Lecture 5
Context-dependent Classification and Markov Models

Context Dependent Classification

➔ **Remember: Bayes rule**

$$P(\omega_i|\underline{x}) > P(\omega_j|\underline{x}), \ \forall j \neq i$$

➔ **Here: The class to which a feature vector belongs depends on:**

- ⇨ Its own value
- ⇨ The values of the other features
- ⇨ An existing relation among the various classes

- ➔ **This interrelation demands the classification to be performed simultaneously for all available feature vectors**

- ➔ **But.. what happens if the training vectors $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$ occur in sequence, one after the other?**
  - ⇨ we will refer to them as **observations**

➔The Context Dependent Bayesian Classifier

⇨ Let $X : \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$

⇨ Let $\omega_i, \; i = 1,2,\ldots,M$

⇨ Let Ω_i be a sequence of classes, that is

$$\Omega_i : \omega_{i1} \; \omega_{i2} \; \ldots \; \omega_{iN}$$

There are M^N of those

⇨ Thus, the Bayesian rule can equivalently be stated as

$$\boxed{X \rightarrow \Omega_i : \; P(\Omega_i | X) > P(\Omega_j | X) \;\; \forall i \neq j, \;\; i, j = 1,2,\ldots,M^N}$$

➔Markov Chain Models (for class dependence)

$$P(\omega_{i_k} | \omega_{i_{k-1}}, \omega_{i_{k-2}}, \ldots, \omega_{i_1}) = P(\omega_{i_k} | \omega_{i_{k-1}})$$

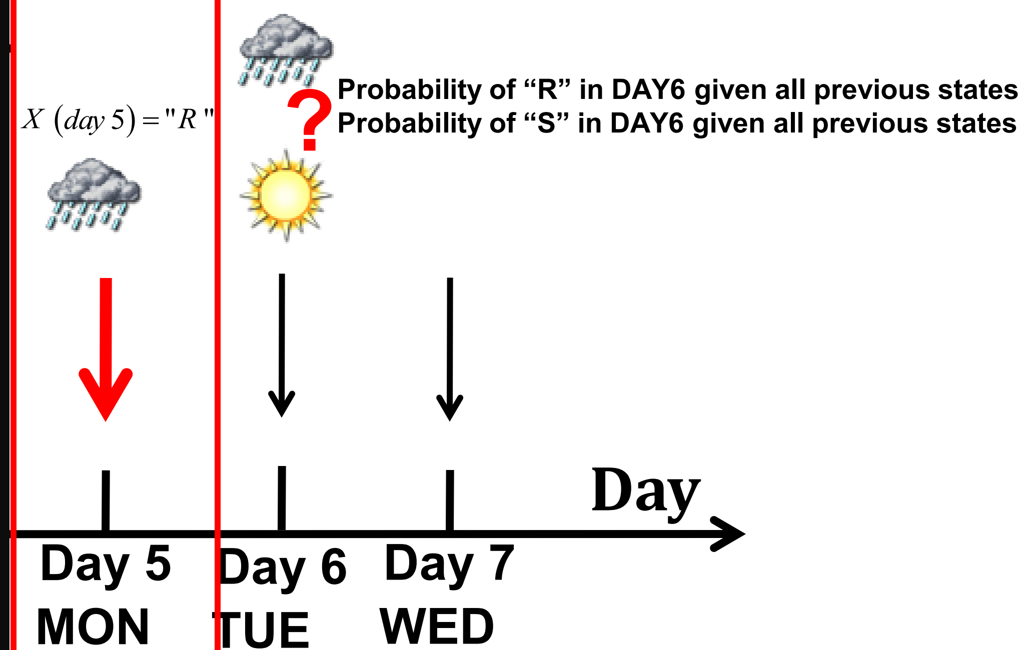
Other memory models are possible!!

What is the “Markov Property”?

$$\Pr\{X_{DAY\ 6} = "S" \mid X_{DAY\ 5} = "R", X_{DAY\ 4} = "S", \ldots, X_{DAY\ 1} = "S"\} =$$
$$\Pr\{X_{DAY\ 6} = "S" \mid X_{DAY\ 5} = "R"\}$$

PAST EVENTS | **NOW** | **FUTURE EVENTS**

[redacted]



Markov Property: The probability that it will be (FUTURE) SUNNY in DAY 6 given that it is RAINNY in DAY 5 (NOW) is independent from PAST EVENTS

Markov Process

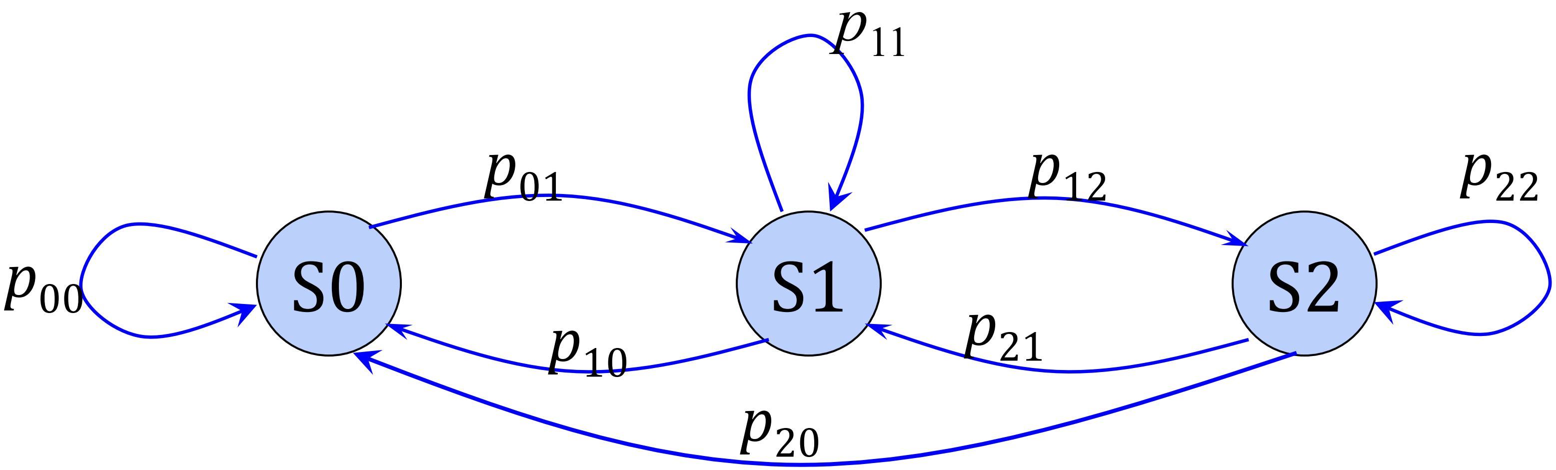
- ➔ **The temporal evolution of classes is ‘correlated’ and depends on class of the previous observation**
  - ⇨ Since decision classes are discrete, i.e. the output can assume a finite number of results, the process is called a ‘chain’
- ➔ **Discrete time Markov Chain:**
  - ⇨ Evolution at discrete time instants (when new observations are available)
  - ⇨ Values in a finite set

Definitions

- **Markov process describing class evolution:**
  - The future of the process does not depend on the whole past, but only on the present
  - Let X(k) be the variable representing the class process at time k and x(k) the value of the class at the same time k

$$\Pr\{X_{k+1}=x_{k+1} \mid X_k=x_k,\ldots,X_0=x_0\}=\Pr\{X_{k+1}=x_{k+1} \mid X_k=x_k\}$$

General Model of a Markov Chain



$S = \{S0, S1, S2\}$ **State Space**

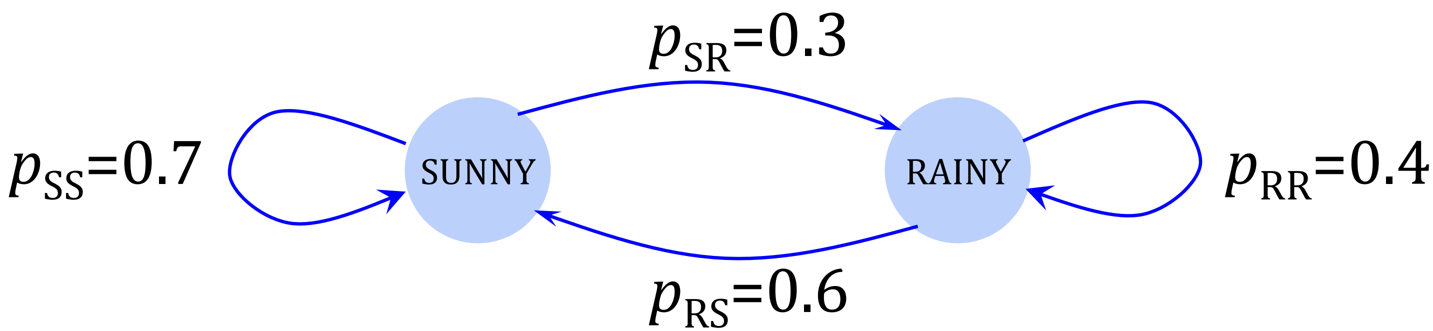
i or Si **State i**

Discrete Time (Slotted Time)

$$time = \{t_0, t_1, t_2, \ldots, t_k\}$$
$$= \{0, 1, 2, \ldots, k\}$$

Transition Probability from State Si to State Sj p_{ij}

Example of a Markov Process
A very simple weather model



State Space

$$S = \{SUNNY, RAINY\}$$

1) If today is Sunny, What is the probability that to have a SUNNY weather after 1 week?

2) If today is rainy, what is the probability to stay rainy for 3 days?

Chapman-Kolmogorov Equations

- ➔ We define the one-step transition probabilities at the instant k as

$$p_{ij}(k) = \Pr\{X_{k+1} = j \mid X_k = i\}$$

- ➔ Necessary Condition: being N the total number of state, for all states i, instants k, and all feasible transitions from state i we have:

$$\sum_{j=1}^{N} p_{\mathrm{ik}}(\mathrm{k}) = 1$$

- ➔ What is the transition probability at n-steps?

$$p_{ij}(k, k+n) = \Pr\{X_{k+n} = j \mid X_k = i\}$$

Chapman-Kolmogorov Equations

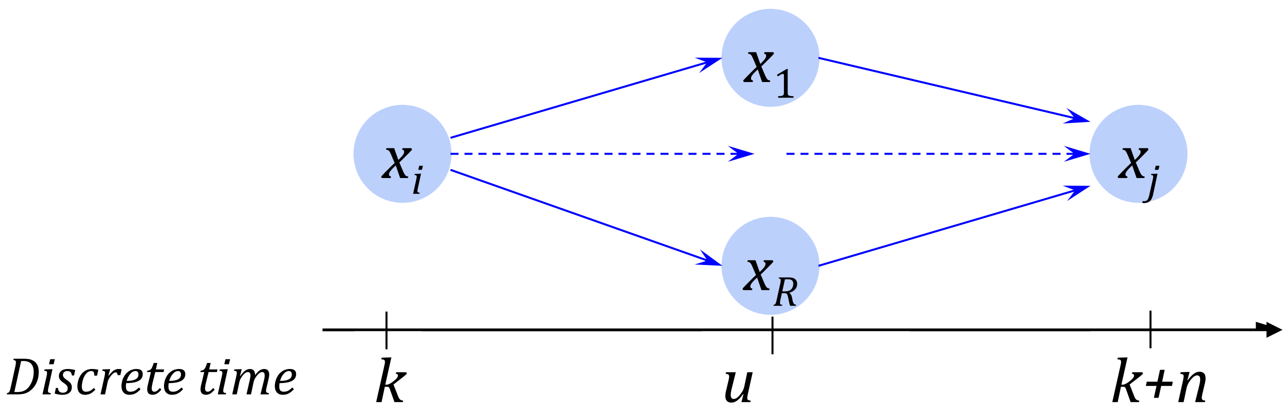
Using **Law of Total Probability**

$$\Pr(A) = \sum_n \Pr(A \mid B_n)\Pr(B_n).$$

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

$$p_{ij}(k, k+n) = \Pr\{X_{k+n} = j \mid X_k = i\}$$

$$= \sum_{r=1}^{R} \Pr\{X_{k+n} = j \mid X_u = r, X_k = i\} \Pr\{X_u = r \mid X_k = i\}$$



Chapman-Kolmogorov Equations

Using the memoryless property of Markov chains

$$\Pr\{X_{k+n} = j \mid X_u = r, X_k = i\} = \Pr\{X_{k+n} = j \mid X_u = r\}$$

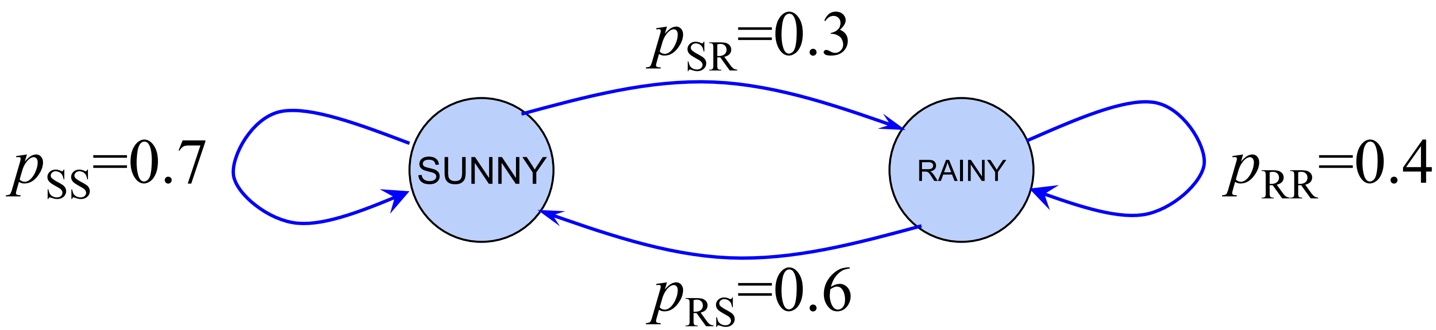
We obtain the Chapman-Kolmogorov Equation

$$p_{ij}(k, k+n) = \Pr\{X_{k+n} = j \mid X_k = i\}$$

$$= \sum_{r=1}^{R} \Pr\{X_{k+n} = j \mid X_u = r\} \Pr\{X_u = r \mid X_k = i\}$$

$$\boxed{p_{ij}(k, k+n) = \sum_{r=1}^{R} p_{ir}(k, u)\, p_{rj}(u, k+n), \qquad k \le u \le k+n}$$

Example on the simple weather model



What is the probability that the weather is rainy on **day 3** knowing that it is sunny on **day 1**?

$$p_{sunny->rainy}(day\,1, day\,3) = p_{sunny->sunny}(day\,1, say\,2)\cdot p_{sunny->rainy}(day\,2, day\,3) + p_{sunny->rainy}(day\,1, day\,2)\cdot p_{rainy->rainy}(day\,2, day\,3)$$

$$p_{sunny->rainy}(day\,1, day\,3) = p_{ss}(day\,1, say\,2)\cdot p_{sr}(day\,2, day\,3) + p_{sr}(day\,1, day\,2)\cdot p_{rr}(day\,2, day\,3)$$

$$p_{sunny->rainy}(day\,1, day\,3) = p_{ss}\cdot p_{sr} + p_{sr}\cdot p_{rr}$$

$$= 0.7\cdot 0.3 + 0.3\cdot 0.4 = 0.21 + 0.12 = 0.33$$

Transition Matrix

➔Define the n-step transition matrix as

$$\mathbf{H}(k, k+n) = \left[p_{ij}(k, k+n) \right]$$

➔We can re-write the Chapman-Kolmogorov Equation as follows:

$$\mathbf{H}(k, k+n) = \mathbf{H}(k, u)\mathbf{H}(u, k+n)$$

➔Choosing u=k+n-1:

$$\begin{aligned}\mathbf{H}(k, k+n) &= \mathbf{H}(k, k+n-1)\mathbf{H}(k+n-1, k+n) \\ &= \mathbf{H}(k, k+n-1)\mathbf{P}(k+n-1)\end{aligned}$$

Forward equation

Transition Matrix

➔Choosing u=k+1:

$$\mathbf{H}(k,k+n)=\mathbf{H}(k,k+1)\mathbf{H}(k+1,k+n)$$
$$=\mathbf{P}(k)\mathbf{H}(k+1,k+n)$$

Backward equation

➔P(k) is called transition matrix

➔Markov processes are said homogeneous if

$$p_{ij}=\Pr\{X_{k+1}=j\mid X_k=i\}=\Pr\{X_k=j\mid X_{k-1}=i\}$$

⇨For this processes, P(k)=P

State Probabilities

- **An interesting quantity we are usually interested in is the probability of finding the chain at various states, i.e., we define**

$$\pi_i(k) \equiv \Pr\{X_k = i\}$$

For all possible states, we define the vector

$$\boldsymbol{\pi}(k) = [\pi_0(k), \pi_1(k) \ldots]$$

Using total probability we can write

$$\pi_i(k) = \sum_j \Pr\{X_k = i \mid X_{k-1} = j\} \Pr\{X_{k-1} = j\}$$

$$= \sum_j p_{ij}(k) \pi_j(k-1)$$

In vector form, one can write $\boldsymbol{\pi}(k) = \boldsymbol{\pi}(k-1)\mathbf{P}(k)$

Or, for the <u>homogeneous case</u> $\boldsymbol{\pi}(k) = \boldsymbol{\pi}(k-1)\mathbf{P}$

Limiting and Stationary Distribution for homogeneous chains

➔ **State probabilities (time-dependent)**

$$\pi_j^n = P\{X_n = j\}, \qquad \pi^n = (\pi_0^n, \pi_1^n, \ldots)$$

In matrix form:

$$P\{X_n = j\} = \sum_{i=0}^{\infty} P\{X_{n-1} = i\} P\{X_n = j \mid X_{n-1} = i\} \Rightarrow \pi_j^n = \sum_{i=0}^{\infty} \pi_i^{n-1} P_{ij}$$

➔ **If time-dependent distribution converges to a limit, which does not depend on the initial state..**

$$\pi^n = \pi^{n-1} P = \pi^{n-2} P^2 = \ldots = \pi^0 P^n$$

➔ **π is called the *limiting distribution*** $\quad \pi = \lim_{n \to \infty} \pi^n$

- ⇨ Existence depends on the structure of Markov chain

➔ **A distribution is said a stationary distribution if** $\quad \pi = \pi P$

- ⇨ For finite state chains, if limiting distribution exists, limiting= stationary
- ⇨ For infinite state chains, limiting= stationary if the chain is irreducible and aperiodic

State Probabilities Example

➔ **Suppose that**

$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.35 & 0.5 & 0.15 \\ 0.245 & 0.455 & 0.3 \end{bmatrix} \qquad \text{with} \qquad \boldsymbol{\pi}(0) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Find $\boldsymbol{\pi}$(k) for k=1,2,…

$$\boldsymbol{\pi}(1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.35 & 0.5 & 0.15 \\ 0.245 & 0.455 & 0.3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 \end{bmatrix}$$

Transient behavior of the system

In general, the transient behavior is obtained by solving the difference equation $\boldsymbol{\pi}(k) = \boldsymbol{\pi}(k-1)\mathbf{P}$

Exercise

- ➔ **Consider P=[0 1; 1 0]. Does this transition matrix has a limiting distribution? And a stationary one?**
- ➔ **Consider now the same problem for the following P:**

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

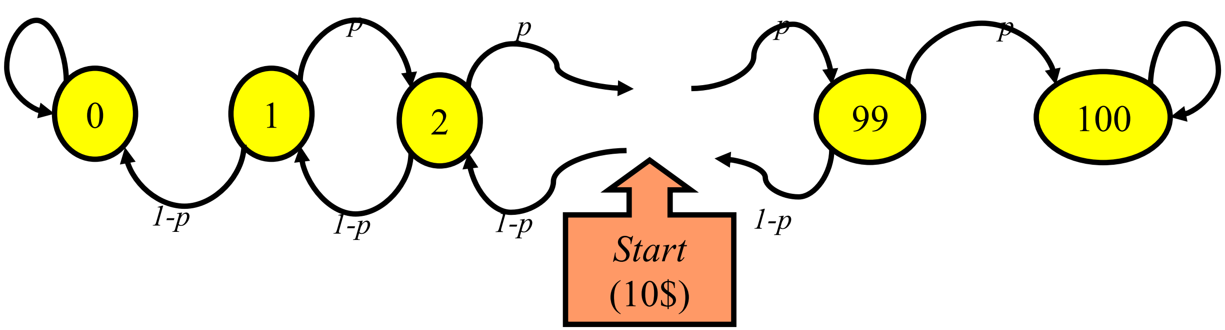
The period of a generic state j is the greatest common divisor of a set of integers n, such that $P_{jj}^{n}>0$

Balance equations

- **For an ergodic Markov chain, i.e. an homogeneous chain with stationary distribution:**
  - Rate of transitions leaving a state is equal to the rate of transitions entering a state
  - Why? Intuition: j visited infinitely often; for each transition out of j there must be a subsequent transition into j with probability 1

The Gambler's example

- ➔ **Gambler starts with $10**
- ➔ **At each play we have one of the following:**
  - ⇨ Gambler wins $1 with probability p
  - ⇨ Gambler looses $1 with probability $1-p$
- ➔ **Game ends when gambler goes broke, or gains a fortune of $100**
  - ⇨ (Both 0 and 100 are absorbing states)

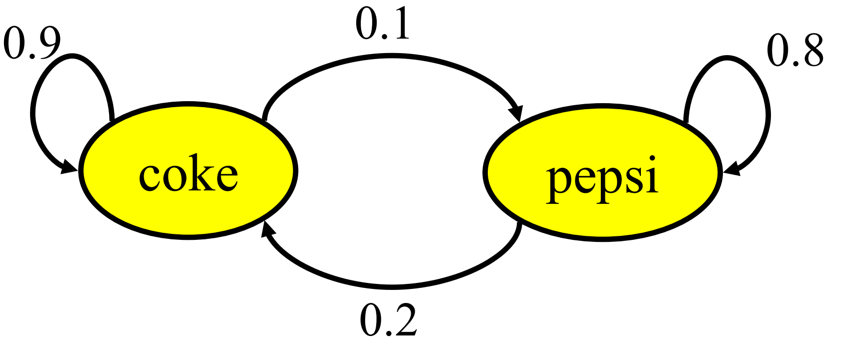


Coke vs. Pepsi example

- ➔ **Given that a person's last cola purchase was Coke, there is a 90% chance that his next cola purchase will also be Coke.**
- ➔ **If a person's last cola purchase was Pepsi, there is an 80% chance that his next cola purchase will also be Pepsi.**

transition matrix:

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$



Coke vs. Pepsi Example (cont)

Given that a person is currently a Pepsi purchaser, what is the probability that he will purchase Coke two purchases from now?

Pr[Pepsi→?→Coke] =

Pr[Pepsi→Coke→Coke] + Pr[Pepsi→ Pepsi →Coke] =

$0.2 * 0.9 + 0.8 * 0.2 = 0.34$

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix}$$

Pepsi → ?   ? → Coke

Coke vs. Pepsi Example (cont)

Given that a person is currently a Coke purchaser, what is the probability that he will purchase Pepsi **three** purchases from now?

$$P^3 = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix} = \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix}$$

Coke vs. Pepsi Example (cont)

1) Assume each person makes one cola purchase per week

2) Suppose 60% of all people now drink Coke, and 40% drink Pepsi

3) What fraction of people will be drinking Coke three weeks from now?

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \qquad P^3 = \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix}$$

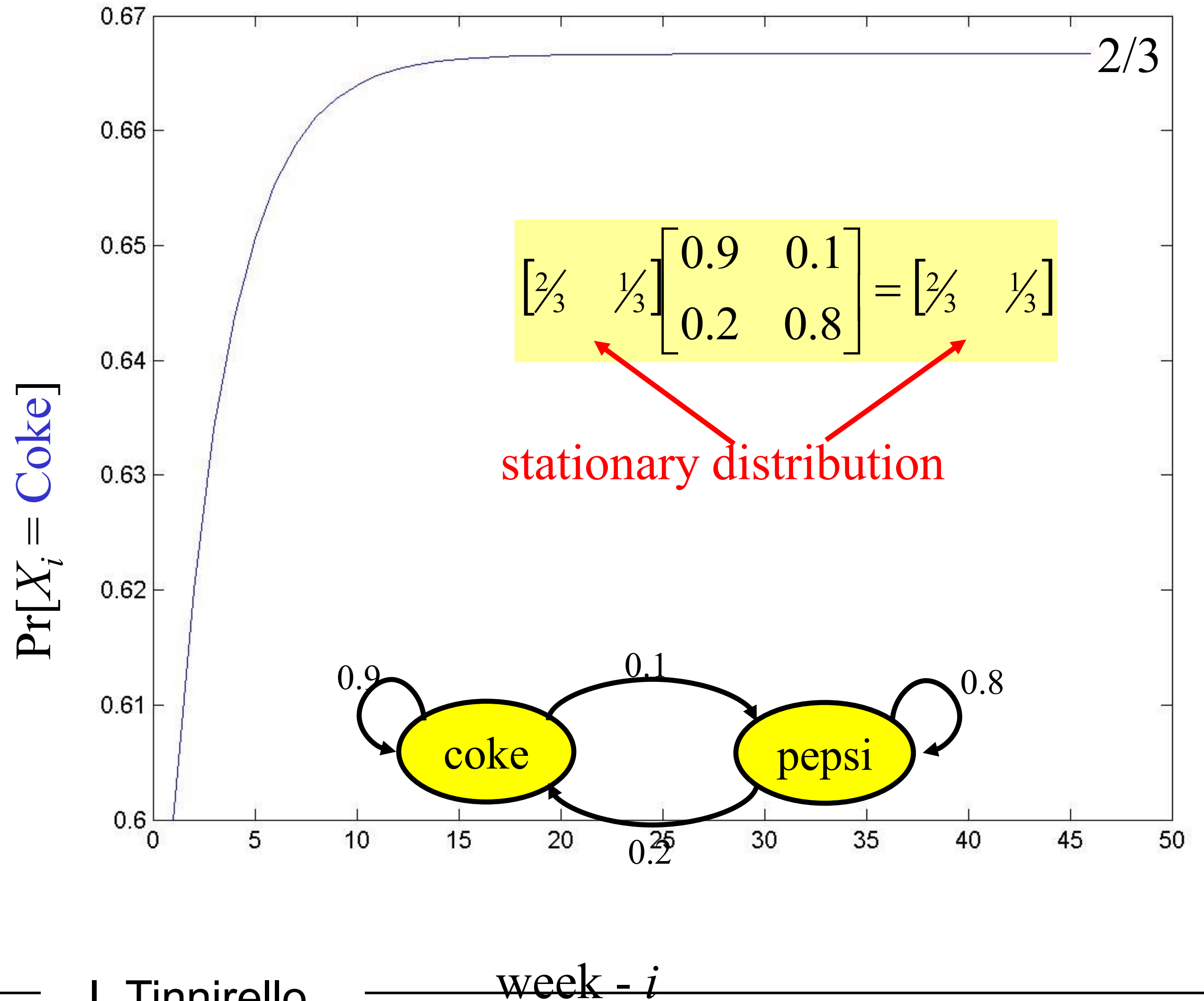
$$\Pr[X_3=\text{Coke}] = 0.6 * 0.781 + 0.4 * 0.438 = 0.6438$$

Q_i - the distribution in week i

$Q_0=(0.6,0.4)$ - initial distribution

$Q_3= Q_0 * P^3 =(0.6438,0.3562)$

Coke vs. Pepsi Example (cont)



Markov Chain Application Example: PageRank

Citation Analysis

- ➔ **Citation frequency**
- ➔ **Bibliographic coupling frequency**
  - ⇨ Articles that co-cite the same articles are related
- ➔ **Citation indexing**
  - ⇨ Who is this author cited by? (Garfield 1972)
  - ⇨ Common solution for ranking documents..
    - ➔ How many pages link to my page?
  - ⇨ But.. Are all sites of equal relevance? A link by yahoo is equal to a link by joe's web page?
- ➔ **Pagerank preview: Pinsker and Narin '60s**
  - ⇨ Asked: which journals are authoritative?
  - ⇨ Can we use the number of pointers to weight a pointe?

The web isn't scholarly citation

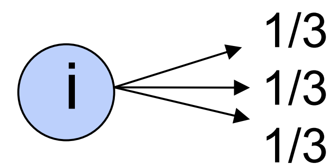
- ➔ **Millions of participants, each with self interests**
- ➔ **Spamming is widespread**
- ➔ **Once search engines began to use links for ranking (roughly 1998), link spam grew**
  - ⇨ You can join a *link farm* – a group of websites that heavily link to one another

Google solution

- ➔**Define page rank recursively**
  - ⇨A page has high rank if the sum of the ranks of its backlinks is high
- ➔**But.. Why Markov chains?**
  - ⇨We can relate page links to probabilities!!!

Pagerank scoring

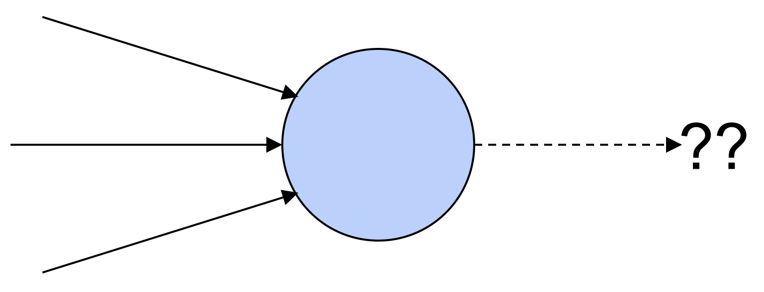
- **Imagine a user doing a random walk on web pages:**
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
    - $P_{ix}=1/k$ toward each destination x in a set of k outgoing links



- **"In the long run" each page has a long-term visit rate - use this as the page's score.**
  - The probability to reach a page j from all the n pages in the world can be expressed as: $\pi_j = \sum_{i=1}^{n} \pi_i P_{ij}.$

Not good enough with real web!

- **The web is full of dead-ends.**
  - Random walk can get stuck in dead-ends.
  - Makes no sense to talk about long-term visit rates.



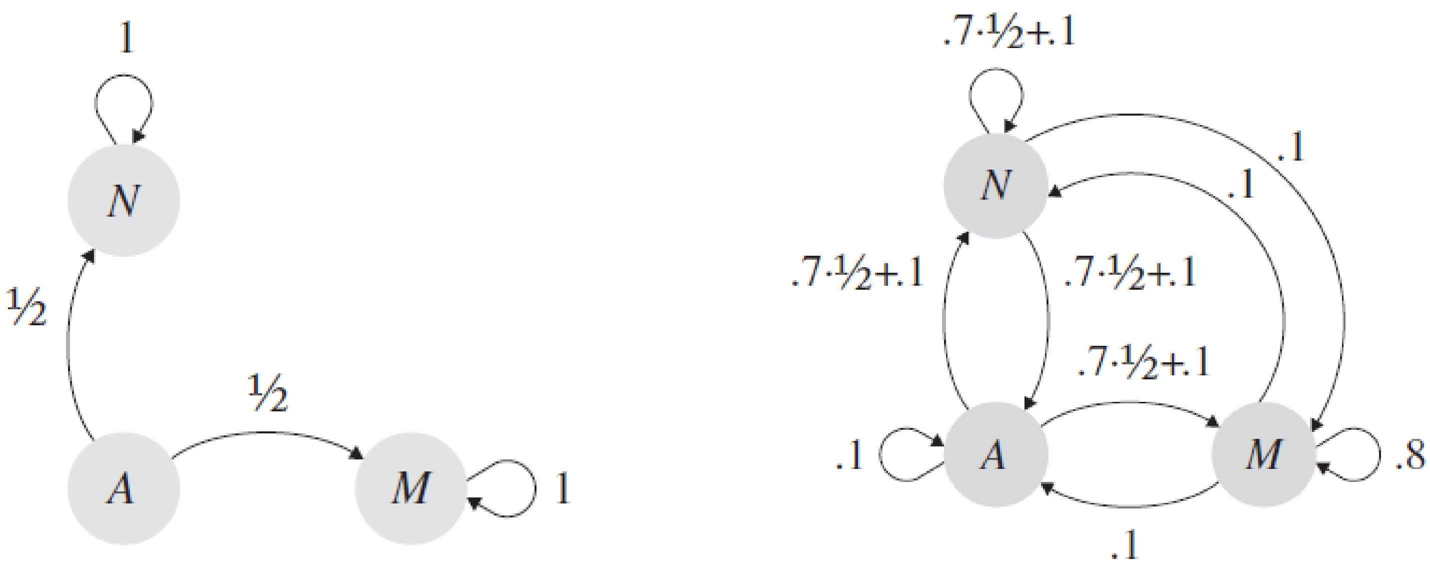
Teleporting

- ➔ **At a dead end, jump to a random web page.**
- ➔ **At any non-dead end, with probability 10%, jump to a random web page.**
  - ⇨ With remaining probability (90%), go out on a random link.
  - ⇨ 10% - a parameter.

Result of teleporting

- ➔ **Now cannot get stuck locally.**
- ➔ **There is a long-term rate at which any page is visited (not obvious, will show this).**
- ➔ **How do we compute this visit rate?**

Solution to Dead Head and Spider Traps



Apply 30% of 'tax' to each transition to be partioned among all the possible pages..

But, how can we find the limiting probability with millions of entry? Powers of P are more efficient than solving a linear system

Back to the Bayesian Classifier

$$P(\Omega_i) = P(\omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_N}) =$$

$$= P(\omega_{i_N} \mid \omega_{i_{N-1}}, \ldots, \omega_{i_1}).$$

$$P(\omega_{i_{N-1}} \mid \omega_{i_{N-2}}, \ldots, \omega_{i_1}) \ldots P(\omega_{i_1})$$

or

$$\boxed{P(\Omega_i) = \left(\prod_{k=2}^{N} P(\omega_{i_k} \mid \omega_{i_{k-1}})\right) P(\omega_{i_1})}$$

➔ **Assume:**

- ⇨ $\underline{x}_i$ statistically mutually independent
- ⇨ The pdf in one class independent of the others, then

$$\boxed{p(X \mid \Omega_i) = \prod_{k=1}^{N} p(\underline{x}_k \mid \omega_{i_k})}$$

➔ **From the above, the Bayes rule is readily seen to be equivalent to:**

$$P(\Omega_i | X)(><)P(\Omega_j | X)$$

$$P(\Omega_i)\,p(X | \Omega_i)(><)P(\Omega_j)\,p(X | \Omega_j)$$

that is, for Markov memory models, equivalent to

$$\boxed{p(X | \Omega_i)P(\Omega_i) = P(\omega_{i_1})\,p(\underline{x}_1 | \omega_{i_1}). \prod_{k=2}^{N} P(\omega_{i_k} | \omega_{i_{k-1}})\,p(\underline{x}_k | \omega_{i_k})}$$

$$\boxed{\log(p(X | \Omega_i)P(\Omega_i)) = \log(P(\omega_{i_1})\,p(\underline{x}_1 | \omega_{i_1})) + \sum_{k=2}^{N} \log(P(\omega_{i_k} | \omega_{i_{k-1}})\,p(\underline{x}_k | \omega_{i_k}))}$$

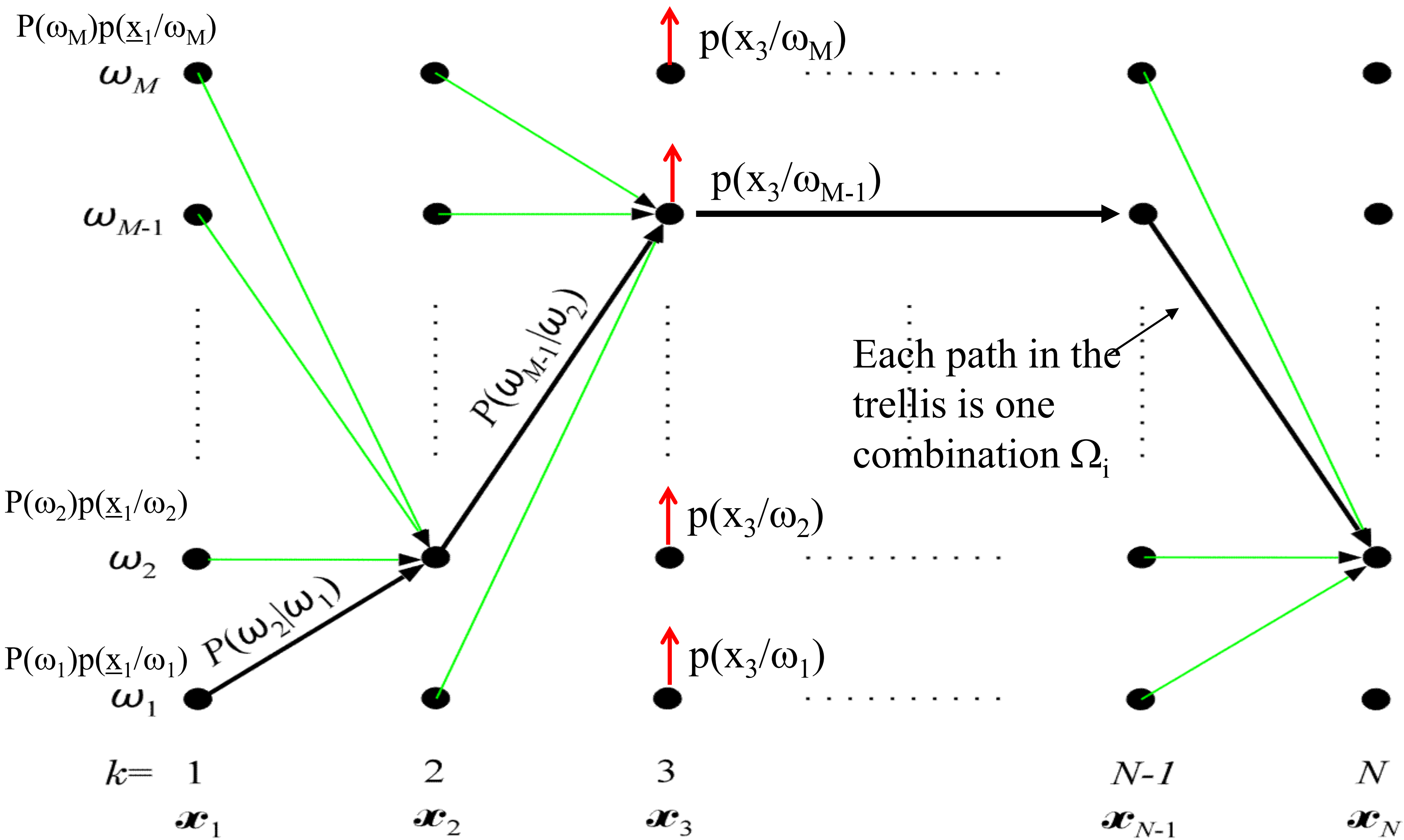
It is maximized if each term is maximized!

➔ **To find the above maximum in brute-force task we need $O(NM^N)$ operations!!**

⇨ Each sequence of class decisions requires N products

⇨ M^N total number of possible sequences

The Viterbi Algorithm



⇨ Thus, each Ω_i corresponds to one path through the trellis diagram. One of them is the optimum (e.g., black).

→ The classes along the optimal path determine the classes to which ω_i are assigned.

⇨ To each transition corresponds a cost. For our case

→ $$\hat{d}(\omega_{i_k}, \omega_{i_{k-1}}) = P(\omega_{i_k} | \omega_{i_{k-1}}).p(\underline{x}_k | \omega_{i_k})$$

→ $$\hat{d}(\omega_{i_1}, \omega_{i_0}) \equiv P(\omega_{i_1}) p(\underline{x}_i | \omega_{i_1})$$

→ $$\hat{D} = \prod_{k=1}^{N} \hat{d}(\omega_{i_k}, \omega_{i_{k-1}}) = p(X | \Omega_i) P(\Omega_i)$$

→Equivalently

$$\ln \hat{D} = \sum_{k=1}^{N} \ln \hat{d}(.,.) \equiv D = \sum_{k=1}^{N} d(.,.)$$

where,

$$d(\omega_{i_k}, \omega_{i_{k-1}}) = \ln \hat{d}(\omega_{i_k}, \omega_{i_{k-1}})$$

→Define the cost up to a node *,k,*

$$D(\omega_{i_k}) = \sum_{r=1}^{k} d(\omega_{i_r}, \omega_{i_{r-1}})$$

⇨ Bellman's principle now states

$$\boxed{\begin{aligned}&D_{\max}(\omega_{i_k}) = \max_{i_{k-1}}\left[D_{\max}(\omega_{i_{k-1}}) + d(\omega_{i_k}, \omega_{i_{k-1}})\right]\\ &i_k, i_{k-1} = 1,2,\ldots,M\end{aligned}}$$

$$D_{\max}(\omega_{i_0}) = 0$$

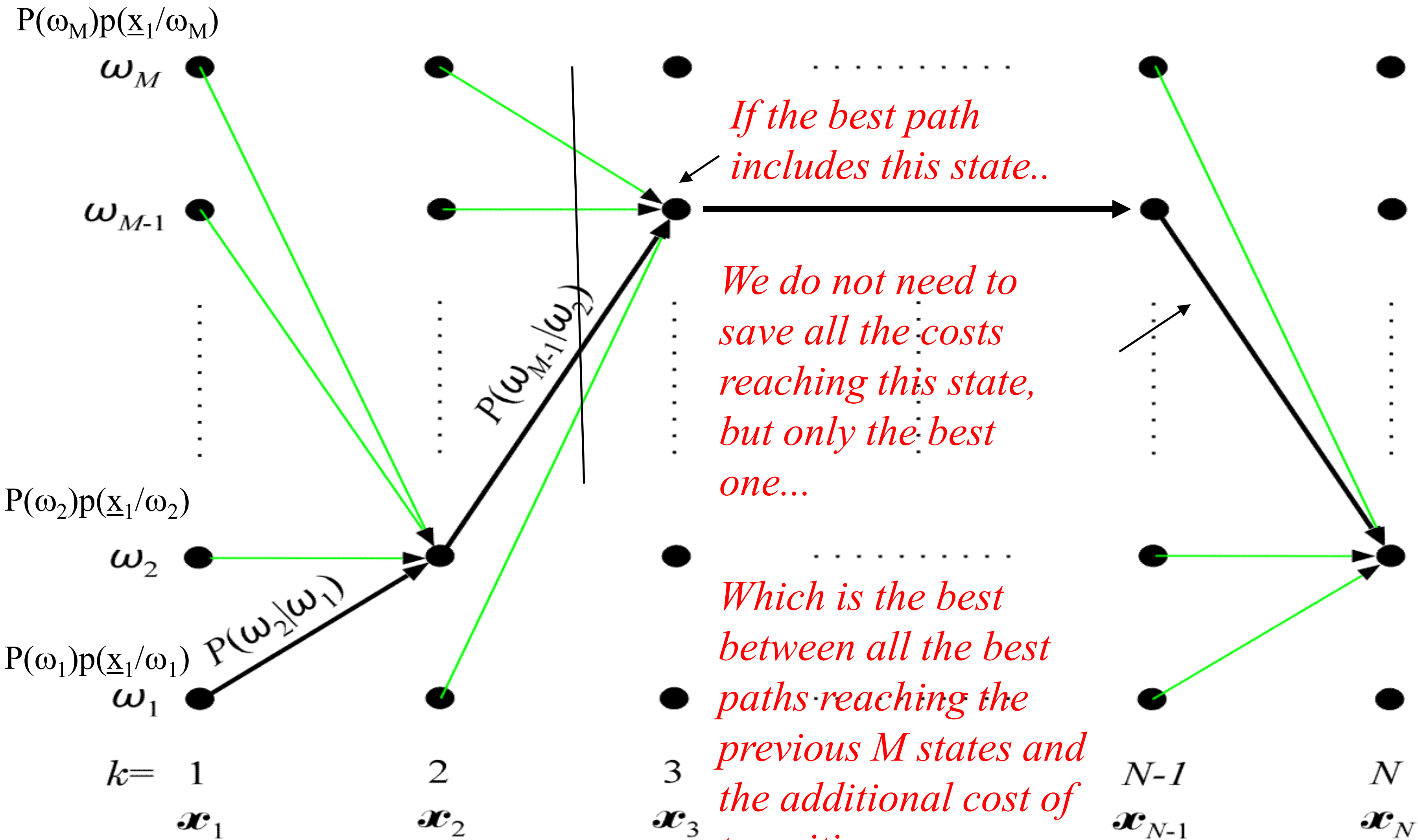
⇨ The optimal path terminates at ω_{iN}^{*} :

$$\omega_{i_N}^{*} = \arg\max_{\omega_{i_N}} D_{\max}(\omega_{i_N})$$

→Complexity $O\ (NM^2)$

» At each step, M starting points and M destination points to be compared; N total steps

The Bellman's Principle



If the best path includes this state..

We do not need to save all the costs reaching this state, but only the best one...

Which is the best between all the best paths reaching the previous M states and the additional cost of transition

A numerical example

- Assume we have a problem with 3 classes who transition probability is P=[0.1 0.7 0.2, 0.4 0.3 0.3, 0.3 0.1 0.6] and equal initial costs. It is also $p(x/\omega i)=N(\sigma i, \mu i)$, with $\sigma 1=0.03$, $\mu 1=1$; $\sigma 2=0.02$, $\mu 2=1.5$; $\sigma 3=0.1$, $\mu 3=0.5$. Assuming $x1=0.8$, $x2=1.2$ and $x3=0.9$, find the optimal classification results with/without using the memory model.

An application example: channel equalization

- ➔ **The problem: each symbol received at time k depends on information sent at time k, k-1, k-n+1**

  $\rightarrow x_k = f(I_k, I_{k-1}, \ldots, I_{k-n+1}) + n_k$

- ➔ **Can we estimate symbol I_k? From which observations?**

  $\rightarrow \underline{x}_k \equiv [x_k, x_{k-1}, \ldots, x_{k-l+1}]^T$

  $\rightarrow \underline{x}_k \rightarrow \text{equalizer} \rightarrow \hat{I}_{k-r}$

⇨ Example

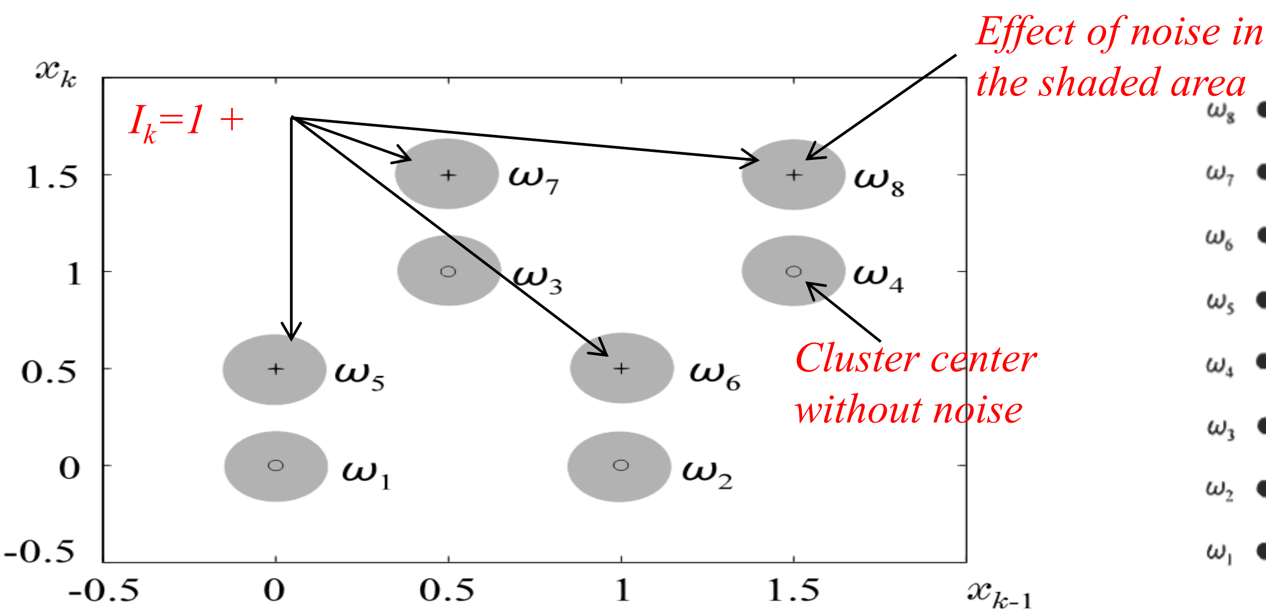
→ $x_k = 0.5 I_k + I_{k-1} + n_k$

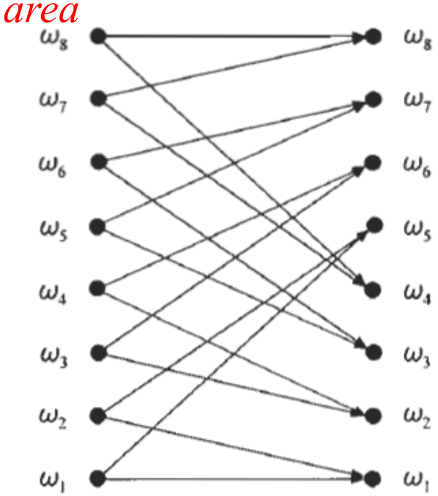
→ $\underline{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}, l = 2$

→In $\underline{x}_k$ three input symbols are involved:

I_k, I_{k-1}, I_{k-2}

I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	ω_1
0	0	1	0	1	ω_2
0	1	0	1	0.5	ω_3
0	1	1	1	1.5	ω_4
1	0	0	0.5	0	ω_5
1	0	1	0.5	1	ω_6
1	1	0	1.5	0.5	ω_7
1	1	1	1.5	1.5	ω_8





⇨ Not all transitions are allowed

→ $(I_k, I_{k-1}, I_{k-2}) = \quad 1)$

→ Then

$$(I_{k+1}, I_k, I_{k-1})$$

→ $\omega_2 \to \omega_5$, $\omega_2 \to \omega_1$

$$P(\omega_2 | \omega_i) = \begin{cases} 0.5, i = 5, 1 \\ 0, \text{otherwise} \end{cases}$$

How to 'train' the equalizer?

- ➔ **Make a choise on the length of the equalizer/M classes; send a pre-defined sequence of information bits for estimating the centers μ_i of each cluster**
- ➔ **Define a binary classifier by considering to which cluster center a new point belongs**
  - ⇨ Union of clusters define the decision regions
  - ⇨ Different distance metrics can be considered
    - → E.g. euclidean or mahalanobis distance
  - ⇨ Can we do better? by exploiting the fact that not all the class transitions are possible?
    - → i.e. correlating consecutive decisions?

Equalization as context-dependent classifier

- In this context, ω_i are related to states. Given the current state and the transmitted bit, I_k, we determine the next state. The probabilities $P(\omega_i|\omega_j)$ define the state dependence model.

- The transition cost $d(\omega_{i_k}, \omega_{i_{k-1}}) = d_{\omega_{i_k}}(\underline{x})$

$$\rightarrow = \left\| \underline{x}_k - \underline{\mu}_{ik} \right\| = ((\underline{x}_k - \underline{\mu}_{i_k})^T \sum_{i_k}^{-1} (\underline{x}_k - \underline{\mu}_{i_k}))^{\frac{1}{2}}$$

for all allowable transitions, with matrix sigma estimated during training

Hidden Markov Models - HMM

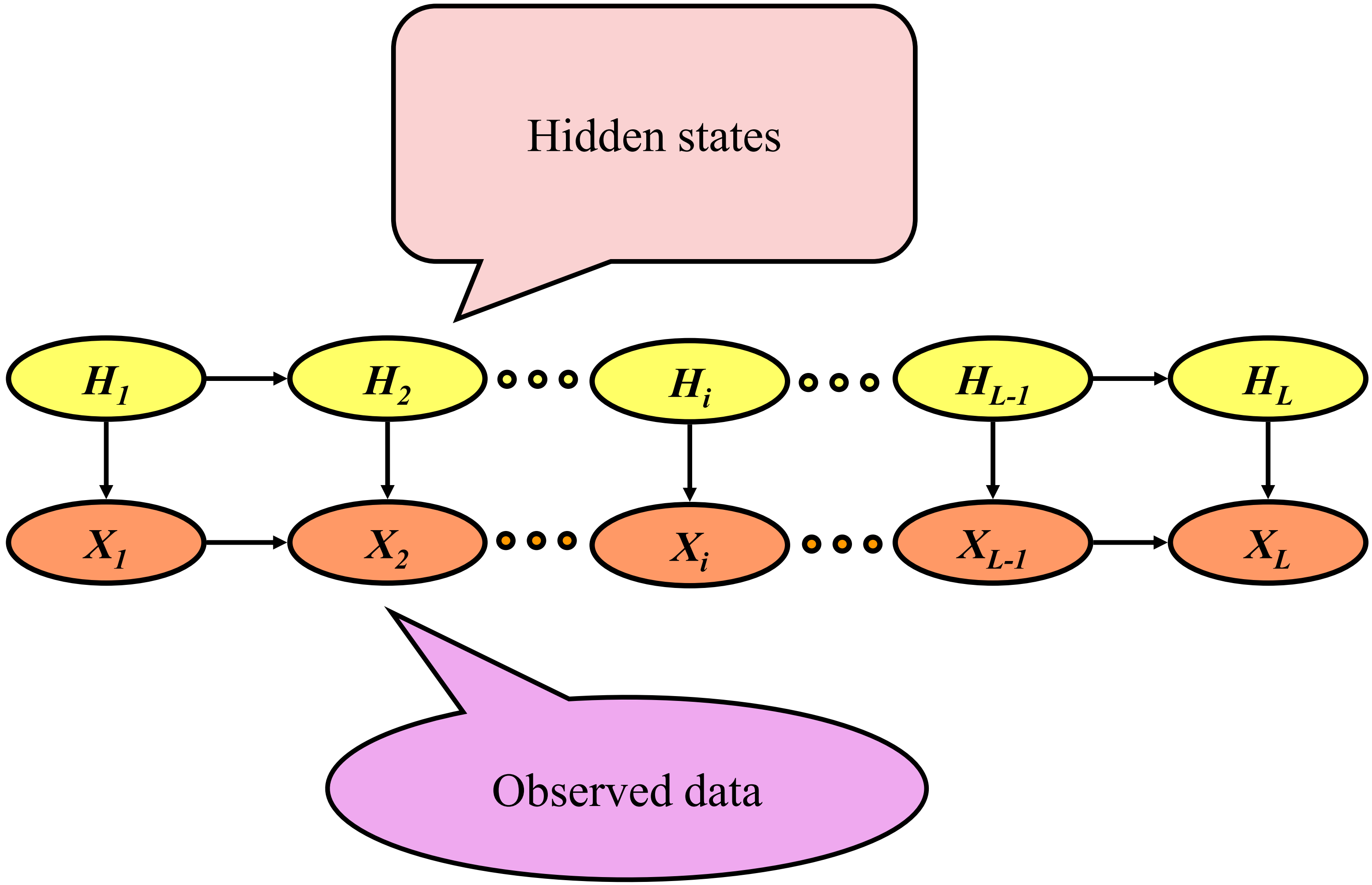
- ⇨ In the channel equalization problem, the states are observable and can be "learned" during the training period

- ⇨ Now we shall assume that states are not observable and can only be inferred from the training data

- ⇨ Applications:
  - → Speech and Music Recognition
  - → OCR
  - → Blind Equalization
  - → Bioinformatics

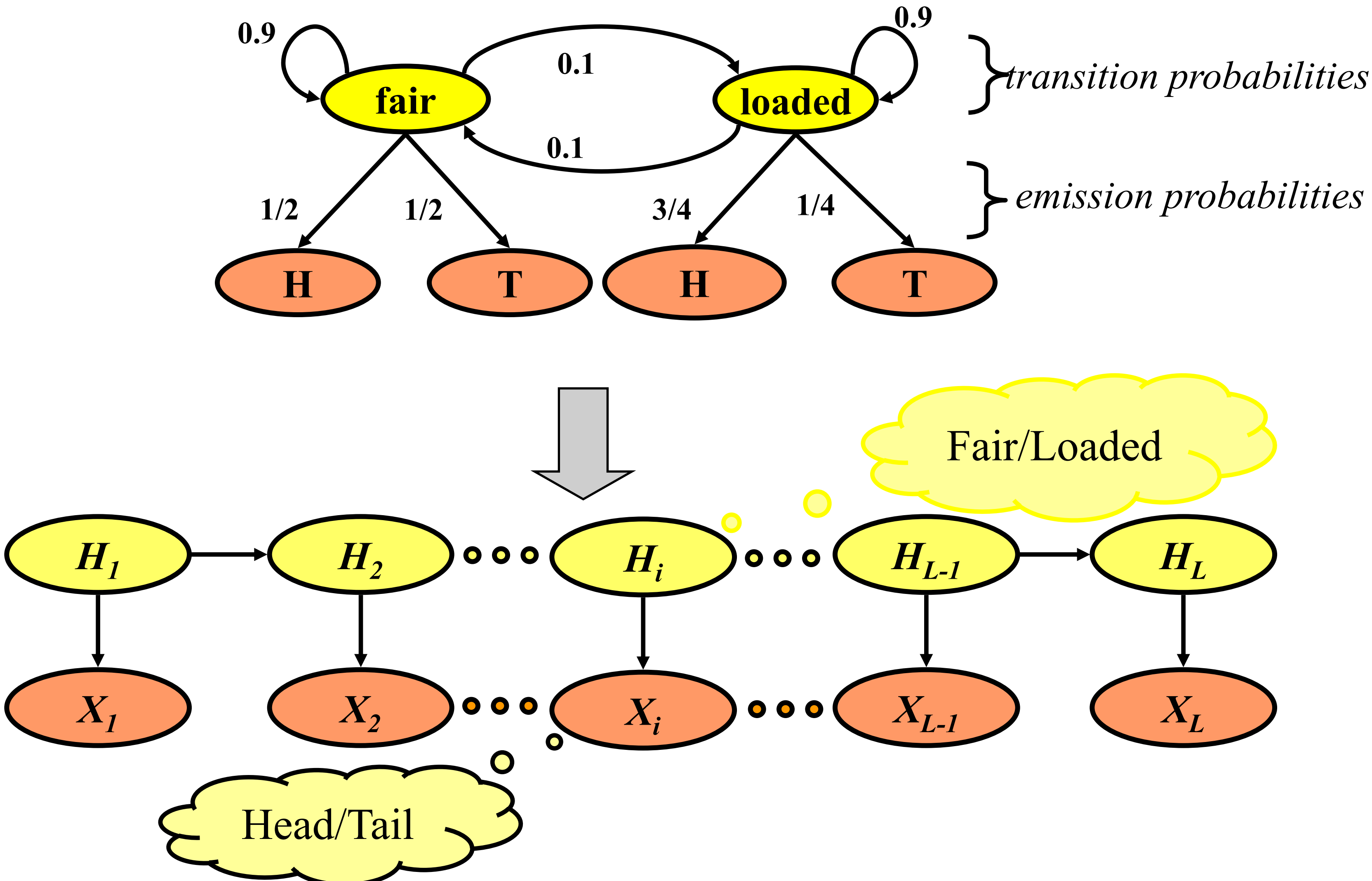
- ⇨ What we need for HMM?
  - → State model
  - → Emission model

Hidden Markov Models - HMM



Hidden Markov Models - HMM

Coin-Tossing Example



An example

Current	A	B	End
Start	0.7	0.3	0
A	0.2	0.7	0.1
B	0.7	0.2	0.1

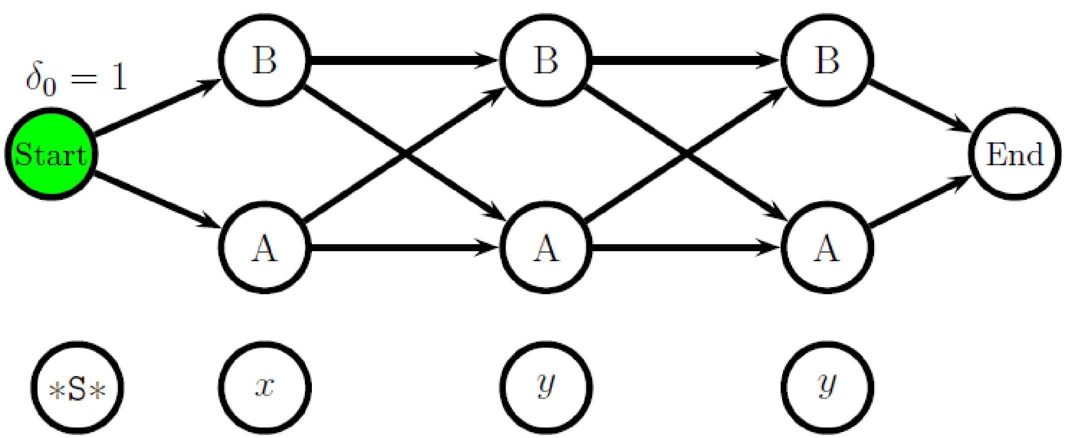
(Columns: Next)

State model

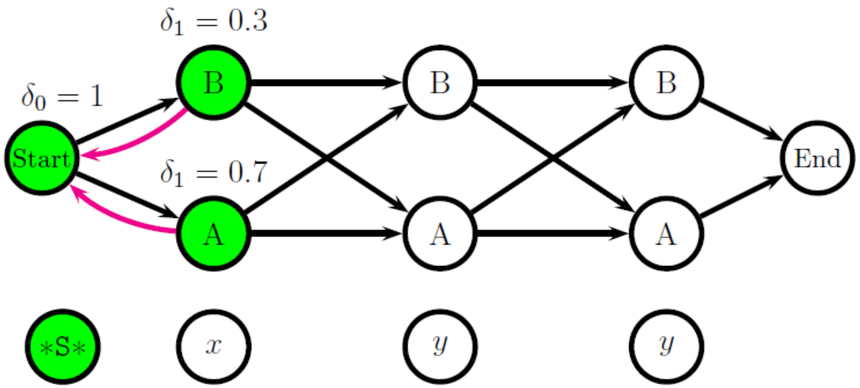
State	*S*	x	y
Start	1	0	0
A	0	0.4	0.6
B	0	0.3	0.7

(Columns: Word)

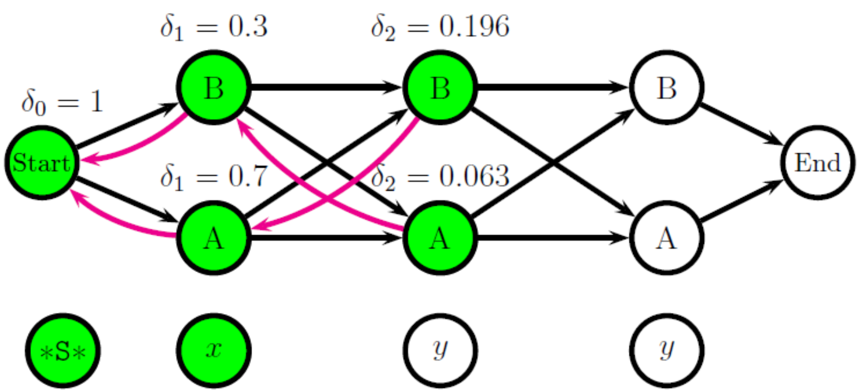
Emission model



Viterbi algorithm on HMM



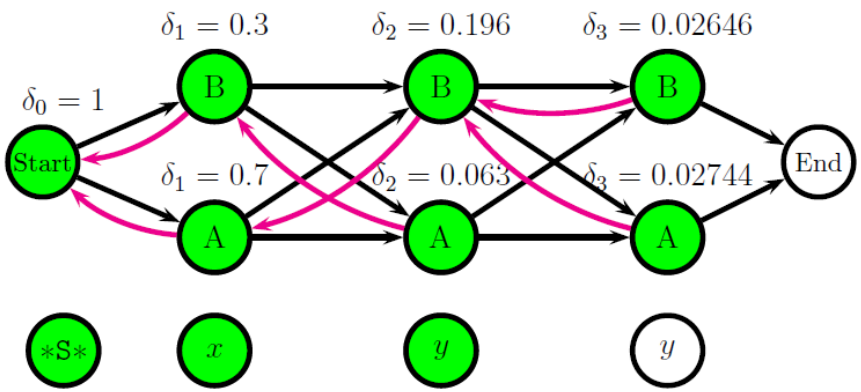
	Next		
Current	A	B	End
Start	0.7	0.3	0
A	0.2	0.7	0.1
B	0.7	0.2	0.1



	Word		
State	*S*	x	y
Start	1	0	0
A	0	0.4	0.6
B	0	0.3	0.7

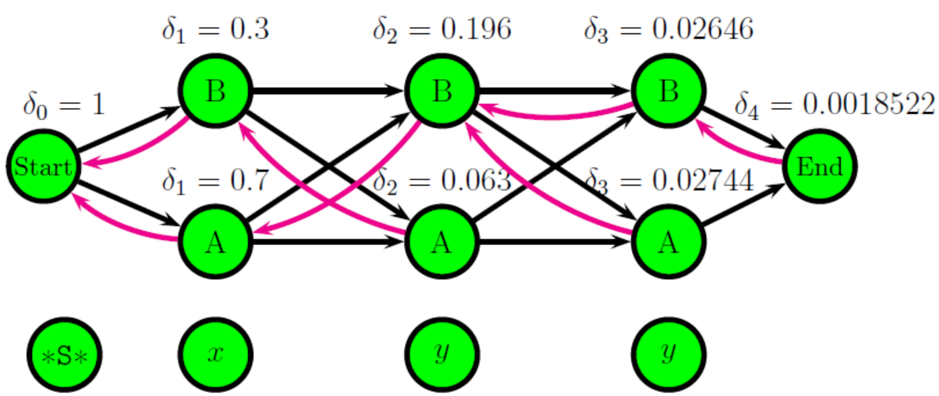
$$\delta_2(A) = \max_{s_1} P(A|s_1)P(*\mathtt{S}*|s_1)\delta_1(s_1)$$
$$= \max\{0.2 \times 0.4 \times 0.7, 0.7 \times 0.3 \times 0.3\}$$
$$\delta_2(B) = \max\{\overbrace{0.7 \times 0.4 \times 0.7}^{A}, \overbrace{0.2 \times 0.3 \times 0.3}^{B}\}$$

Viterbi Algorithm on HMM



Viterbi sequence: ABB

$P(ABB, xyy) = 0.00185522$



Which sequence without state model? e.g. based on P(A/x) and P(A/y)?

General HMM

⇨ A general HMM model is characterized by the following set of parameters

→ K, number of states

→ $P(i|j), i, j = 1,2,\ldots,K$

→ $p(\underline{x}|i), i = 1,2,\ldots,K$

→ $P(i), i = 1,2,\ldots,K$, initial state probabilities, $P(.)$

That is: $\boxed{S = \{P(i|j), p(\underline{x}|i), P(i), K\}}$

⇨ What is the problem in Pattern Recognition

→Given ***M*** **reference patterns,** each described by an HMM, find the parameters, S, for each of them (training)

→Given **an unknown pattern**, find to which one of the M, known patterns, matches best (recognition)

⇨ Recognition: Any path method

→Assume the M models to be known (M classes).

→A sequence of observations, X, is given.

→Assume observations to be emissions upon the arrival on successive states

→Decide in favor of the model S^* (from the M available) according to the Bayes rule

$$\boxed{S^* = \arg\max_S P(S|X)}$$

for equiprobable patterns

$$\boxed{S^* = \arg\max_S p(X|S)}$$

→For each model S there is more than one possible sets of successive state transitions Ω_i, each with probability

$$P(\Omega_i | S)$$

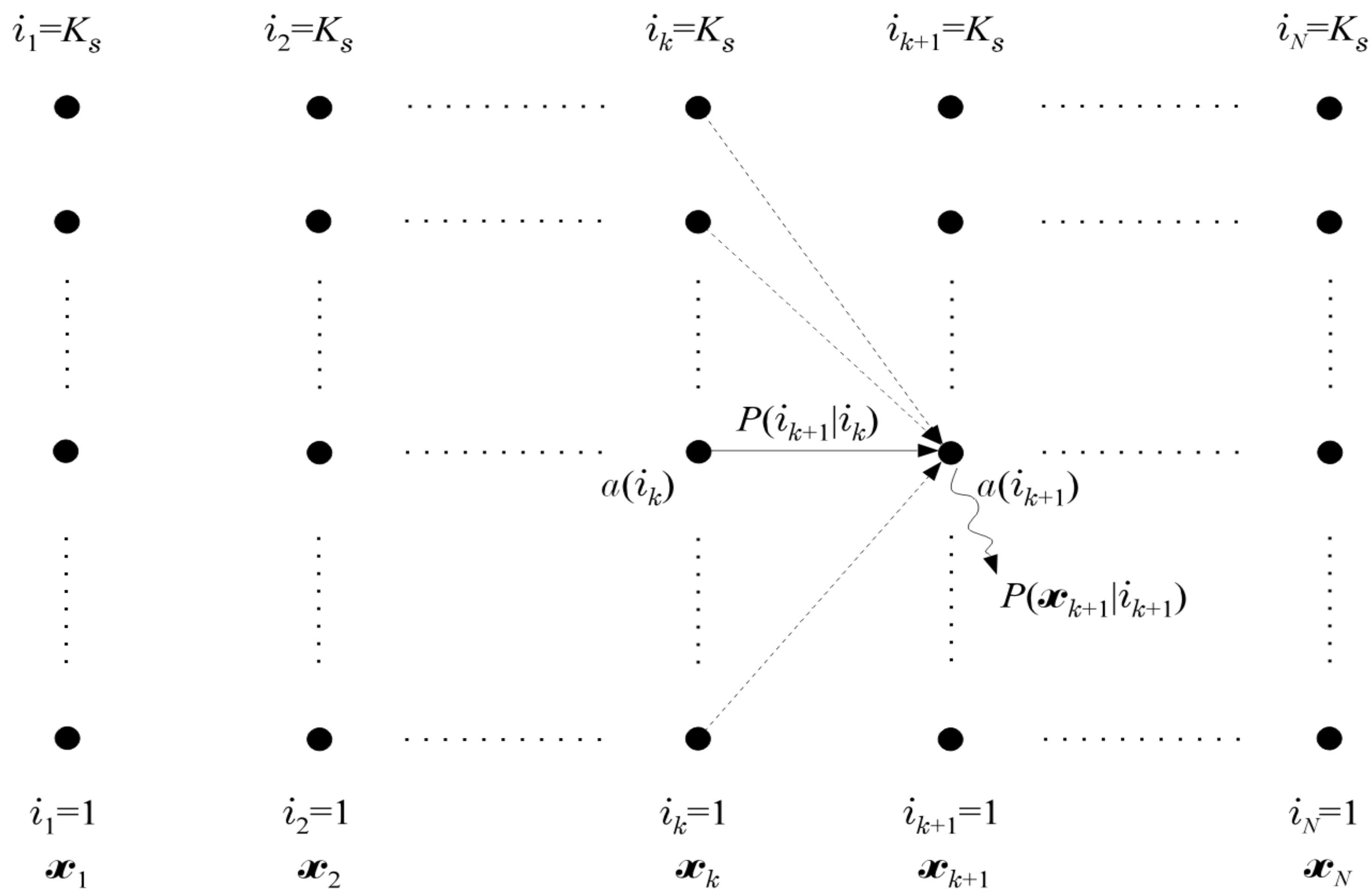
Thus:

$$\boxed{\begin{aligned} P(X|S) &= \sum_i p(X, \Omega_i | S) \\ &= \sum_i p(X|\Omega_i, S) P(\Omega_i | S) \end{aligned}}$$

→Given the state i_k at step k, for the efficient computation of the above DEFINE

» $$\alpha(i_{k+1}) = p(\underline{x}_1, \ldots, \underline{x}_{k+1}, i_{k+1} | S)$$

$$= \underbrace{\sum_{i_k} \alpha(i_k)}_{\text{History}} \underbrace{P(i_{k+1} | i_k) p(\underline{x}_{k+1} | i_{k+1})}_{\text{Local activity}}$$



→Observe that

$$\boxed{P(X|S) = \sum_{i_N=1}^{K_S} \alpha(i_N)}$$

Compute this for each S and find the maximum!

→Some more quantities

» $$\beta(i_k) = p(\underline{x}_{k+1}, \underline{x}_{k+2}, \ldots, \underline{x}_N \,|\, i_k, S)$$
$$= \sum_{i_{k+1}} \beta(i_{k+1}) P(i_{k+1} \,|\, i_k)\, p(\underline{x}_{k+1} \,|\, i_{k+1})$$

» $$\gamma(i_k) = p(\underline{x}_1, \ldots, \underline{x}_N, i_k \,|\, S)$$
$$= \alpha(i_k)\beta(i_k)$$

⇨ **Training**

→The philosophy:

Given a training set *X*, known to belong to the specific model, estimate the unknown parameters o f *S*, so that the **output** of the model, e.g.

$$p(X|S) = \sum_{i_{N=1}}^{K_s} \alpha(i_{\mathrm{N}})$$

to be maximized

⇨ This is a ML estimation problem with missing data

⇨ Assumption: Data $\underline{x}$ discrete

$$\underline{x} \in \{1,2,\ldots,r\} \Rightarrow p(\underline{x}|i) \equiv P(\underline{x}|i)$$

⇨ Definitions:

→ $$\xi_k(i,j) = \frac{\alpha(i_k = i) P(j|i) P(\underline{x}_{k+1}|j) \beta(i_{k+1} = j)}{P(X|S)}$$

→ $$\gamma_k(i) = \frac{\alpha(i_k = i) \beta(i_k = i)}{P(X|S)}$$

⇨ The Algorithm:

→Initial conditions for all the unknown parameters. Compute $P(X|S)$

→Step 1: From the current estimates of the model parameters reestimate the new model S from

- $$\overline{P}(j|i) = \frac{\sum_{k=1}^{N-1} \xi_k(i,j)}{\sum_{k=1}^{N-1} \gamma_k(i)} \quad \left(= \frac{\#\text{ of transitions from } i \text{ to } j}{\#\text{ of transitions from } i} \right)$$

- $$\boxed{\overline{P}_{\underline{x}}(r|i) = \frac{\sum_{k=1 \text{ and } \underline{x} \to r)}^{N} \gamma_k(i)}{\sum_{k=1}^{N} \gamma_k(i)}} \quad \left(= \frac{\text{at state } i \text{ and } \underline{x} = r}{\neq \text{of being at state } i} \right)$$

- $$\boxed{\overline{P}(i) = \gamma_1(i)}$$

→Step 3: Compute $P(X|\bar{S})$. If $P(X|\bar{S}) - P(X|S) > \varepsilon,\ S = \bar{S}$ go to step 2. Otherwise stop

→Remarks:

» Each iteration improves the model

$$\bar{S}:\ P(X|\bar{S}) > P(X|S)$$

» The algorithm converges to a maximum (local or global)

» The algorithm is an implementation of the EM algorithm