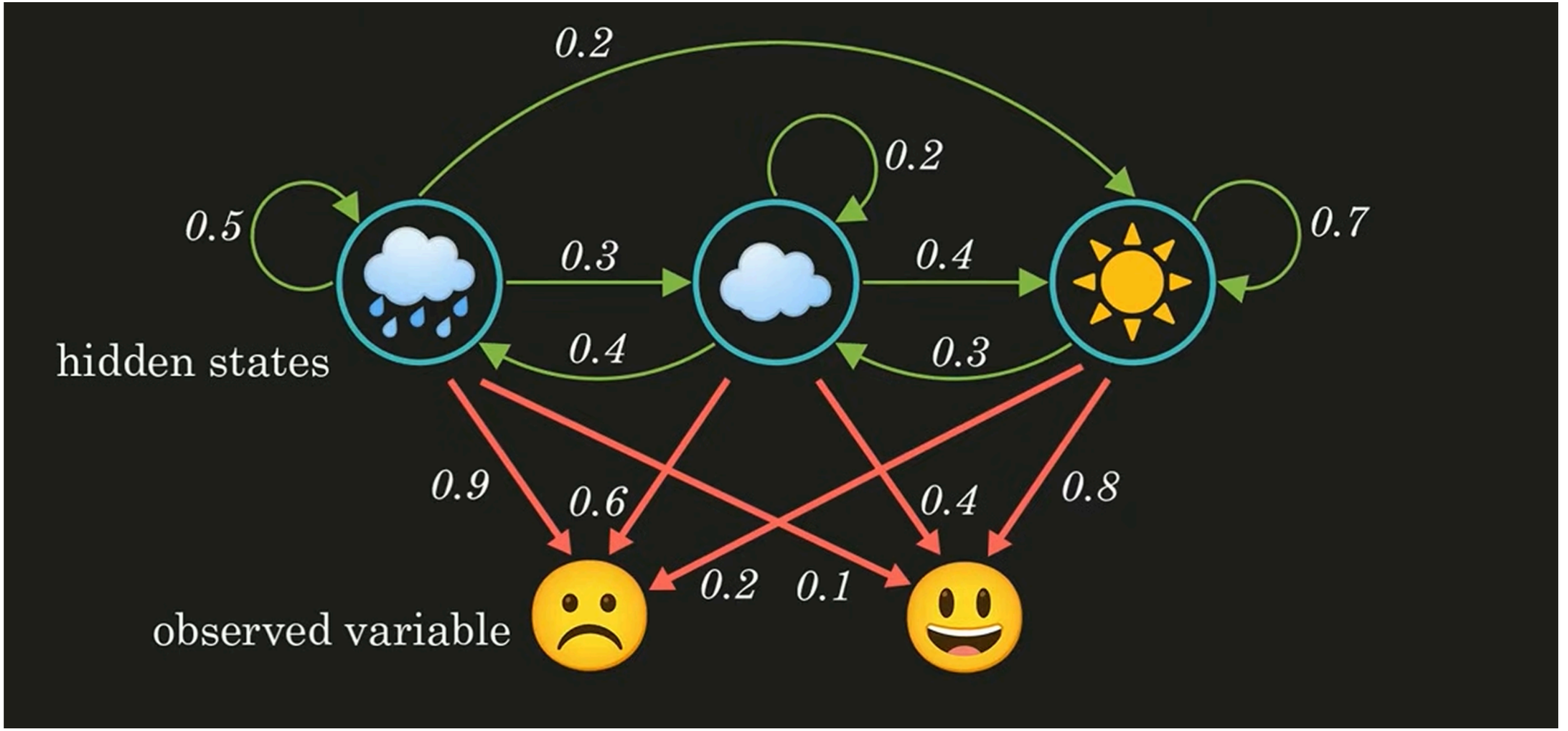
Figure 13.5 We can represent sequential data using a Markov chain of latent variables, with each observation conditioned on the state of the corresponding latent variable. This important graphical structure forms the foundation both for the hidden Markov model and for linear dynamical systems.



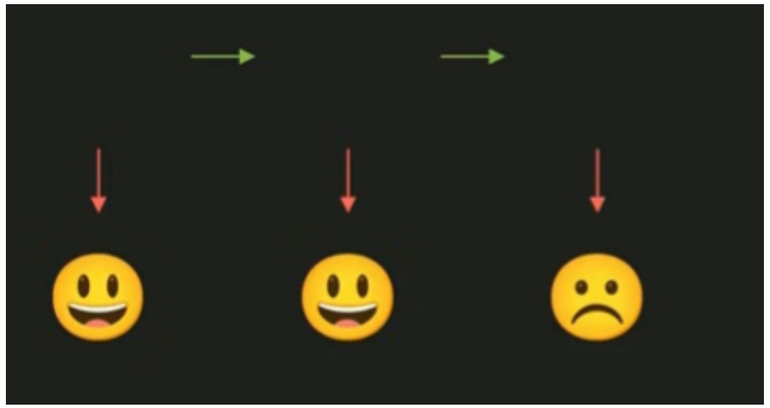
As we mentioned in Section 10.2.2, a **hidden Markov model** or **HMM** consists of a discrete-time, discrete-state Markov chain, with hidden states $z_t \in \{1, \ldots, K\}$, plus an **observation** model $p(\mathbf{x}_t|z_t)$. The corresponding joint distribution has the form

$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{z}_{1:T})p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) = \left[p(z_1)\prod_{t=2}^{T} p(z_t|z_{t-1})\right]\left[\prod_{t=1}^{T} p(\mathbf{x}_t|z_t)\right]$$

Hidden Markov Models : Hidden Markov Chain + Observed Variables



Observed Variables



Inference in HMMs

What is the most likely weather sequence for the observed mood sequence?

Let the hidden state be represented by X and the observed variable by Y. The goal of this inference problem is:

$$\underset{X=X_1,X_2,\ldots,X_n}{\arg\max} \; P\big(X = X_1, X_2, \ldots, X_n \mid Y = Y_1, Y_2, \ldots, Y_n\big) \tag{1}$$

Using Bayes' theorem (and HMM assumptions), this becomes:

$$\underset{X=X_1,X_2,\ldots,X_n}{\arg\max} \; \prod_{i=1}^{n} P(Y_i \mid X_i)\, P(X_i \mid X_{i-1}) \tag{2}$$

Forwards Algorithm (Dyanamic Programming Approach)

- N : Number of hidden states
- X_i : i-th hidden state, $i = 0, 1, \ldots, N-1$
- Y^t : Observation at time t
- $\pi[i] = P(X_i \text{ at } t = 1)$: Initial state probability
- $P(X_i \mid X_j)$: Transition probability from state X_j to X_i
- $P(Y^t \mid X_i)$: Emission probability of observation Y^t from state X_i
- $\alpha_t(X_i) = P(Y^1, \ldots, Y^t,\ X_i \text{ at time } t)$: Forward variable

Initialization:

$$\alpha_1(X_i) = \pi[i]\, P(Y^1 \mid X_i)$$

Recursion:

$$\alpha_t(X_i) = \sum_{j=0}^{N-1} \alpha_{t-1}(X_j)\, P(X_i \mid X_j)\, P(Y^t \mid X_i)$$

Termination (Likelihood):

$$P(Y^1, Y^2, \ldots, Y^T) = \sum_{i=0}^{N-1} \alpha_T(X_i)$$

The Flaw of Static Distributions & Introduction to Latent Variables

• Static Distributions Underestimate Risk: Modelling stock returns with a naive, static normal distribution is highly ineffective in practice because it fails to capture the true dynamics of the market.

• The Reality of Time-Varying Distributions: In reality, the true data-generating distribution that produces stock returns is not static; its shape, mean, and variance change continuously over time.

• Driven by Latent Variables: These time-varying distributions are heavily influenced by unobservable underlying factors known as "latent variables".

• Volatility as a Latent Variable: A primary example of a latent variable is market volatility.

• Defining Discrete States: To correct the assumptions of static distributions, we can use a Markov chain to model the latent variable (volatility) by breaking it into discrete states, such as low, mid, and high volatility regimes.

• Regime-Dependent Distributions: Instead of lumping all returns together, the Markov chain assigns a conditional distribution to each specific state. For example, the distribution during a low-volatility state will look drastically different and have a different expected return than one during a high-volatility state.

• Better Expected Returns and Likelihoods: By matching the market's current volatility state to the correct conditional distribution, the model accurately adjusts for the wider spread of returns during uncertain periods.

The Shift to Hidden Markov Models (HMMs)

• The Limitation of Deterministic Variables: Explicitly defining a single latent variable like volatility into a Markov chain is a great first step, but it omits the reality that return distributions are driven by multiple interacting latent processes, such as volatility, trend, and momentum.

• Compressing Complex Market Dynamics: Instead of manually defining strict, deterministic rules for every single latent factor, Hidden Markov Models (HMMs) compress all of these underlying latent processes into a finite number of unspecified, hidden states.

• Learning Directly From Data: HMMs remove the need to proxy or pre-define the criteria for these regimes. Through algorithms like Baum-Welch, HMMs automatically infer the hidden states and optimal transition probabilities directly from historical data.

13.1 The "Chicken-and-Egg" Problem

As we mentioned in previous sections, a **hidden Markov model** or **HMM** consists of a discrete-time, discrete-state Markov chain. The fundamental challenge in fitting this model to financial data is that we only observe the sequence of daily returns $\mathbf{x}_t$, while the regime labels $z_t \in \{1, \ldots, K\}$ and their associated emission parameters (e.g., μ, σ^2) remain latent.

If the regime labels were known, parameter estimation would reduce to a simple empirical frequency calculation. Conversely, known parameters would allow for straightforward state inference using Bayes' theorem. To resolve this mutual dependency, we utilize the **Baum-Welch algorithm**, which is a specific application of the broader Expectation-Maximization (EM) framework. We initialize the model with a prior distribution of parameters $\boldsymbol{\lambda}$, and iteratively refine them.

13.2 The E-Step and M-Step Formulations

The Baum-Welch algorithm alternates between inferring the latent states and updating the model parameters. In the Expectation step (E-step), we calculate the posterior probability of being in state j at time t, denoted as $\gamma_t(j)$, using the forward (α) and backward (β) variables:

$$\gamma_t(j) = p(z_t = j|\mathbf{X}, \boldsymbol{\lambda}) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{k=1}^{K} \alpha_t(k)\beta_t(k)} \tag{13.1}$$

In the Maximization step (M-step), these posteriors act as adaptive, context-dependent weights. For example, the updated mean $\boldsymbol{\mu}_j$ for a Gaussian emission model is given by the weighted sum of the observations:

$$\boldsymbol{\mu}_j = \frac{\sum_{t=1}^{T} \gamma_t(j)\mathbf{x}_t}{\sum_{t=1}^{T} \gamma_t(j)} \tag{13.2}$$

13.2.1 Intuition: The Expectation Step

The term $\gamma_t(j)$ acts as a "soft label" or fractional count. Rather than assigning a definitive regime to each observation via a hard assignment (e.g., asserting $z_t = 1$), the Bayesian approach leverages this context-dependent weight.

The forward pass computes the joint probability $p(\mathbf{x}_{1:t}, z_t = j)$ of the historical context up to time t. Symmetrically, the backward pass computes the conditional probability of future observations $p(\mathbf{x}_{t+1:T}|z_t = j)$. By taking the product of these two passes, the algorithm effectively incorporates the full sequence context to form a smooth probability distribution over the possible states at every single time step t.

13.2.2 Intuition: The Maximization Step

The only remaining question is: how do we appropriately update the transition and emission matrices using these fractional counts? The M-step resolves this by computing a probability-weighted average.

Equation 13.2 demonstrates that the new mean is derived by taking the proportional contribution of every observation $\mathbf{x}_t$, strictly weighted by the probability that the system was in state j at that time. Similarly, the transition matrix updates combine the expected number of transitions from state i to j divided by the expected time spent in state i. This process mathematically guarantees an increase in the marginal likelihood of the sequence with each iteration.

Dynamic Asset Allocation: If the HMM detects a shift from a low-volatility bull regime to a high-volatility bear regime, a portfolio manager might automatically reduce equity exposure and increase allocations to cash or bonds.

Strategy Switching: Certain algorithmic strategies (like mean-reversion) work exceptionally well in sideways markets but fail catastrophically in strong trending markets. An HMM acts as an "on/off" switch, deploying specific algorithms only when their optimal regime is detected.

Risk Management: By understanding the emission probabilities of the current regime, risk managers can calculate more accurate Value at Risk (VaR) metrics, recognizing that the potential for extreme losses is much higher in certain hidden states than in others.

What Next?

- Forward-Backward Algorithm:
  - Forward pass (α) calculates P(observations up to t, state i at t)
  - Backward pass (β) calculates P(future observations given state i at t)
  - Combine to get state probabilities γ and transition probabilities ξ
- Viterbi Algorithm
- Baum-Welch Algorithm (EM):
  - E-step: Use forward-backward to calculate expected state occupancies
  - M-step: Update model parameters (transitions, emissions) to maximize likelihood
  - Iterate until convergence
- Live Kalman Filter Model with Regime Dynamics (MCs/HMMs)
- Hidden semi-Markov models (HSMMs)
- Use Bayesian offline methods (reversible jump MCMC, dynamic programming) to identify historical structural breaks in volatility, returns, or macro covariates
- Include changepoint posterior distributions, regime persistence diagnostics, MS-VAR impulse response functions, and performance metrics of regime-based strategies.