Introduction to Reinforcement Learning

2025. 1st semester

단국대학교
DANKOOK UNIVERSITY

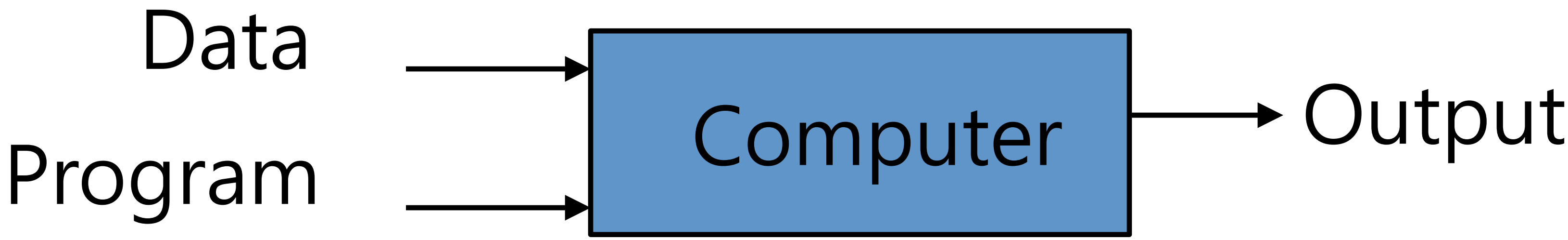
Mobile System Engineering

What is Machine Learning?

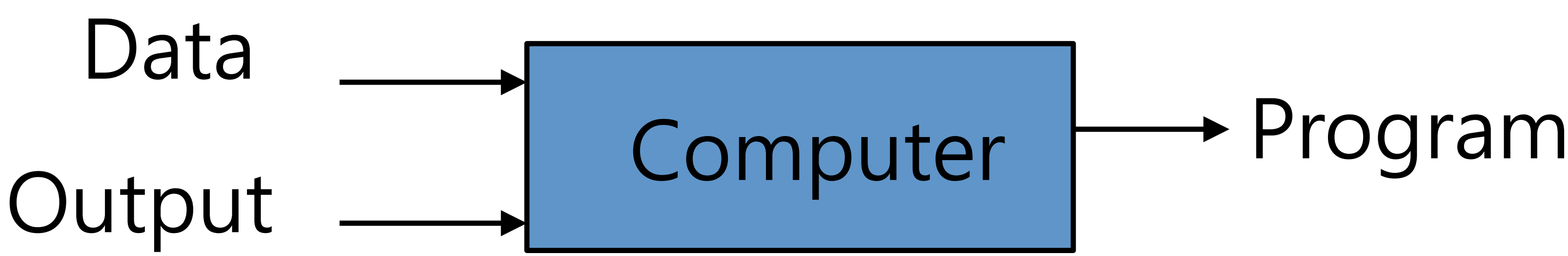
- "*Learning is any process by which a system improves performance from experience.*" -- *Herbert Simon*

- *Definition by Tom Mitchell (1998):*
  - *Machine Learning is the study of algorithms that*
    - *improve their performance P*
    - *at some task T*
    - *with experience E.*
  - *A well-defined learning task is given by <P, T, E>.*

Traditional Programming



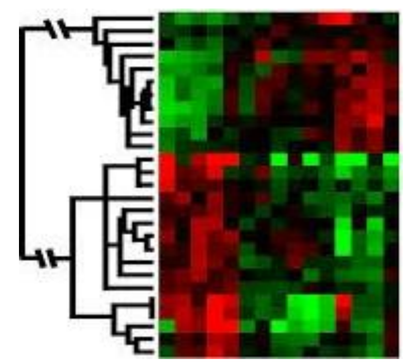
Machine Learning



When Do We Use Machine Learning?

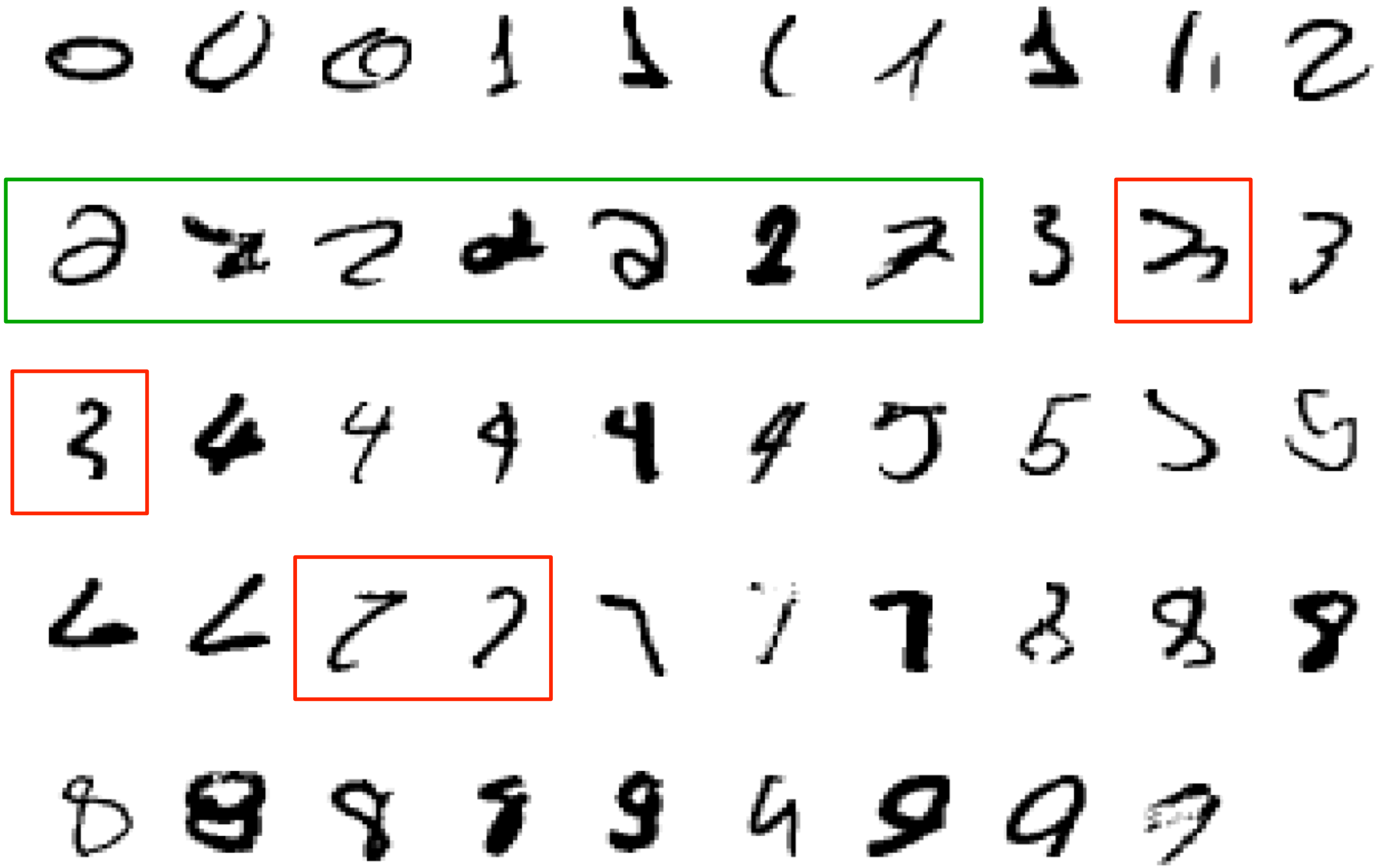
- ML is used when:
  - Human expertise does not exist (navigating on Mars)
  - Humans can't explain their expertise (speech recognition)
  - Models must be customized (personalized medicine)
  - Models are based on huge amounts of data (genomics)



- Learning isn't always useful:
  - There is no need to "learn" to calculate payroll

A classic example of a task that requires machine learning: It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

Some more examples of tasks that are best solved by using a learning algorithm

- ***Recognizing patterns:***
  - *Facial identities or facial expressions*
  - *Handwritten or spoken words*
  - *Medical images*
- ***Generating patterns:***
  - *Generating images or motion sequences*
- ***Recognizing anomalies:***
  - *Unusual credit card transactions*
  - *Unusual paderns of sensor readings in a nuclear power plant*
- ***Prediction:***
  - *Future stock prices or currency exchange rates*

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

Samuel's Checkers-Player

- "*Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.*"
  - Arthur Samuel (1959)

Improve on task T,
with respect to performance metric P,
based on experience E

Defining the Learning Task

- *T: Playing checkers*
- *P: Percentage of games won against an arbitrary opponent*
- *E: Playing practice games against itself*

- *T: Recognizing hand-written words*
- *P: Percentage of words correctly classified*
- *E: Database of human-labeled images of handwritten words*

- *T: Driving on four-lane highways using vision sensors*
- *P: Average distance traveled before a human-judged error*
- *E: A sequence of images and steering commands recorded while observing a human driver*

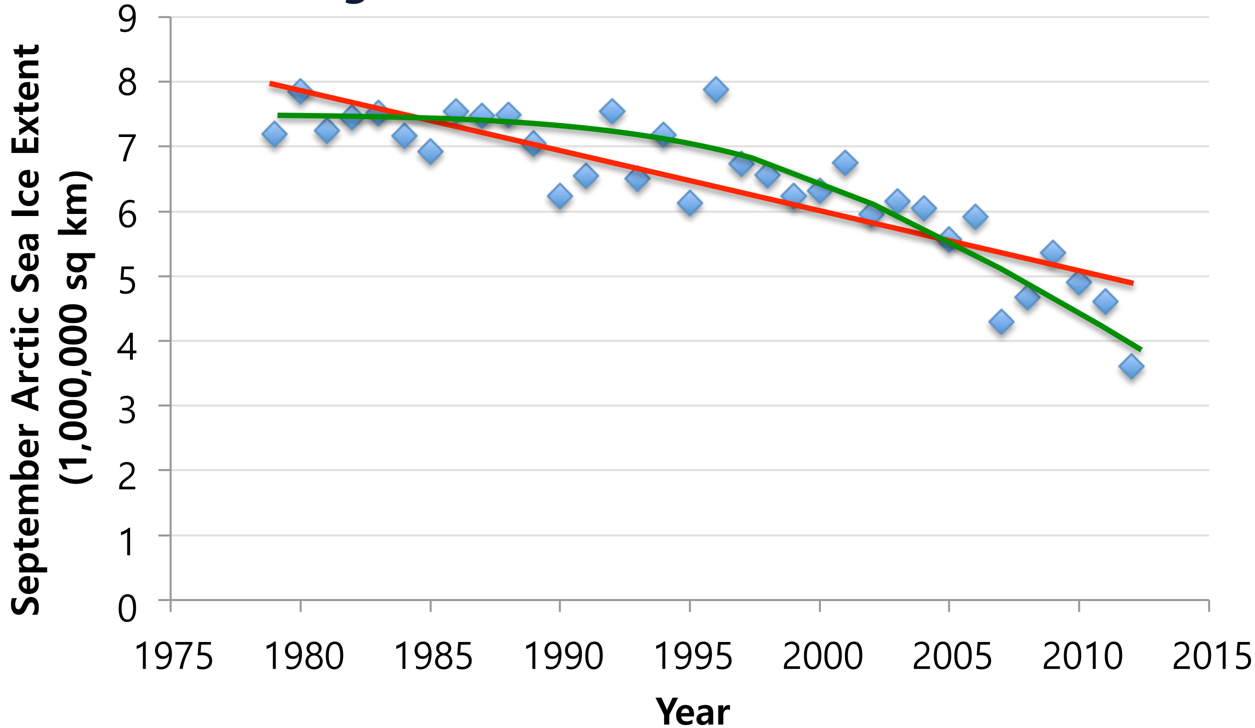
- *T: Categorize email messages as spam or legitimate*
- *P: Percentage of email messages correctly classified*
- *E: Database of emails, some with human-given labels*

Types of Learning

- *Supervised (inductive) learning*
  - *Given: training data + desired outputs (labels)*
- *Unsupervised learning*
  - *Given: training data (without desired outputs)*
- *Semi-supervised learning*
  - *Given: training data + a few desired outputs*
- *Reinforcement learning*
  - *Rewards from sequence of actions*

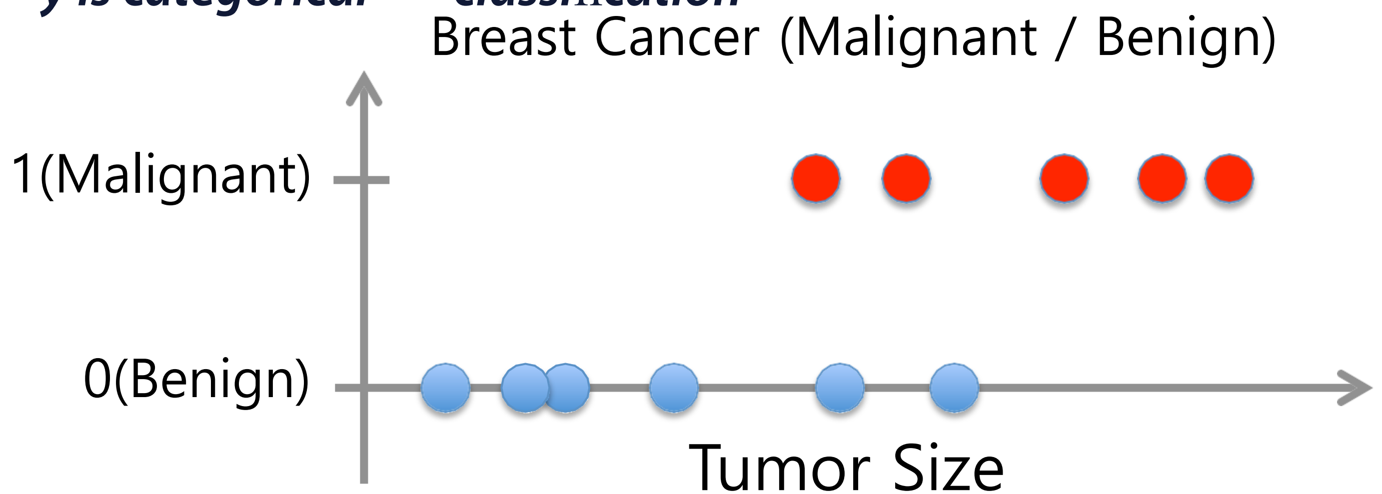
Supervised Learning: Regression

- Given (x1, y1), (x2, y2), ..., (xn, yn)
- Learn a function f(x) to predict y given x
  - y is real-valued == regression



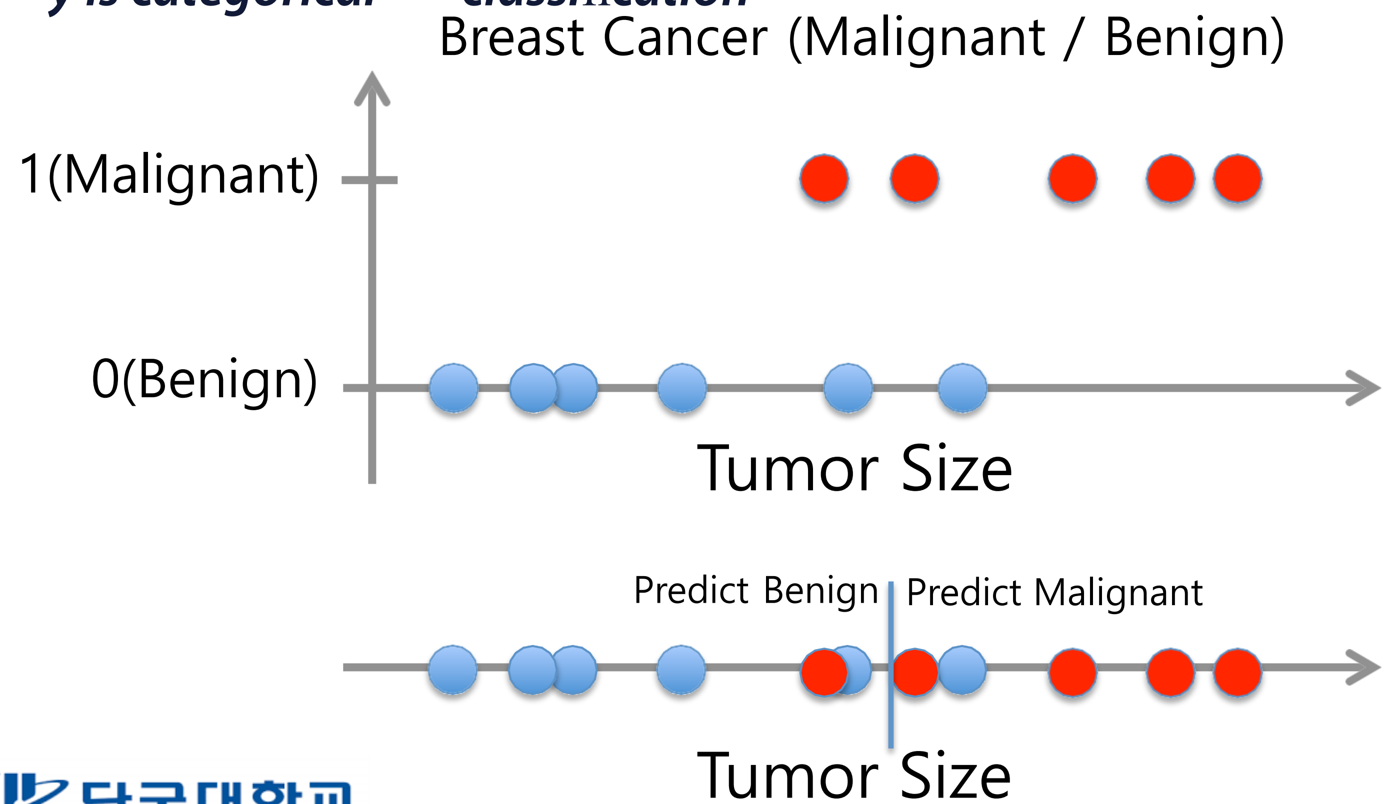
Supervised Learning: Classification

- Given (x1, y1), (x2, y2), ..., (xn, yn)
- Learn a function f(x) to predict y given x
  - y is categorical == classification



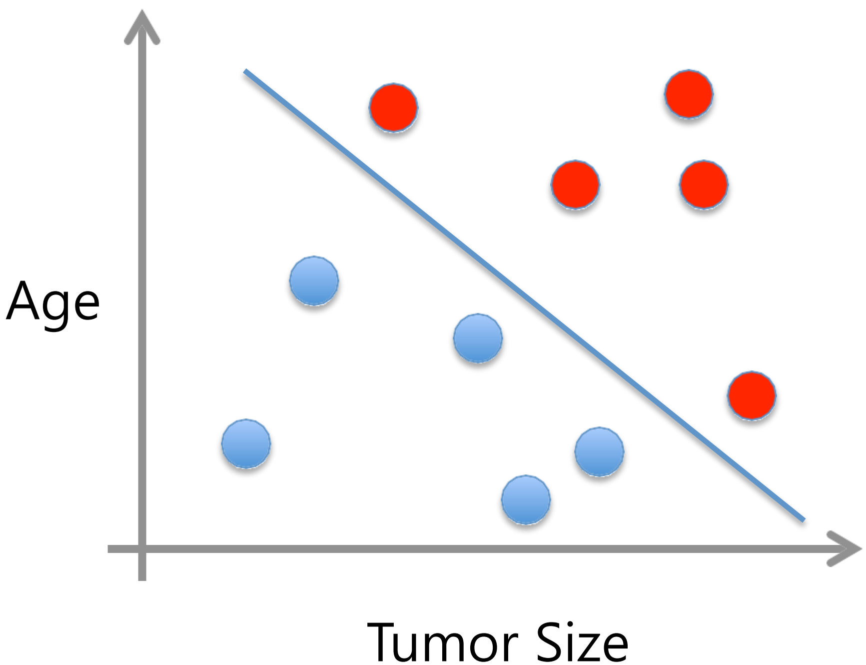
Supervised Learning: Classification

- Given (x1, y1), (x2, y2), ..., (xn, yn)
- Learn a function f(x) to predict y given x
  - y is categorical == classification



Supervised Learning

- *x can be multi-dimensional*
  - *Each dimension corresponds to an attribute*

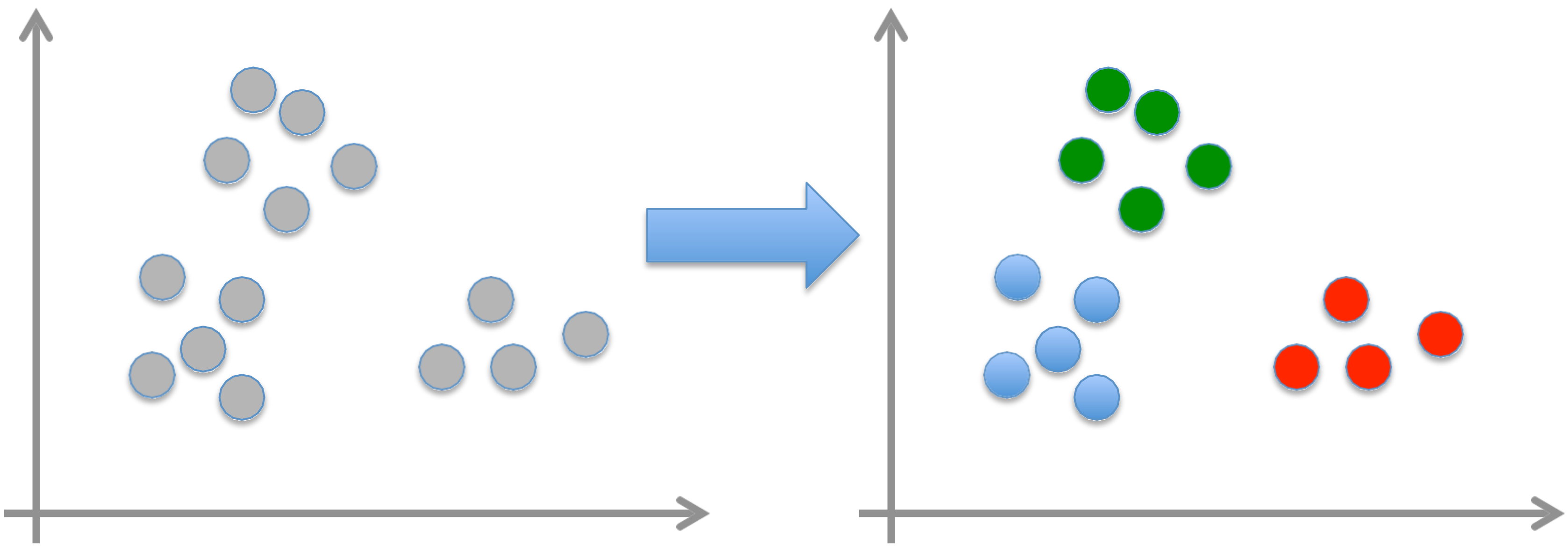


- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

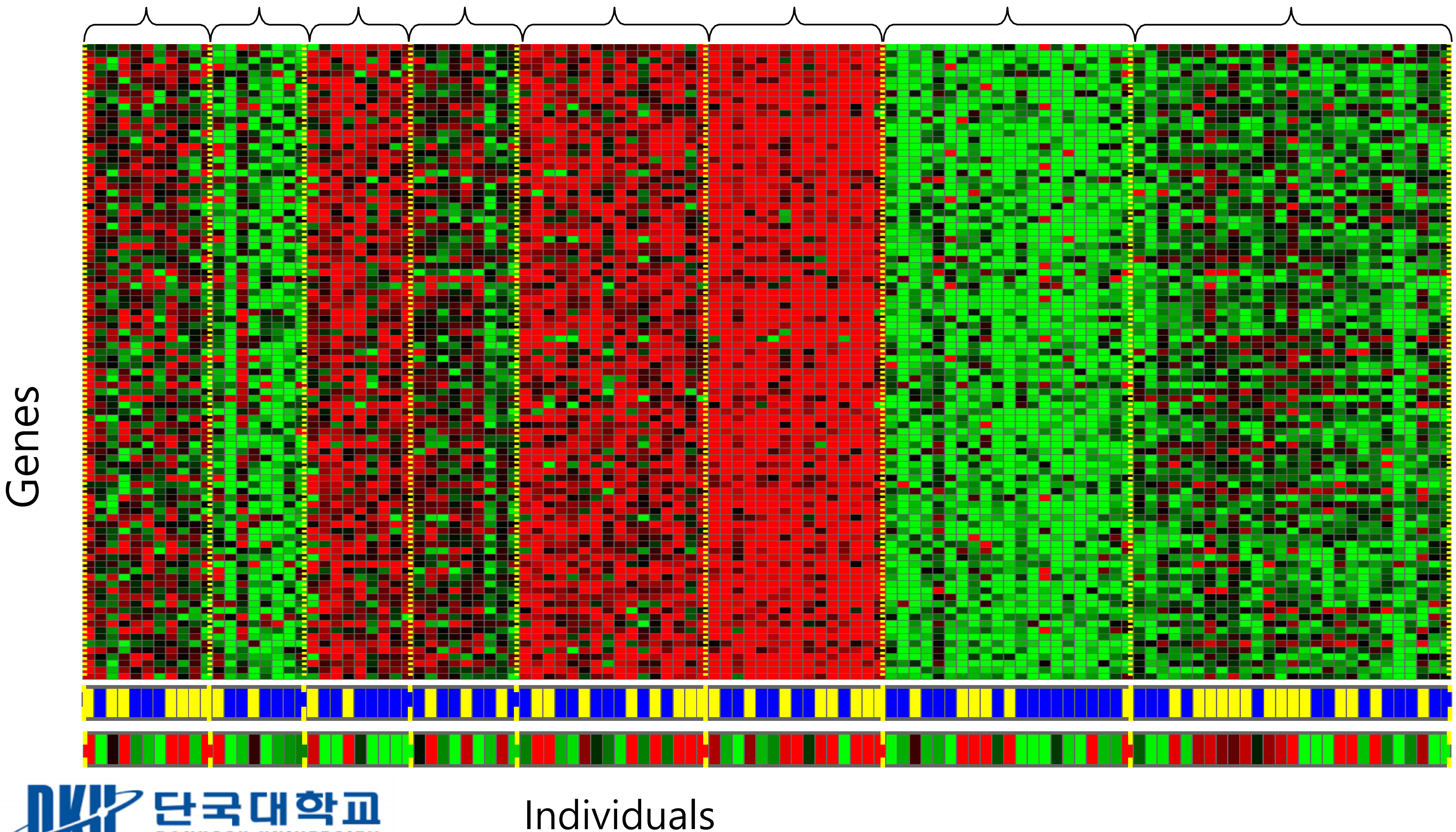
Unsupervised Learning

- *Given x1, x2, …, xn (without labels)*
- *Output hidden structure behind the x's*
  - *E.g., clustering*



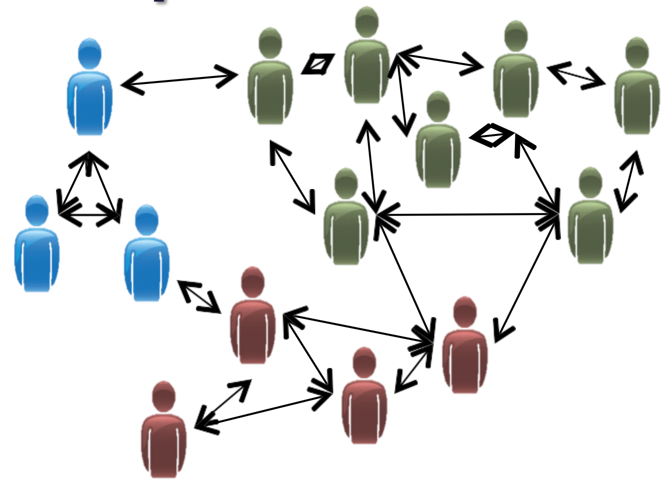
Unsupervised Learning

Genomics application: group individuals by genetic similarity



Unsupervised Learning

Organize computing clusters



Social network analysis

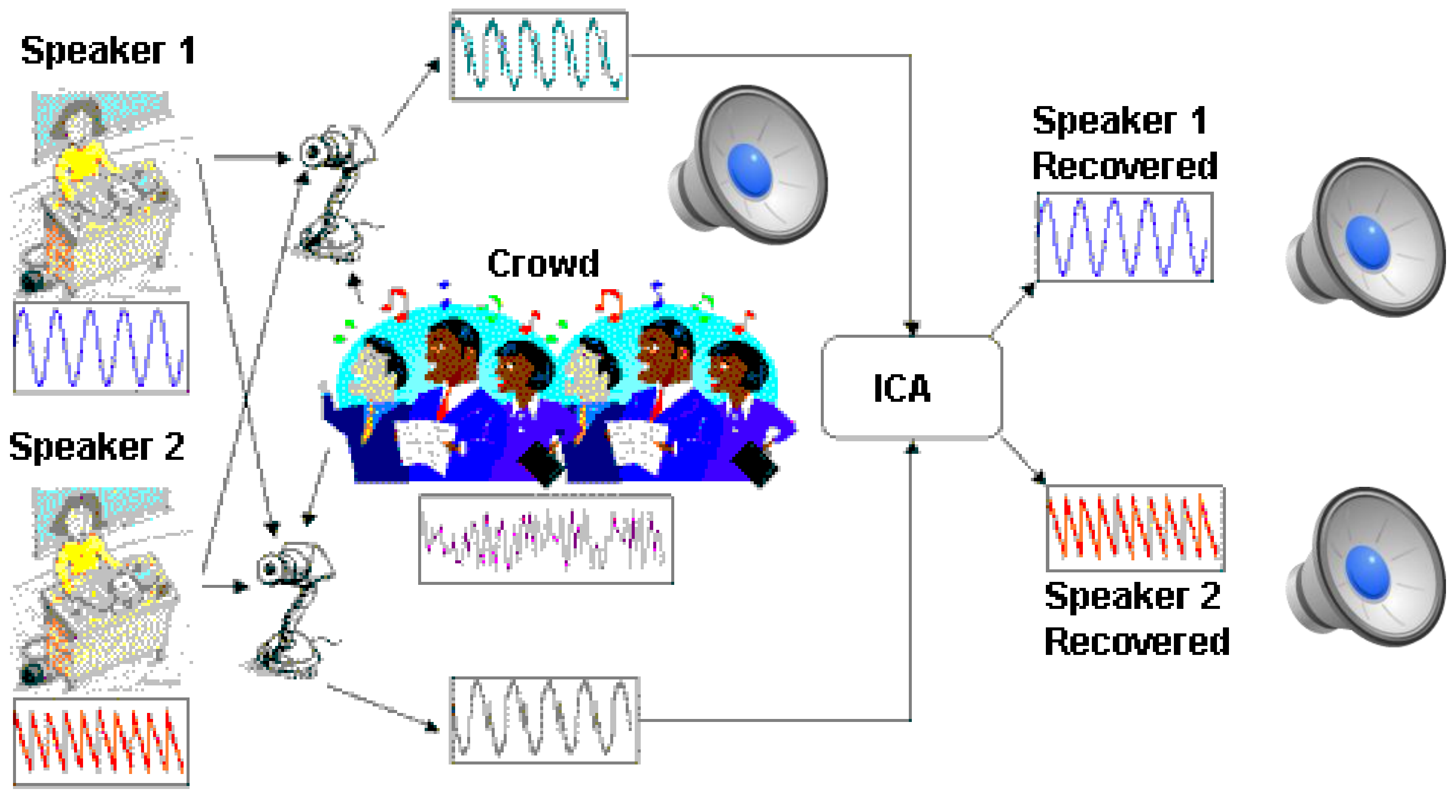
Market segmentation

Image credit: NASA/JPL-Caltech/E. Churchwell (Univ. of Wisconsi
dison)

Astronomical data analysis

Unsupervised Learning

- Independent component analysis
  - separate a combined signal into its original sources



Reinforcement Learning

https://youtu.be/ldXxDNjS5jw

단국대학교 DANKOOK UNIVERSITY

Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states → actions that tells you what to do in a given state

- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze
  - Balance a pole on your hand

What is RL?

- *The process of developing through trial and error*

- *A learning process that corrects behavior through trial and error to maximize cumulative rewards in sequential decision-making problems*

Sequential Decision-Making Process

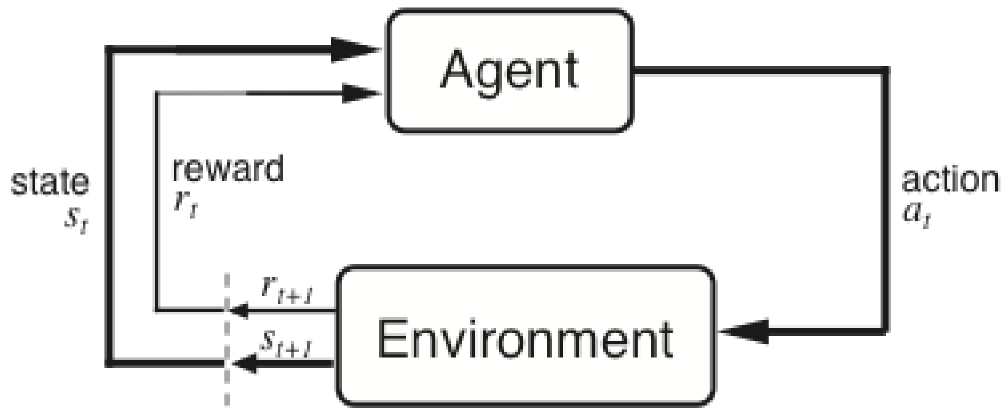
- *Shower Problem*
  - *Taking off clothes*
  - *Taking a shower*
  - *Drying up*
  - *Wearing clothes*

- *No matter how simple a process is, several decisions must be made "sequentially" in order to successfully complete it.*

Examples of Sequence decision-making

- *Portfolio Management in Stock Investments*
  - *What stocks do I buy/sell every moment?*
- *Drive*
  - *Which road will you use? highway? national highway?*
  - *Which lane will you use?*
    - *What if the car in front is a beginner driver? Or a truck?*
  - *Should I step on the accelerator or brake now?*
- *Game (LOL)*
  - *Which champion will you play?*
  - *which line are you going to go on?*
  - *which item to buy?*

The Agent-Environment Interface



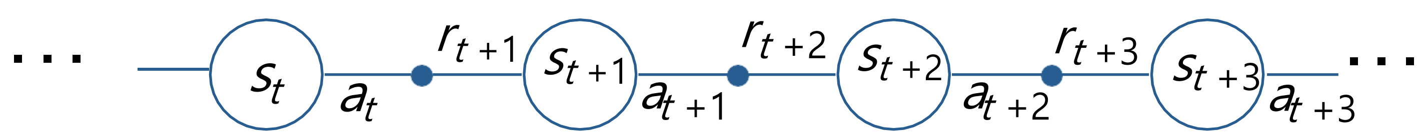
Agent and environment interact at discrete time steps : $t = 0, 1, 2, K$

Agent observes state at step t: $s_t \in S$

produces action at step t : $a_t \in A(s_t)$

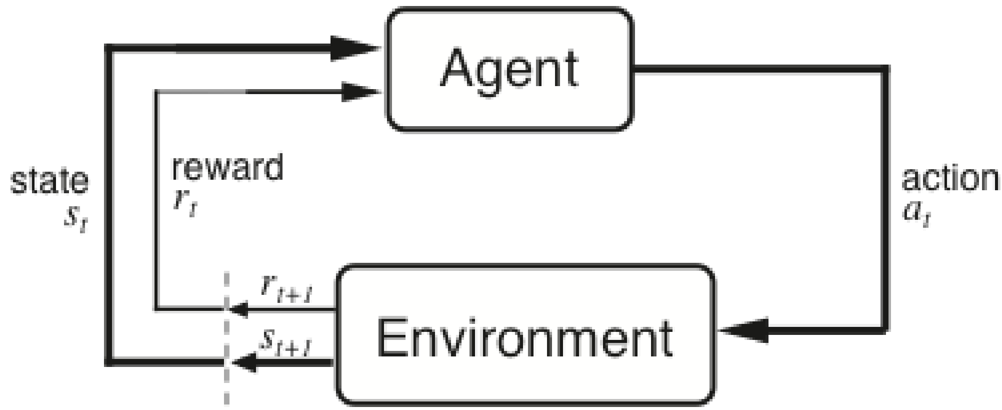
gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



DKU 단국대학교 DANKOOK UNIVERSITY

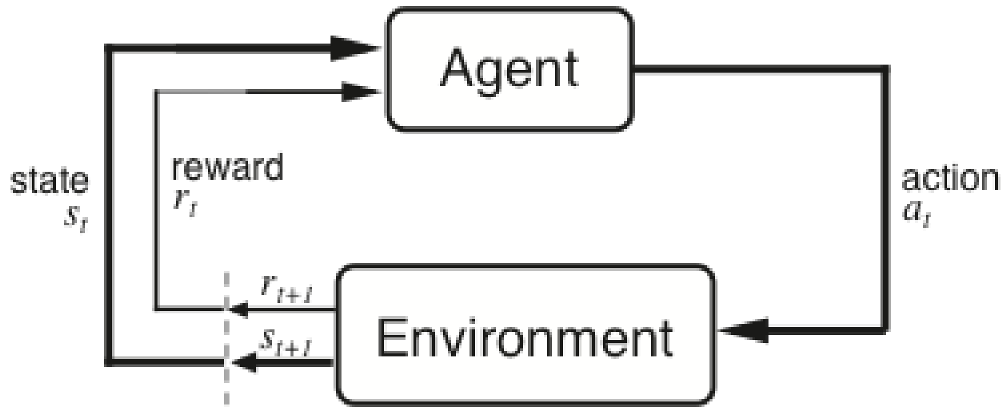
The Agent-Environment Interface



- **Agent**: *The protagonist (hero), subject, of reinforcement learning*
  - *Cyclists, drivers, game characters, etc.*

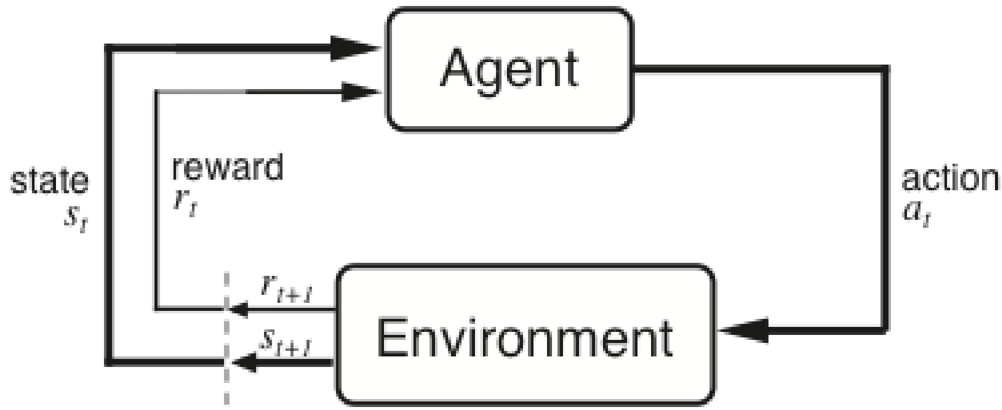
1. *Decide which action a_t should be taken in the current state s_t*
2. *The environment changes through the determined action a_t*
3. *Receive information about the reward r_t and the next state s_{t+1} from the changed environment*

The Agent-Environment Interface



- **Environment**: *everything except the agent*
  - *wind, bike, floor, etc…*
  1. ***Cause state change through action a_t received from agent***
  2. ***State: $s_t \rightarrow s_{t+1}$***
  3. ***Calculate the reward r_{t+1} for the agent***
  4. ***Deliver state, reward s_{t+1}, r_{t+1} to agent***

The Agent-Environment Interface



- *State: A record of all information about the current state in numerical form*
  - *A position of a bike* = $\{Left, Center, Right\}$
  - *An angle of a handle* = $\{Left, Center-left, Center, Center-right, Right\}$

Reward

- *Signs of how well you are (or subject is) making decisions*

- *cumulative reward*
  - *The sum of rewards received in the process of reinforcement learning*

- *E.g., Cycling*
  - *+1 per 1m moving forward*

Property of Reward

- Not how but how much
- Quantitative rewards have no “How” information
- So how can you know about “How”?
- Numerous trials and errors
  - Stepped on the pedal slowly and fell quickly!
  - Aha, if you pedal slowly, you fall quickly!
  - I tried pedaling quickly and I could go 3m more!
  - Shall we step on the pedal a little faster then?

- Depending on how you set up the reward, the direction of trial-and-error changes

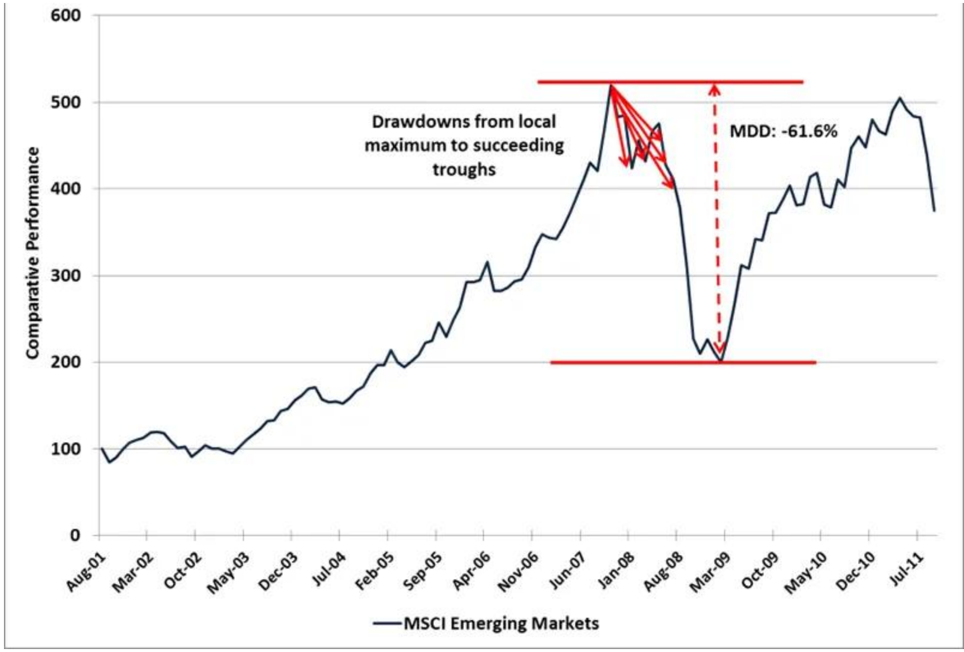
Property of Reward

- Reward is a scalar, not a vector
  - Only one goal should be set
  - "Is this really an appropriate assumption?"
- Multiple goals can be set as one reward
  - through weighting $+x$ per 1m $-y$ whenever crossing a restricted area
  - Reward $= x - y$

- Reinforcement learning may not be appropriate for problems that are difficult to represent reward in scalar form

Property of Reward

- Benefits from Asset Portfolio Allocation
  - rate of return
  - Maximum drawdown



- Distance traveled on a bicycle without falling

- winning on the game

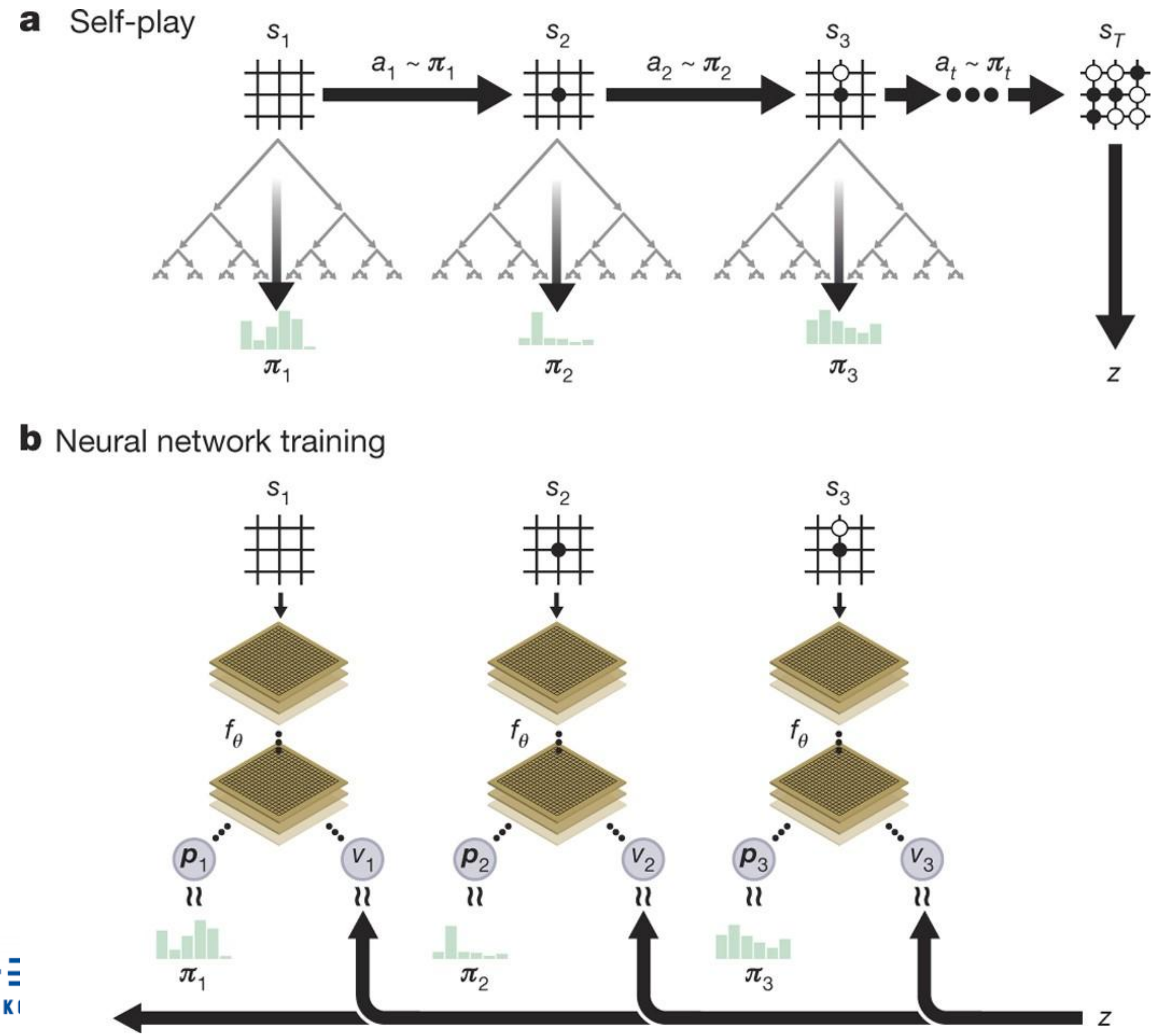
Property of Reward

- Rare and delayed rewards
- Baduk (바둑) : +10000 if you win, but the impact of the current pick happens after a long time
- For supervised learning, "instant" rewards occur

An advantage of RL

- *Parallelization*
  - *What if 100 agents went through trial and error at the same time?*

Reinforcement learning + Massive computing power



MARKOV PROCESS

Stochastic Process

- *Stochastic Process (Random Process)*
  - *widely used as mathematical models of systems and phenomena that appear to vary in a random manner*
  - *a sequence of possible events in which happens with probabilities*

Markov Process

- *Markov Process (Markov Chain)*
  - *A stochastic model describing a sequence of possible events in which the probability of each event **depends only on the state attained in the previous event***
- *Markov Property*
  - *The conditional probability distribution of future states of the process depends only upon the present state*
  - *Also called as **memoryless** property*

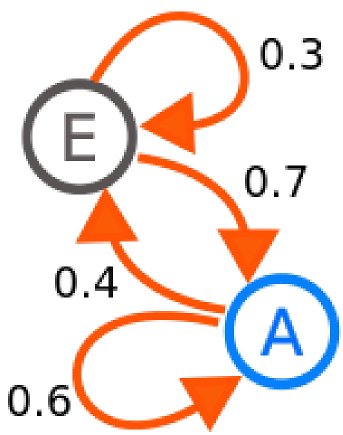
Markov chain

- *Markov chain model*
  - *Markov chains were introduced in 1906 by Andrey Markov (Russian mathematician, 1856-1922) and were named in his honor*
  - *One of the most powerful tools for analyzing complex stochastic system*
  - *Markov Chain has been applied to short term market forecasting and business decision, analysis of algorithms, network protocols, diverse social issues and phenomenon*

Markov Process

- *Markov Process ≡ {State, Transition Probability}*
- *State*
  - *Discrete set of states* $i \in S$
  - *All states of the Markov chain communicate with each other*
  - *Let state of a system at time t be* X_t
- *Transition Probability*
  - *The probability of moving from one state to another is defined regardless of which state you have been through*

$$p_{ij}(t) = \Pr\{X_{t+1} = j \mid X_t = i\} \quad i, j \in S$$

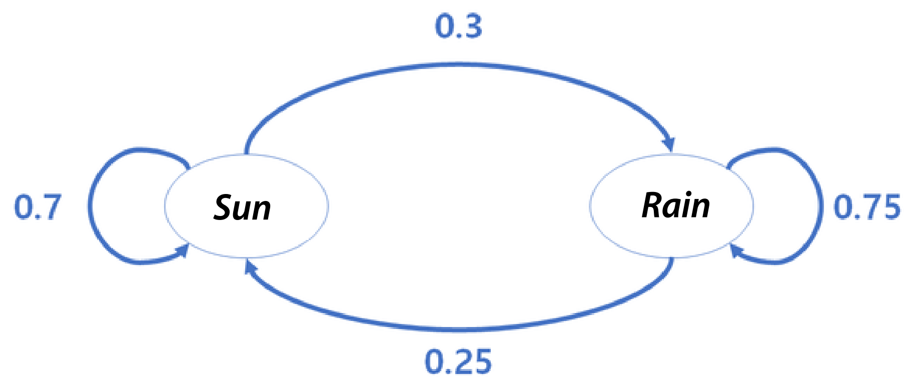


Markov Process

- ***Memoryless Property***

$$\Pr(X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = \Pr(X_{n+1} = x \mid X_n = x_n),$$

- ***Example***
  - ***Weather Model***



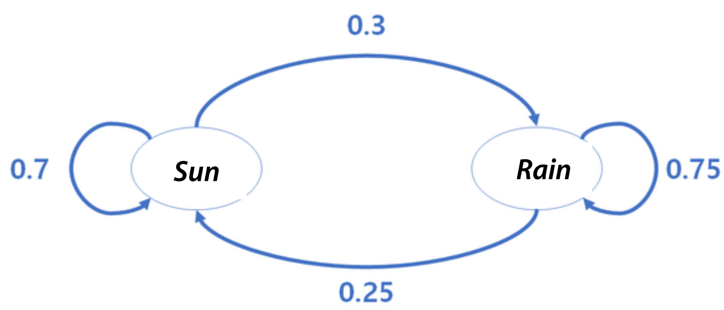
Definition. The *transition matrix* at time n is the matrix $P(n) = (p_{ij}(n))$, i.e. the (i,j)th element of $P(n)$ is $p_{ij}(n)$.[1] The transition matrix satisfies:

(i) $p_{ij}(n) \geq 0 \quad \forall i,j$ (the entries are non-negative)

(ii) $\sum_j p_{ij}(n) = 1 \quad \forall i$ (the rows sum to 1)

Markov Process

- *Example*
  - *Weather model*



$$P = \begin{bmatrix} 0.7 & 0.3 \\ 0.25 & 0.75 \end{bmatrix}$$

  - *When today is clear, what is the probability that tomorrow will be clear?*
    - ➢ *One-step transition probability*
  - *When today is clear, what is the probability that the day after tomorrow will be clear?*
    - ➢ *Two-step transition probability*
  - *When observed for a long time, what is the ratio of sunny days to cloudy days?*
    - ➢ *Stationary distribution*

Markov Process

- ***Stationary Assumption***
  - ***Transition probabilities are independent of time (t)***

$$p_{ij}(t) = p_{ij}$$

  - ***Time-homogeneity***

Given a Markov chain with transition probabilities P and initial condition $X_0 = i$, we know how to calculate the probability distribution of X_1; indeed, this is given directly from the transition probabilities. The natural question to ask next is: what is the distribution at later times? That is, we would like to know the n-step transition probabilities $P^{(n)}$, defined by

$$P_{ij}^{(n)} = P(X_n = j | X_0 = i). \tag{3}$$

For example, for $n = 2$, we have that

$$\begin{aligned}
P(X_2 = j | X_0 = i) &= \sum_k P(X_2 = j | X_1 = k, X_0 = i) P(X_1 = k | X_0 = i) && \text{Law of Total Probability} \\
&= \sum_k P(X_2 = j | X_1 = k) P(X_1 = k | X_0 = i) && \text{Markov Property} \\
&= \sum_k P_{kj} P_{ik} && \text{time-homogeneity} \\
&= (P^2)_{ij}
\end{aligned}$$

That is, the two-step transition matrix is $P^{(2)} = P^2$.

Markov Process

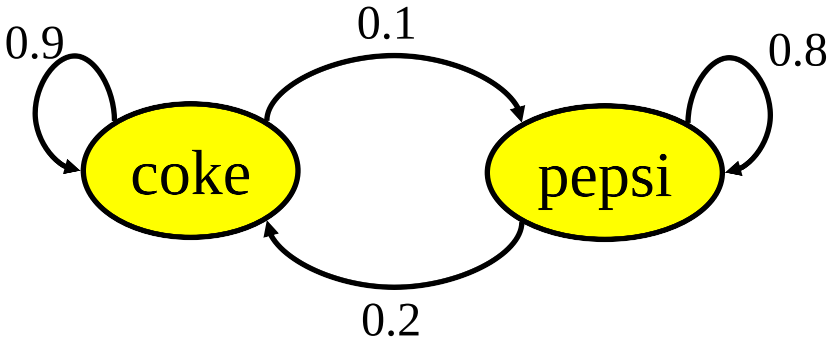
Coke vs. Pepsi Example

- 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:

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



단국대학교 DANKOOK UNIVERSITY

Markov Process

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

단국대학교 DANKOOK UNIVERSITY

Markov Process

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?

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

Markov Process

Coke vs. Pepsi Example (cont)

- Assume each person makes one cola purchase per week
- Suppose 60% of all people now drink Coke, and 40% drink Pepsi
- 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

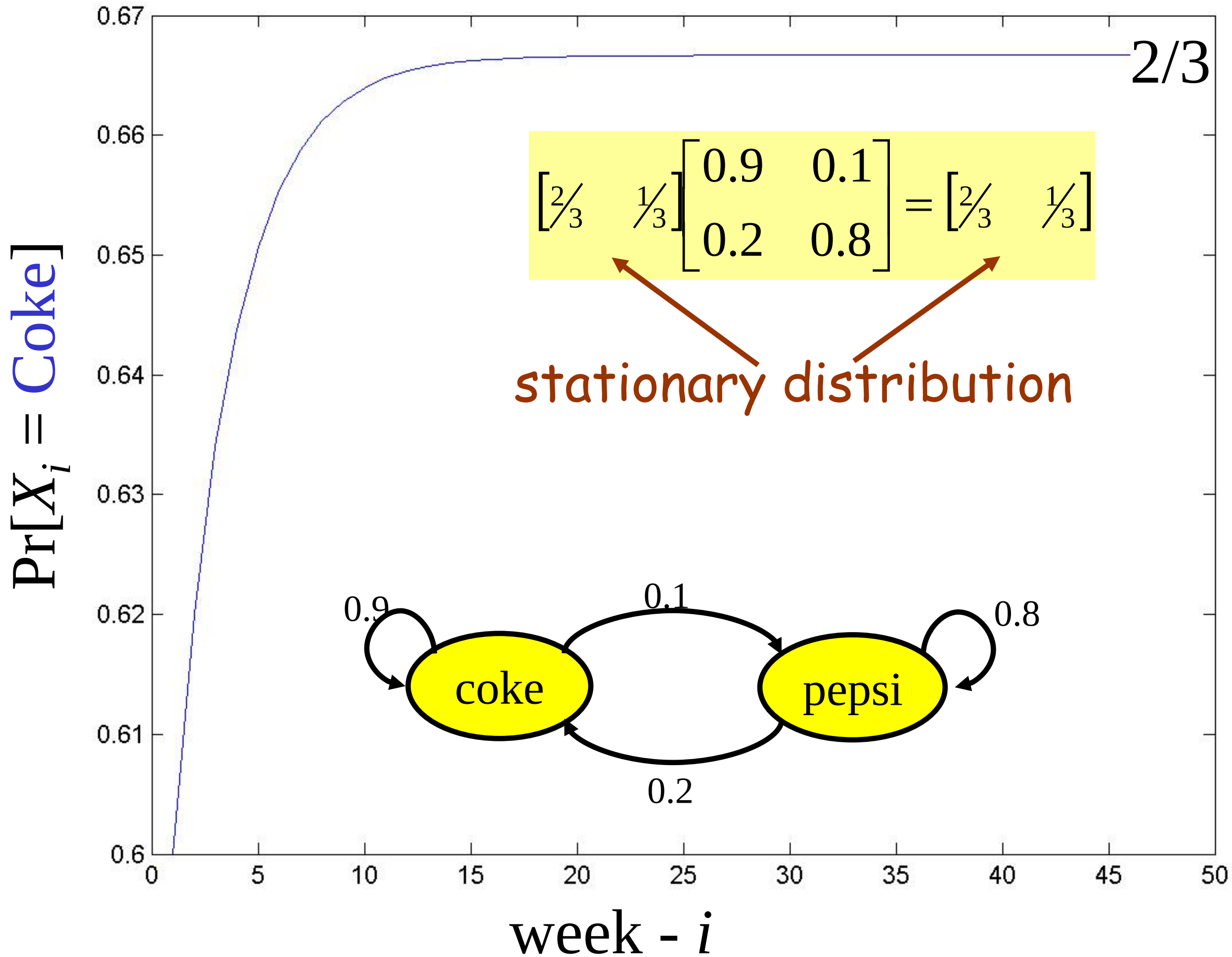
$Q_0 = (0.6, 0.4)$ - initial distribution

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

Markov Process

Coke vs. Pepsi Example (cont)

Simulation:



Markov Process

- *Stationary distribution*
  - *Long-term behavior and Probability distribution over states*
  - *Linear algebra connection*
    - ➢ *Is it an eigenvector of transition matrix P?*
- *Let's solve weather model*
  - *Let state space S={Sun, Rain} with Transition matrix*

$$P = \begin{array}{c} \\ \text{sun} \\ \text{rain} \end{array} \begin{array}{c} \begin{array}{cc} \text{sun} & \text{rain} \end{array} \\ \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix} \end{array}$$

n	P(sun)	P(rain)
0	0	1
1	0.4000	0.6000
2	0.5600	0.4400
3	0.6240	0.3760
4	0.6496	0.3504
5	0.6598	0.3402
6	0.6639	0.3361
7	0.6656	0.3344
8	0.6662	0.3338
9	0.6665	0.3335
10	0.6666	0.3334
11	0.6666	0.3334
12	0.6667	0.3333
13	0.6667	0.3333
14	0.6667	0.3333

Markov Process

- ***Ex2***
  - ***Consider a Markov chain on state space {0, 1} with transition matrix, and suppose the random walker starts at state 0.***

$$P = \begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix}$$

  - ***But, if we start with initial distribution (0.5, 0.5), then we obtain***

n	P(0)	P(1)
0	1	0
1	0	1
2	1	0
3	0	1
4	1	0
5	0	1
6	1	0
$\vdots$	$\vdots$	$\vdots$

Diverge!

n	P(0)	P(1)
0	0.5	0.5
1	0.5	0.5
2	0.5	0.5
$\vdots$	$\vdots$	$\vdots$

Converge!

- *Limiting and Stationary distributions*

In applications we are often interested in the long-term probability of visiting each state.

Definition. Consider a time-homogeneous Markov chain with transition matrix P. A row vector λ is a *limiting distribution* if $\lambda_i \geq 0$, $\sum_j \lambda_j = 1$ (so that λ is a probability distribution), and if, for every i,

$$\lim_{n\to\infty} (P^n)_{ij} = \lambda_j \qquad \forall j \in S.$$

In other words,

$$P^n \to \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \dots \\ \lambda_1 & \lambda_2 & \lambda_3 & \dots \\ \lambda_1 & \lambda_2 & \lambda_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \qquad \text{as } n \to \infty.$$

$$\lambda P = \left(\lim_{n\to\infty} P^n_{i,\cdot}\right) P = \left(\lim_{n\to\infty} P^{n+1}_{i,\cdot}\right) = \lambda$$

Markov Process

- ***Stationary distributions***
  - *Given a Markov chain with transition matrix P, a stationary distribution is a probability distribution π which satisfies*

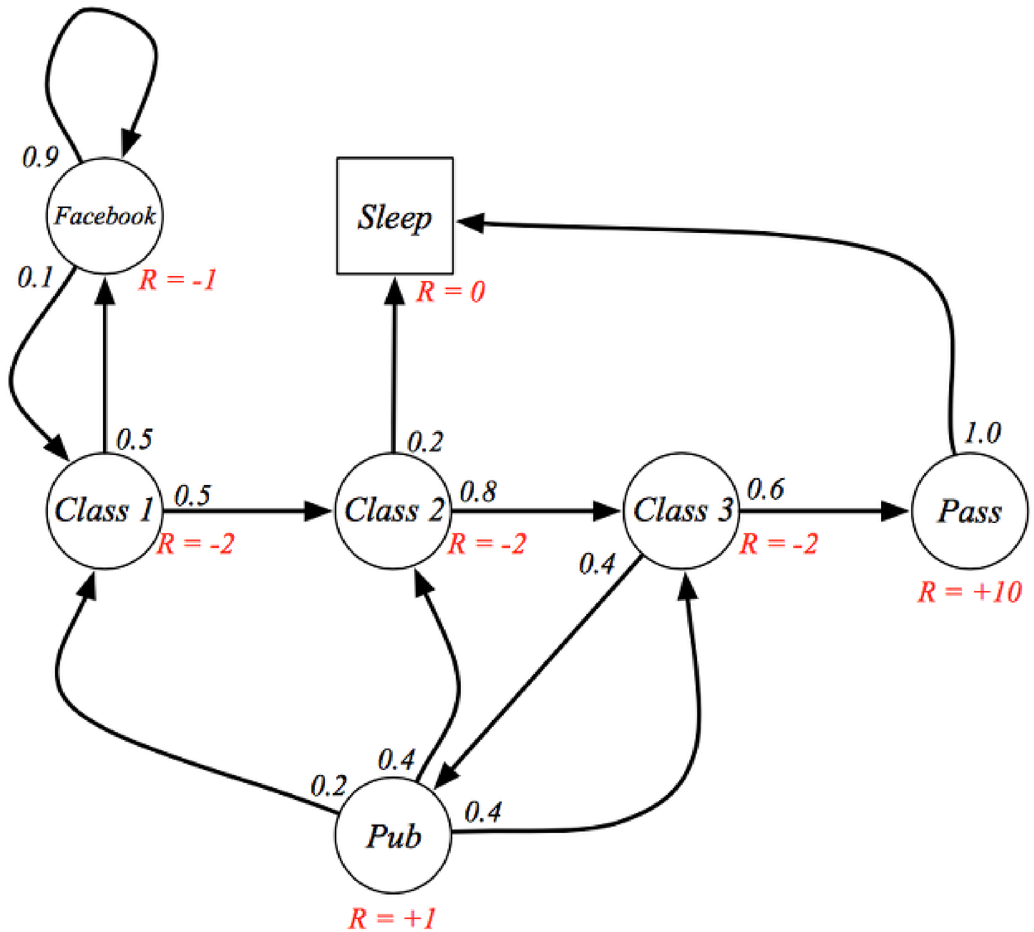
$$\pi = \pi P \qquad \Longleftrightarrow \qquad \boxed{\pi_j = \sum_i \pi_i P_{ij} \quad \forall\, j.}$$

Balance equation

  - $\sum_i \pi_i = 1$

Markov Process

- *More complicated ones*



Markov Process

- *Operation of Wi-Fi (IEEE 802.11)*