인공지능 실습

Chapter 7. Bayesian Network

포항공과대학교 컴퓨터공학과

목차

7.1. 기본 확률 이론

Probability Space $(\Omega, \mathcal{F}, P)$


- Experiment[실험] (or trial[시행]): outcome을 갖는 실험
  - 예. 동전 두 번 던지기
- Outcome[결과]: experiment의 결과
  - 예. (Head, Tail)
- Sample space[표본공간] Ω(omega): experiment에서 발생 가능한 모든 outcome의 집합
  - 예. {(H, H), (H, T), (T, H), (T, T)}
- Event[사건] $\mathcal{F}$: experiment의 일부 outcome의 집합
  - 예. {(H, T), (T, H)} (head가 한 번 나온 outcome의 집합)
- Probability measure P: event에 대한 확률
  - 예. P({(H, T), (T, H)}) = 0.5

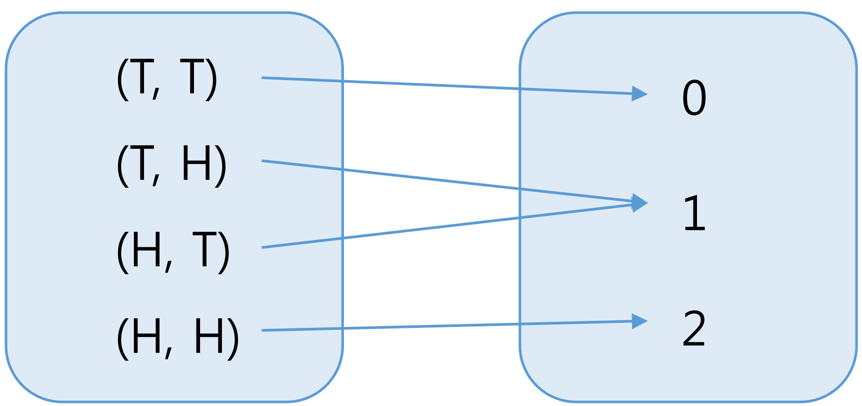
Random Variable 확률변수

- Experiment의 outcome을 우리가 관심 있는 형태로 변환하는 **함수**
  - “Contrary to its name, this procedure itself is neither random nor variable” by Wikipedia
- A random variable $\mathrm{X}: \Omega \rightarrow E$
  - E: measureable space (Usually $E = \mathbb{R}$)

동전을 두 번 던지는 experiment

Sample space Ω

Random variable $X: \Omega \rightarrow \mathbb{N}$
head의 개수



P({(H, T), (T, H)}) = 0.5

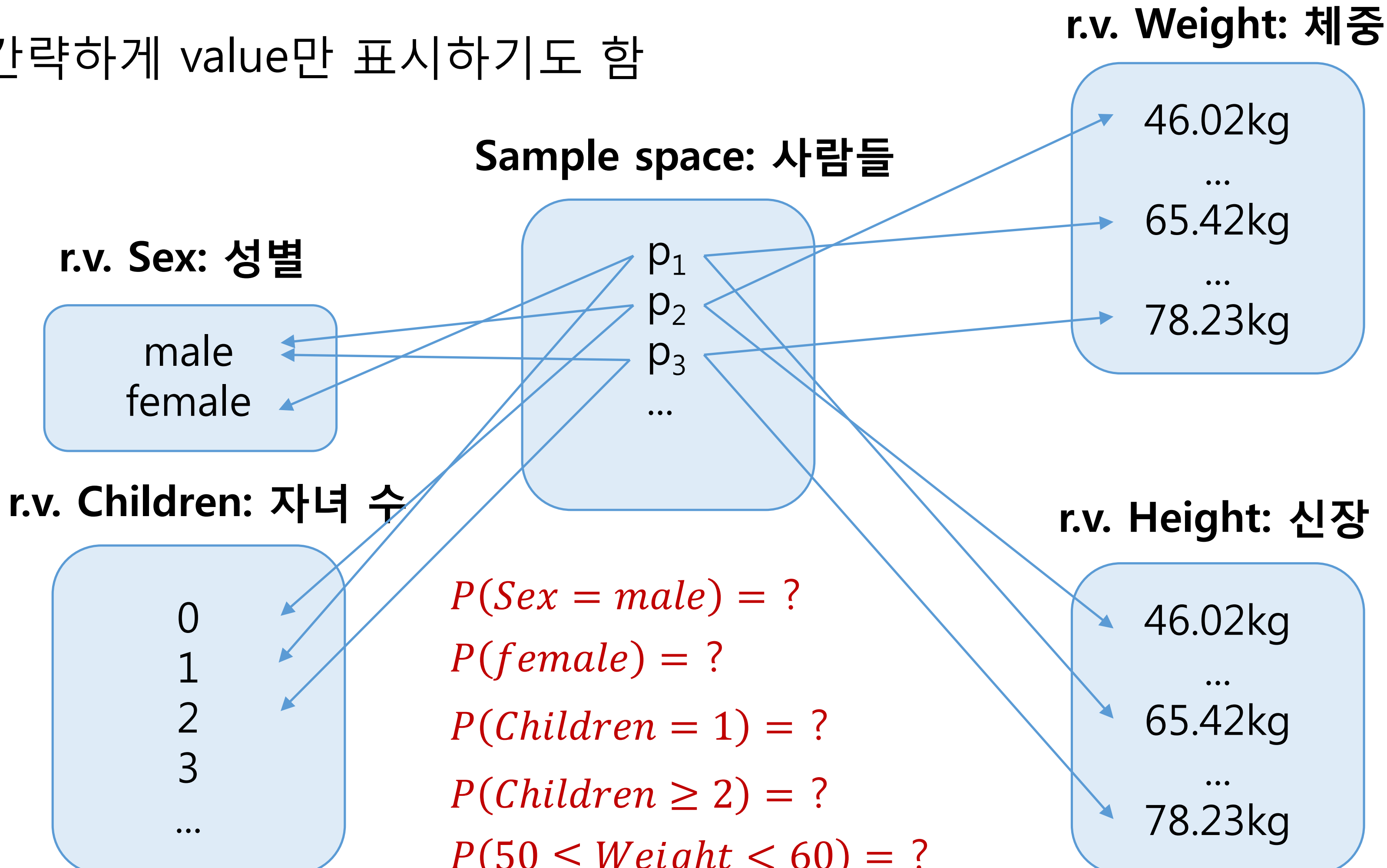
$\Rightarrow$ $P(X = 1) = 0.5$

P({(H, T), (T, H), (H, H)}) = 0.75

$\Rightarrow$ $P(X \geq 1) = 0.75$

Random Variable 확률변수

- Random variable은 대문자, value는 소문자로 표시
- 간략하게 value만 표시하기도 함



$P(Sex = male) = ?$

$P(female) = ?$

$P(Children = 1) = ?$

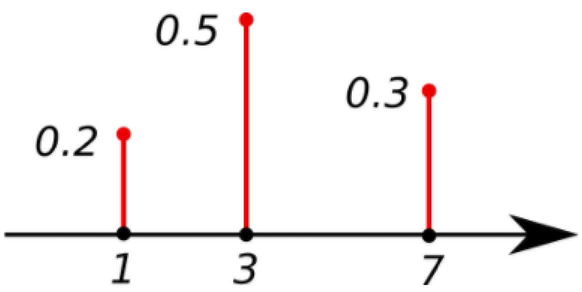
$P(Children \geq 2) = ?$

$P(50 \leq Weight < 60) = ?$

Random Variable and Probability Distribution

- **Discrete** random variable 이산확률변수
  - 예. 성별 (finite or categorical)
  - 예. 자녀 수 (infinite)

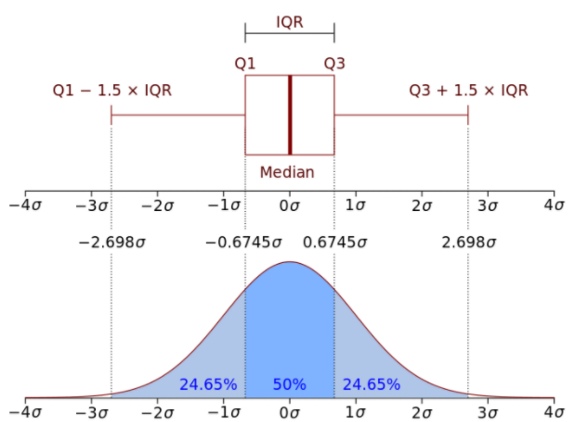
Probability mass function
확률 질량 함수



- **Continuous** random variable 연속확률변수
  - 예. 체중, 신장

- Single value v.s. range

Probability density function
확률 밀도 함수



Random Variable and Probability Distribution

- Discrete random variable에 대한 probability distribution (pmf)은 value에 대한 확률의 리스트로서 표현 가능
- 예. **P**(Coin)=<0.5, 0.5>

$Coin$	$\boldsymbol{P}(Coin)$
$head$	0.5
$tail$	0.5

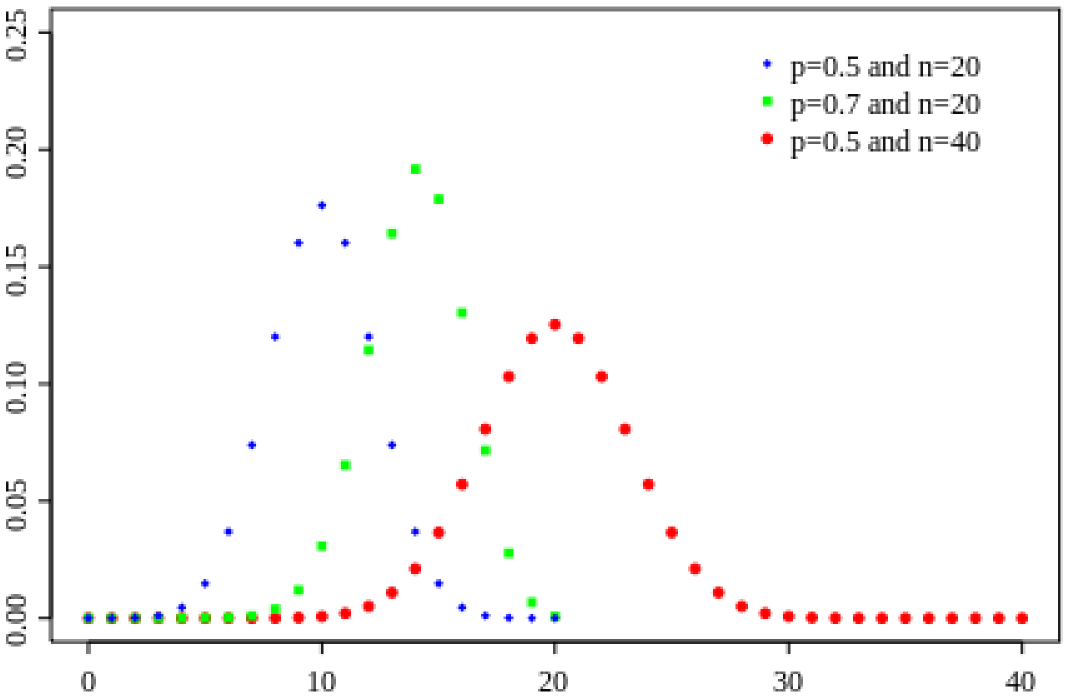
- 예. **P**(Weather)=<0.6, 0.1, 0.29, 0.01>

$Weather$	$\boldsymbol{P}(Weather)$
$sunny$	0.6
$rain$	0.1
$cloudy$	0.29
$snow$	0.01

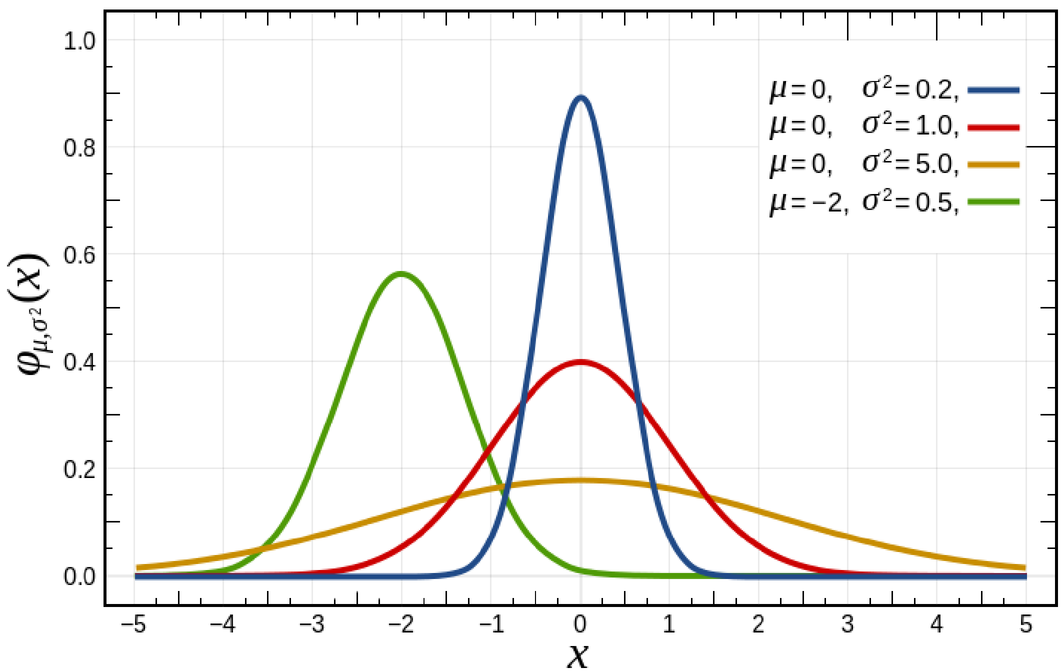
확률 및 통계 관련 표기법

https://en.wikipedia.org/wiki/Notation_in_probability_and_statistics

- Binomial distribution: $B(n, p)$
  - n: number of trials
  - p: success probability in each trial
  - 예. 동전 던지기, 압정 던지기



- Normal (or Gaussian) distribution 정규분포 : $N(\mu, \sigma)$
  - Standard normal distribution 표준정규분포: $N(0, 1)$
  - 예. 사람의 키, 체중



z

표준정규분포표

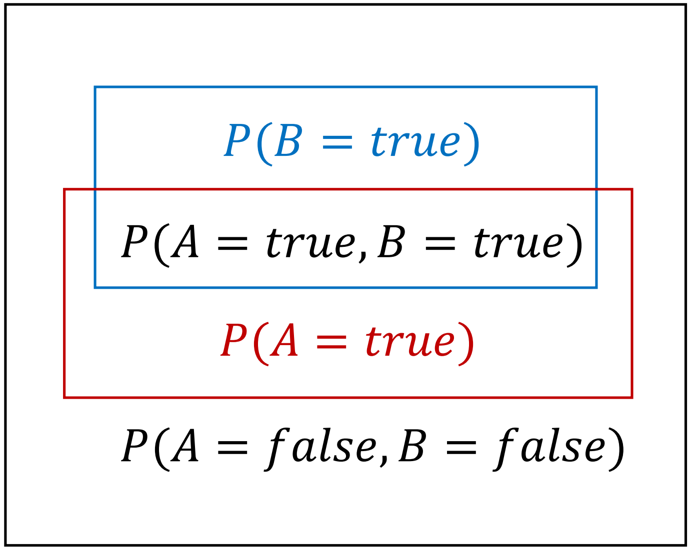
z	0.00	0.02	…
0.0	0.500000	0.503989	
0.1	0.539828	0.543795	
0.2	0.579260	0.583166	
0.3	0.617911	0.621720	
0.4	0.655422	0.659097	
0.5	0.691462	0.694974	
…			

Joint Prob., Conditional Prob.

- Conditional probability^조건부확률^: $\boldsymbol{P}(A|B)$
    - “Probability of A conditioned on B”
    - “Probability of A given B”

- Joint probability^결합확률^: $\boldsymbol{P}(A,B)$

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



Full Joint Distribution

Full joint distribution for Toothache (치통), Cavity (충치), Catch world

$Toothache$	$Cavity$	$Catch$	$\boldsymbol{P}(Toothache, Cavity, Catch)$
$toothache$	$cavity$	$catch$	0.108
		$\neg\ catch$	0.012
	$\neg\ cavity$	$catch$	0.016
		$\neg\ catch$	0.064
$\neg\ toothache$	$cavity$	$catch$	0.072
		$\neg\ catch$	0.008
	$\neg\ cavity$	$catch$	0.144
		$\neg\ catch$	0.576

=

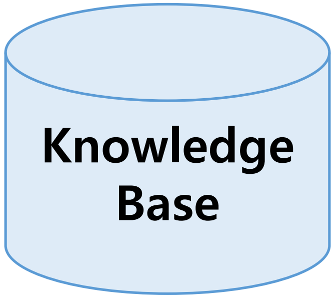
	$toothache$		$\neg\ toothache$	
	$catch$	$\neg\ catch$	$catch$	$\neg\ catch$
$cavity$	0.108	0.012	0.072	0.008
$\neg\ cavity$	0.016	0.064	0.144	0.576

Inference 추론

- "Moving from **premises** to **conclusions**" from Wikipedia

  전제 (알려진 사실) 결론

- 예. 3단 논법
  - 대전제: "모든 사람은 죽는다"
  - 소전제: "소크라테스는 사람이다"
  - 결론: "따라서 소크라테스는 죽는다"



- 현실의 많은 문제들은 규칙(논리)만으로는 표현 할 수 없음
  - $Toothache \Rightarrow Cavity$ (?)
  - "소크라테스는 치통이 있다"
  - "따라서 소크라테스는 충치가 있다" (?)

Probabilistic Inference

- Uncertainty를 다루기 위한 **probabilistic inference**
  - 예. $P(cavity|toothache)$
  - "Computation of posterior probabilities for query propositions given observed evidence." from AIMA

- **Full joint distribution**을 질문에 대한 정답을 찾는 knowledge base로서 사용 할 수 있음

	$toothache$		$\neg\ toothache$	
	$catch$	$\neg\ catch$	$catch$	$\neg\ catch$
$cavity$	0.108	0.012	0.072	0.008
$\neg\ cavity$	0.016	0.064	0.144	0.576

Inference using Full Joint Distribution

- Marginalization (or summing out)

$$\boldsymbol{P}(Y) = \sum_{z \in Z} \boldsymbol{P}(Y, Z)$$

(Z is marginalized out)

	$toothache$		$\neg\, toothache$	
	$catch$	$\neg\, catch$	$catch$	$\neg\, catch$
$cavity$	0.108	0.012	0.072	0.008
$\neg\, cavity$	0.016	0.064	0.144	0.576

Marginal probability

$$P(cavity) = \sum_{z \in \{Catch, Toothache\}} P(cavity, z)$$

$$= 0.108 + 0.012 + 0.072 + 0.008 = 0.2$$

Marginal probability distribution

$Cavity$	$\boldsymbol{P}(Cavity)$
$cavity$	0.2
$\neg\, cavity$	0.8

Inference using Full Joint Distribution

- Conditioning

$$\boldsymbol{P}(Y|Z=z) = \frac{\boldsymbol{P}(Y, Z=z)}{\boldsymbol{P}(Z=z)}$$

	$toothache$		$\neg\, toothache$	
	$catch$	$\neg\, catch$	$catch$	$\neg\, catch$
$cavity$	0.108	0.012	0.072	0.008
$\neg\, cavity$	0.016	0.064	0.144	0.576

Conditional probability

$$P(cavity|toothache) = \frac{P(cavity \wedge toothache)}{P(toothache)}$$

$$= \frac{0.108 + 0.012}{0.108 + 0.012 + 0.016 + 0.064} = 0.6$$

$$P(\neg\, cavity|toothache) = \frac{P(\neg\, cavity \wedge toothache)}{P(toothache)}$$

$$= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

Conditional probability distribution

$Cavity$	$\boldsymbol{P}(Cavity|toothache)$
$cavity$	0.6
$\neg\, cavity$	0.4

※ $P(toothache)$는 상수이므로 normalization 가능

Inference using Full Joint Distribution

- Conditioning with **normalization**

$$\boldsymbol{P}(Y|Z=z)=\frac{\boldsymbol{P}(Y,Z=z)}{\boldsymbol{P}(Z=z)}$$

	$toothache$		$\neg\, toothache$	
	$catch$	$\neg\, catch$	$catch$	$\neg\, catch$
$cavity$	0.108	0.012	0.072	0.008
$\neg\, cavity$	0.016	0.064	0.144	0.576

Conditional probability

$$P(cavity|toothache)=\alpha\ P(cavity \land toothache)$$
$$=0.108+0.012=0.012$$

$$P(\neg\, cavity|toothache)=\alpha\ P(\neg\, cavity \land toothache)$$
$$=0.016+0.064=0.08$$

Conditional probability distribution

$Cavity$	$\boldsymbol{P}(Cavity\|toothache)$
$cavity$	0.6
$\neg\, cavity$	0.4

$$\alpha\ P(cavity \land toothache)+\alpha\ P(\neg\, cavity \land toothache)=1$$

$$\alpha\ 0.012+\alpha\ 0.08=1$$

$$\alpha=5$$

$P(toothache)$를 몰라도 $P(Cavity|toothache)$ 계산 가능

Full Joint Distribution의 한계

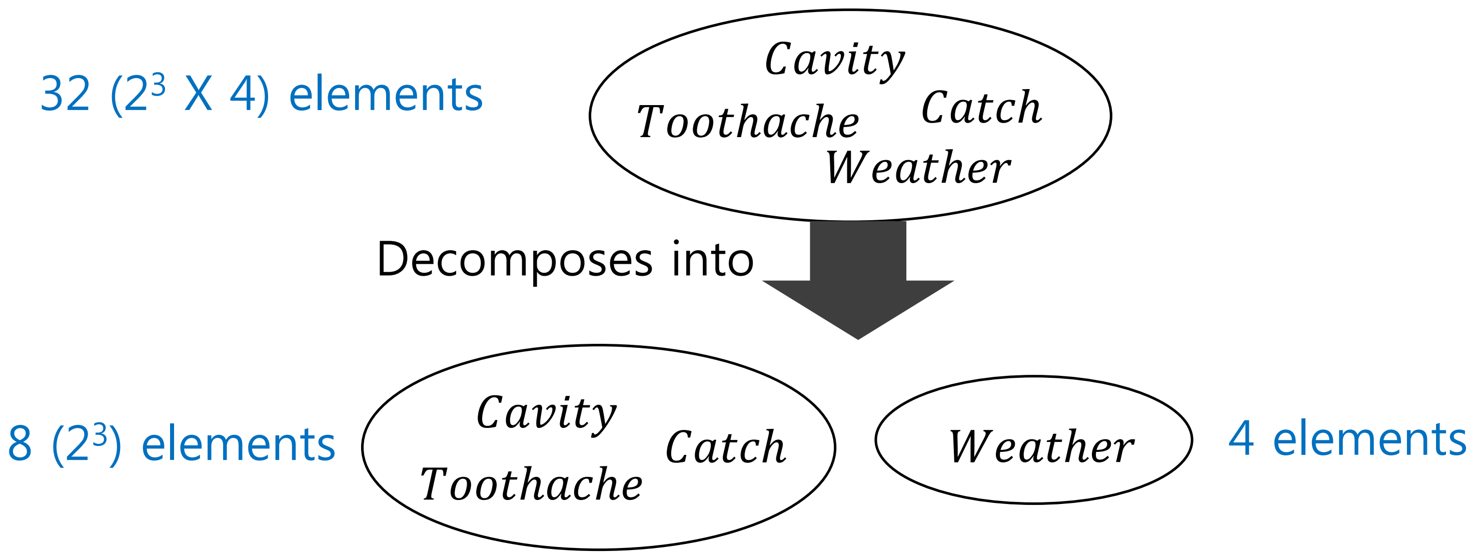
- Toothache, Cavity, Catch world
    - 3개의 Boolean random variable
    - 총 8(2^3)개의 probability 저장 및 처리

- 현실의 문제
    - n개의 Boolean random variable ($n > 100$)
    - 총 $O(2^n)$ 크기의 테이블 및 처리 시간

- Full joint distribution은 현실의 reasoning 문제를 풀기에 부적합

Independence

- Toothache, Cavity, Catch + Weather world



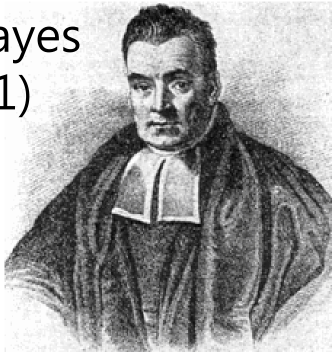
- Independence between variables X and Y can be written as

$$P(X|Y) = P(X) \text{ or } P(Y|X) = P(Y) \text{ or } P(X \land Y) = P(X)P(Y)$$

Bayes' Theorem (or Bayes' Rule)

- Variables: hypothesis H and evidence E

Thomas Bayes
(1701-1761)



$$\underbrace{\boldsymbol{P}(H|E)}_{\text{Posterior (want)}} = \frac{\overbrace{\boldsymbol{P}(H)}^{\text{Prior (have)}}\overbrace{\boldsymbol{P}(E|H)}^{\text{Likelihood (have)}}}{\underbrace{\boldsymbol{P}(E)}_{\text{Marginal likelihood}}} \propto \boldsymbol{P}(H)\boldsymbol{P}(E|H)$$

$$\boldsymbol{P}(Cavity|toothache \wedge catch) = \alpha\, \boldsymbol{P}(Cavity)\, \underbrace{\boldsymbol{P}(toothache \wedge catch|Cavity)}_{\text{역시 joint distribution 계산이 필요}}$$

Conditional Independence

- Conditional independence of two variables X and Y, given a third variable Z, is

$$\boldsymbol{P}(X, Y|Z) = \boldsymbol{P}(X|Z)\ \boldsymbol{P}(Y|Z)$$

$$\boldsymbol{P}(Cavity|toothache \wedge catch) = \alpha\ \boldsymbol{P}(Cavity)\ \boldsymbol{P}(toothache \wedge catch|Cavity)$$

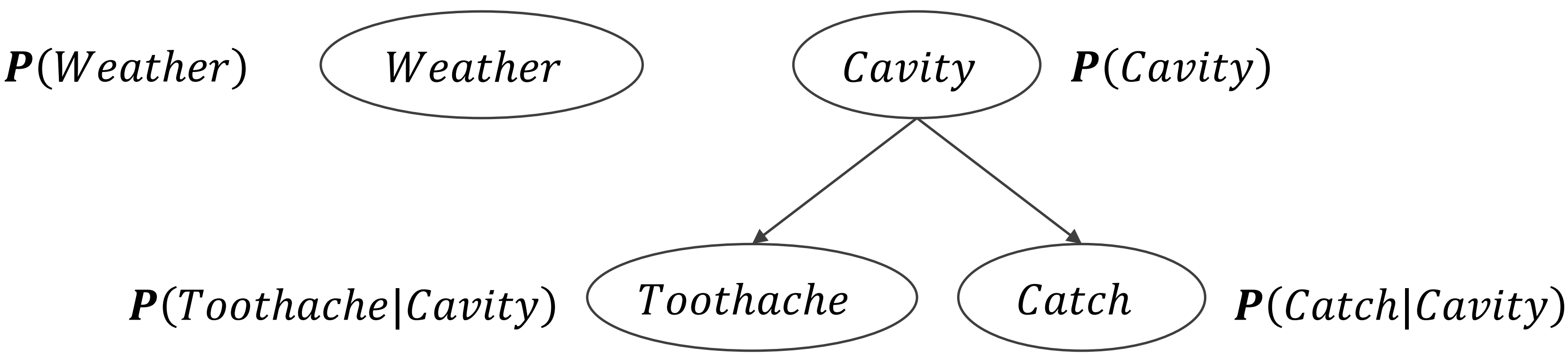
$$\Downarrow$$

$$\boldsymbol{P}(toothache \wedge catch|Cavity) = \boldsymbol{P}(toothache|Cavity)\ \boldsymbol{P}(catch|Cavity)$$

$$\boldsymbol{P}(Cavity|toothache \wedge catch) = \alpha\ \boldsymbol{P}(Cavity)\ \boldsymbol{P}(toothache|Cavity)\ \boldsymbol{P}(catch|Cavity)$$

Conditional Independence

- Conditional independence assertions can allow probabilistic systems to scale up

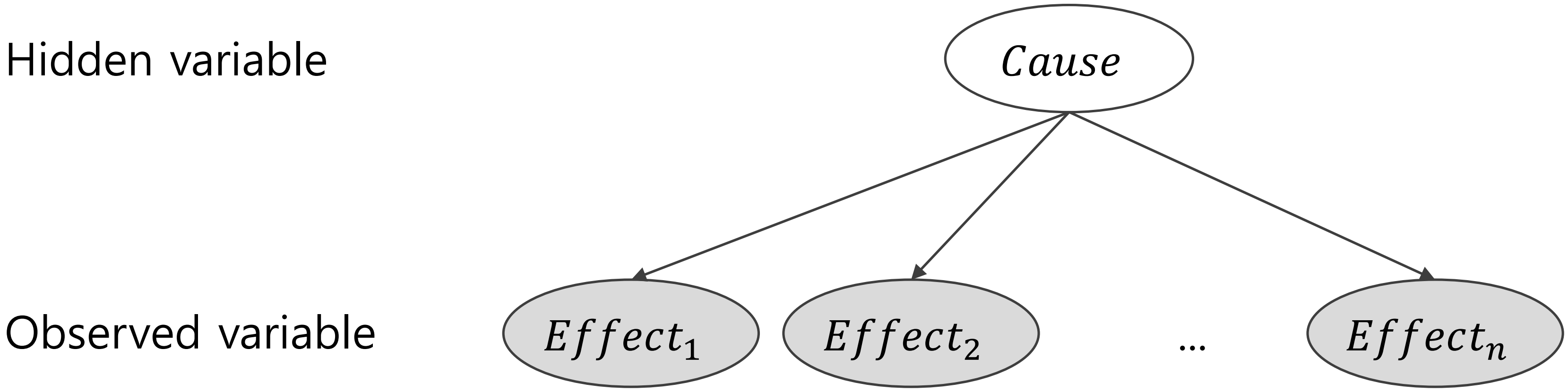


7.2. Naive Bayes Model

Naive Bayes Model

- Naive Bayes model
  - Single cause directly influences a number of effects

$$\boldsymbol{P}(Cause, Effect_1, \cdots, Effect_{\mathrm{n}}) = \boldsymbol{P}(Cause) \prod_i \boldsymbol{P}(Effect_i | Cause)$$



Naive Bayes Classifier

- Posterior

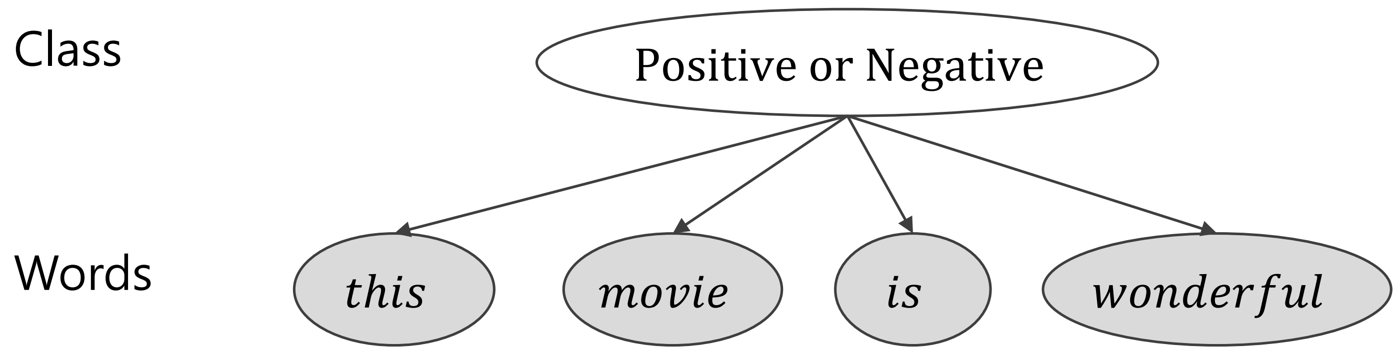
$$\boldsymbol{P}(Cause|Effect_1,\cdots,Effect_n) \propto \boldsymbol{P}(Cause)\boldsymbol{P}(Effect_1,\cdots,Effect_n|Cause)$$

$$\propto \boldsymbol{P}(Cause)\textstyle\prod_i \boldsymbol{P}(Effect_i|Cause)$$

- Classifier

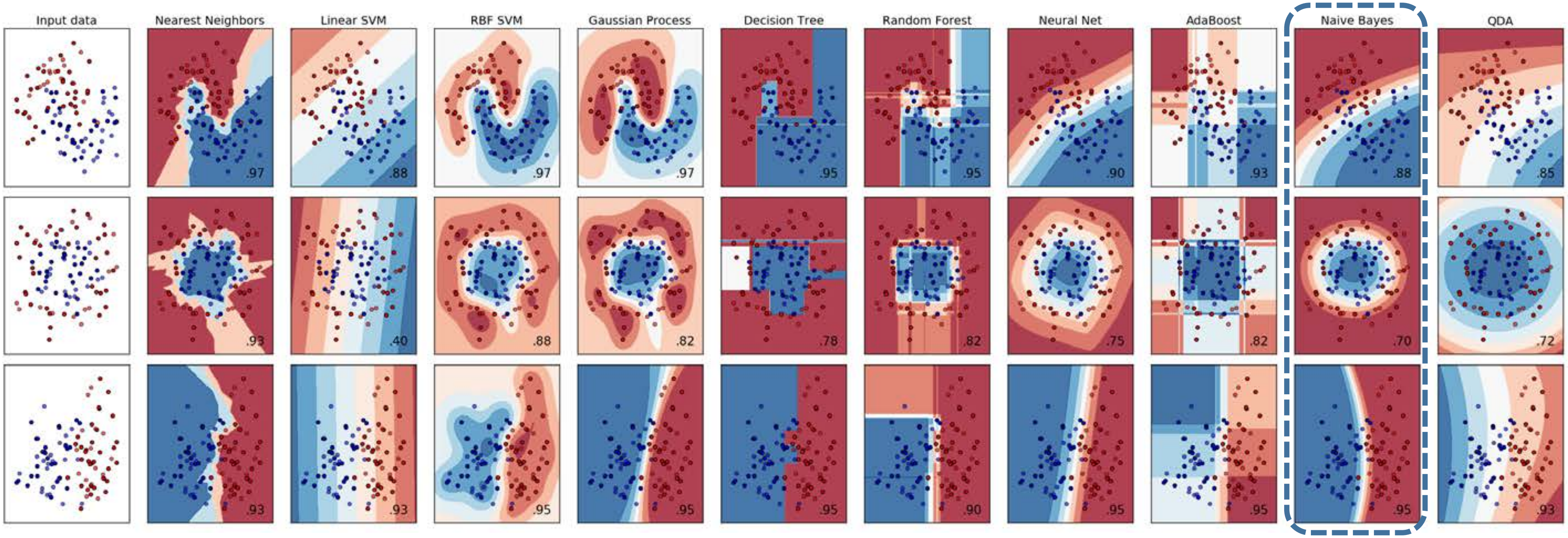
$$\hat{y} = f_{NB}(x) = \operatorname*{argmax}_y p(y)\prod_i p(x_i|y)$$

- 예. Text Classification



Classifier Comparison

- Naive Bayes classifier is one of them!



http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
...
from sklearn.naive_bayes import GaussianNB
```

충치 진단 시스템 구현

- **다음 확률 계산**
  - Cavity에 대한 사전 확률 (prior)
  - Toothache의 Cavity에 대한 조건부 확률
  - Catch의 Cavity에 대한 조건부 확률

$$\boldsymbol{P}(Cavity|Toothache, Catch) \propto \boldsymbol{P}(Cavity)\ \boldsymbol{P}(Toothache|Cavity)\ \boldsymbol{P}(Catch|Cavity)$$

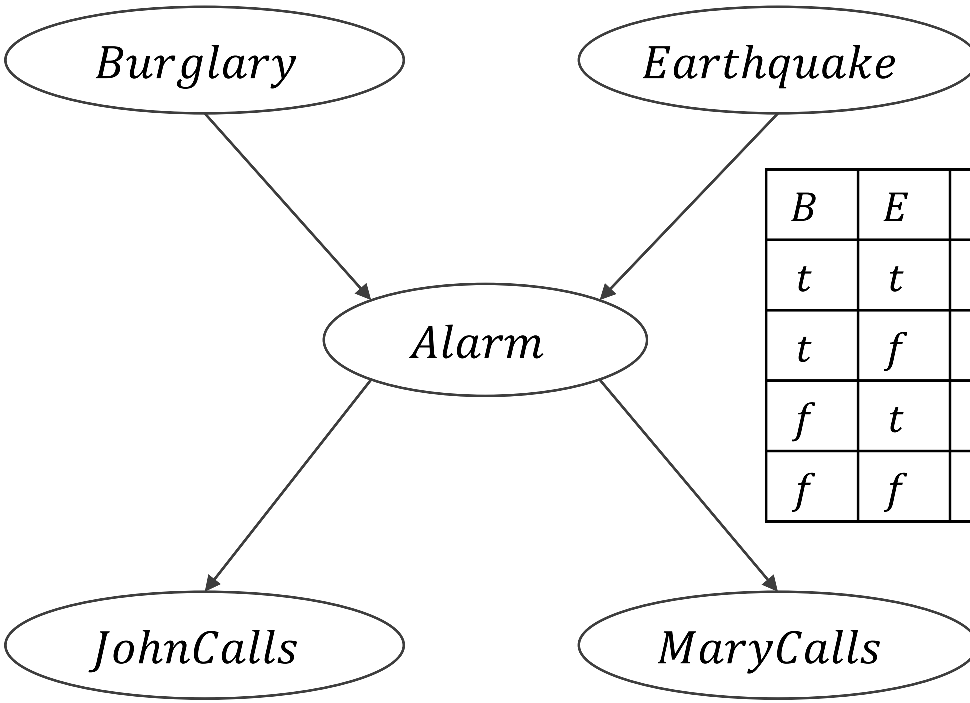
- `cavity_diagnosis.py`의 **compute_cavity_prob 함수 구현**
  - 주어진 evidence를 바탕으로 cavity 확률 계산
  - 특정 evidence가 없는 경우에 대해서도 처리

7.3. Bayesian Network

Bayesian Network

$P(B = t)$
0.001

$P(E = t)$
0.002



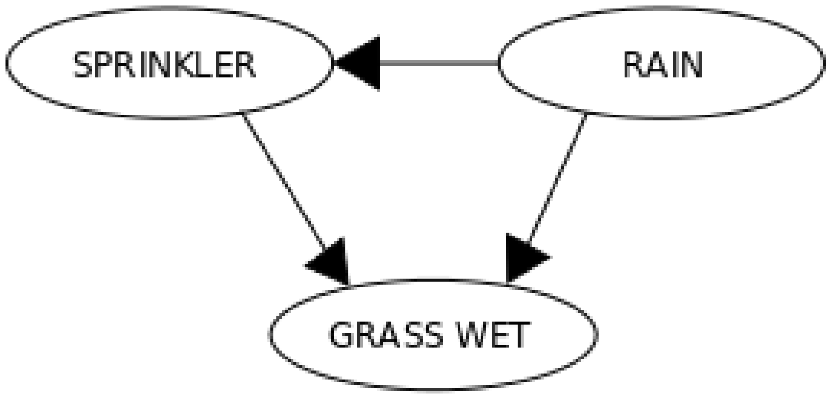
B	E	$P(A = t \mid B, E)$
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	$P(J = t \mid A)$
t	0.90
f	0.05

A	$P(M = t \mid A)$
t	0.70
f	0.01

Bayesian Network

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

A simple Bayesian network with conditional probability tables (from Wikipedia)

Bayesian Network

- Let $X = (X_1, \cdots, X_n)$ be random variables
- A Bayesian network is a directed acyclic graph (DAG) that specifies a joint distribution over X as product of local conditional distributions, one for each node:

$$\mathbb{P}(X_1 = x_1, \cdots, X_n = x_n) \stackrel{\text{def}}{=} \prod_{i=1}^{n} p(x_i | parents(X_i))$$

Judea Pearl
Turing Award Winner

Q. Bayesian Network 그리기

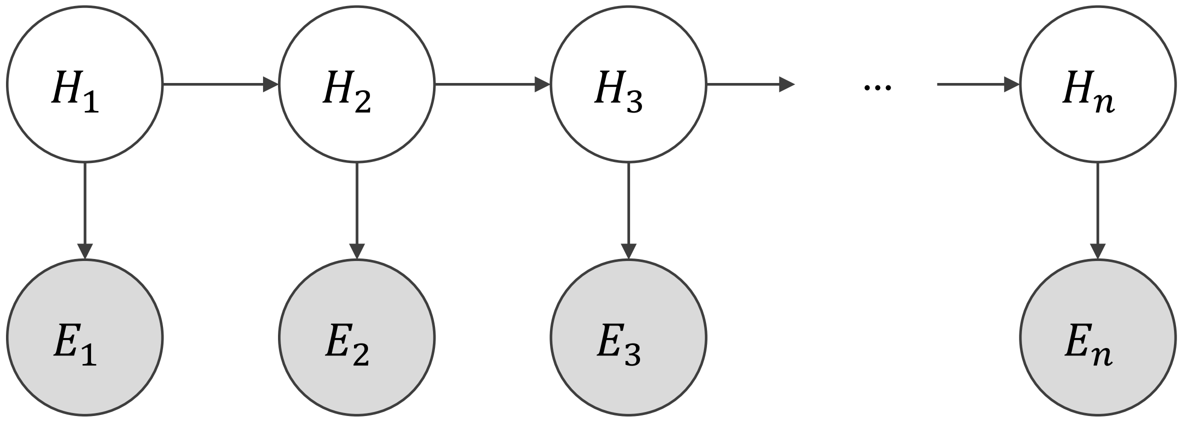
- In your local nuclear power station, there is an alarm that senses when a temperature gauge exceeds a given threshold. The gauge measures the temperature of the core. Consider the Boolean variables A (alarm sounds), F_A (alarm is faulty), and F_G (gauge is faulty) and the multivalued nodes G (gauge reading) and T (actual core temperature).
- Draw a Bayesian network for this domain, given that the gauge is more likely to fail when the core temperature gets too high.

Hidden Markov model (HMM)

For each time step t = 1, …, T

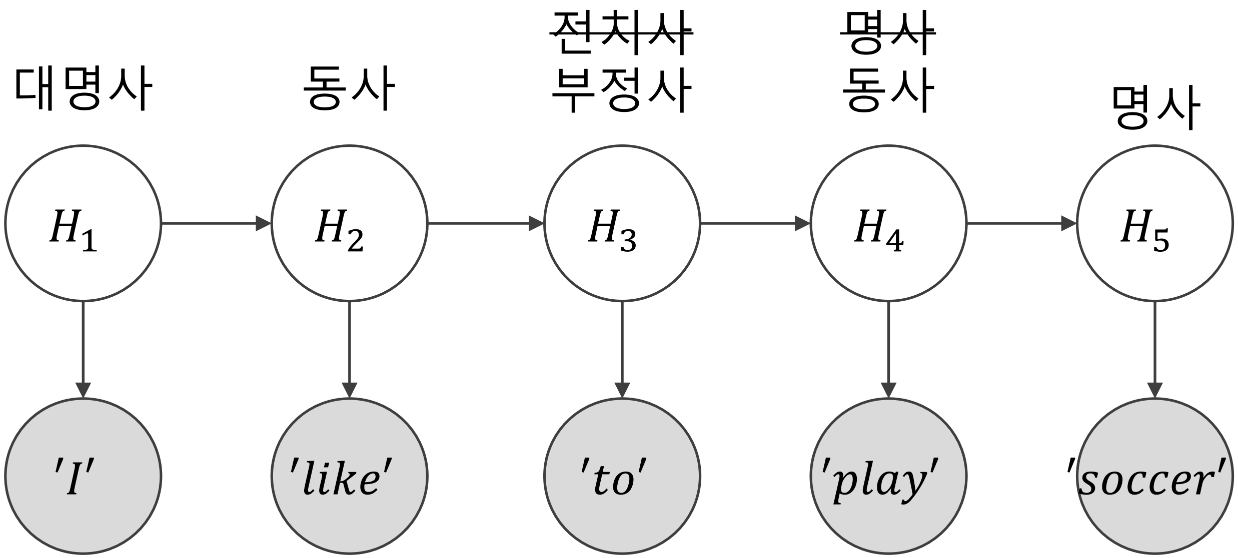
Generate object location $H_t \sim p(H_t|H_{t-1})$

Generate sensor location $H_t \sim p(E_t|H_t)$



Hidden Markov model (HMM)

- 예. Part-of-speech (품사) tagging

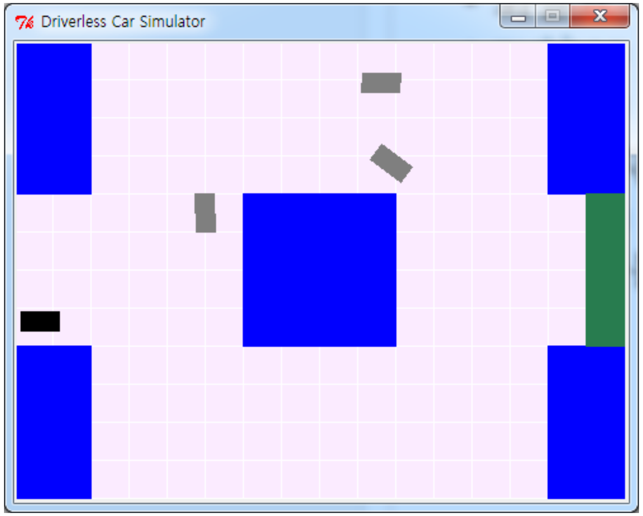


- HMM + Viterbi algorithm

7.4. Car Tracking

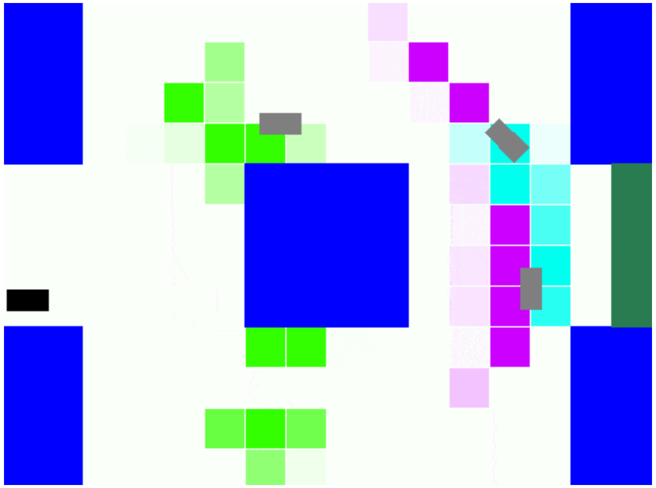
Drive Your Car!

- 출발지점에서 목표지점까지 이동
- 수동으로 플레이
  - 화살표키로 이동 (후진 불가능)
  - `python drive.py -i none -d`
  - `python drive.py -i none`
- 무인 자동차
  - `python drive.py -i none -d -a`
  - 다른 자동차는 없다고 가정하고 목표를 향해 감



Car Tracking Problem

- 2차원의 grid 공간에서 K개의 다른 차가 이동
- 각 time step *t*마다 센서(예. microphone)로 Agent와 다른 **각각**의 차의 **거리**를 측정
- 센서 정보에는 noise를 포함
- 목표: 센서 정보를 바탕으로 다른 자동차의 위치를 **tracking**
- 위치 정보를 바탕으로 운전하는 AI는 제공

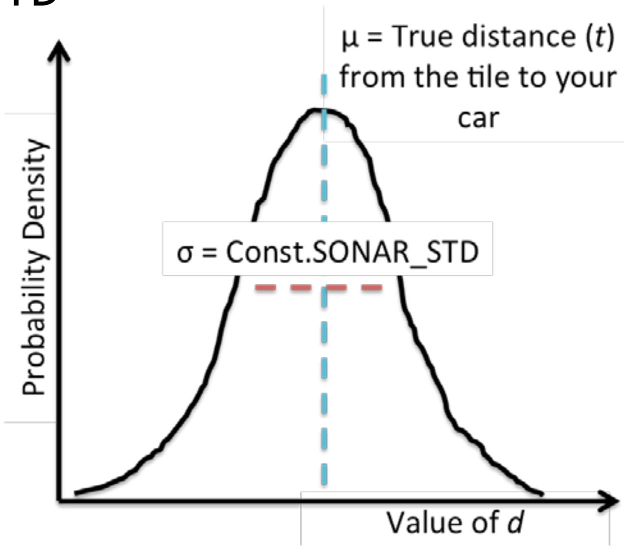


Car Tracking Problem

- `python drive.py`
  - -a: Enable autonomous driving (as opposed to manual)
  - -i <inference method>: Use none, exactInference, particleFilter to compute the belief distributions
  - -l <map>: Use this map (e.g. small or lombard)
  - -d: Debug by showing all the cars on the map
  - -p: All other cars remain parked (so that they don't move)
  - -k <number>: Number of other cars

Car Tracking Problem

- $C_t \in \mathbb{R}^2$: t 시간 다른 자동차의 실제 위치 (unobserved)
- $p(c_t|c_{t-1})$: 다른 자동차의 움직임에 대한 distribution
- $a_t \in \mathbb{R}^2$: t 시간 본인 자동차의 실제 위치
- D_t: Gaussian random variable $D_t \sim N(||a_t - C_t||, \sigma^2)$
  - σ: Const.SONAR_STD



예. $a_t = (1,3)$ $C_t = (4,7)$ $||a_t - C_t|| = 5$ $D_t = 4.6 \text{ or } 5.2, \text{etc.}$

Car Tracking Problem

- 두 종류의 uncertainty
  - Non-deterministic action (MDP에서 다룸)
  - Partially observable state: 정확한 state가 아닌 state에 대한 부정확한 **observation**[관측값]만을 가짐

- Observation을 바탕으로 **belief state**[신뢰상태]를 계산해야 함
  - Posterior probability

$$\mathbb{P}(C_t | D_1 = d_1, \cdots, D_t = d_t)$$

예.
$p\big(C_t = (2,1)\big) = 0.0$
$p\big(C_t = (2,2)\big) = 0.125$
$p\big(C_t = (2,3)\big) = 0.5$

(4,1)	(4,2)	(4,3)	(4,4)
(3,1)	(3,2)	(3,3)	(3,4)
(2,1)	(2,2)	(2,3)	(2,4)
(1,1)	(1,2)	(1,3)	(1,4)

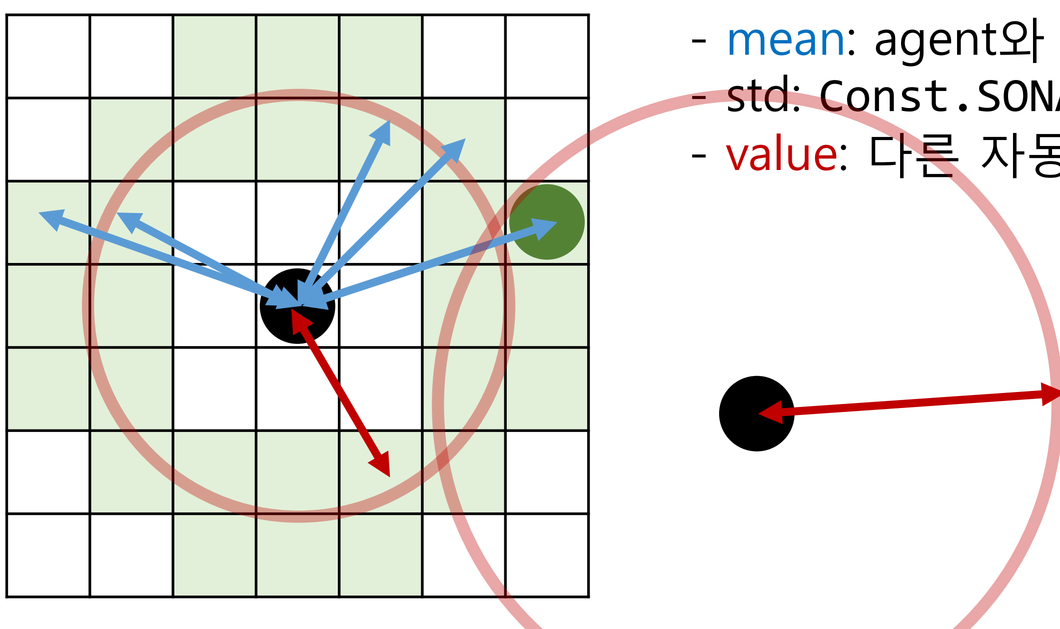
Car Tracking Problem

- 문제를 단순화하기 위해 world를 tile 형태로 discretize[이산화]
  - (row, col) pairs
    - 0 ≤ row < numRows and 0 ≤ col < numCols
  - 관련 함수: `util.rowToY(row)` 및 `util.colToX(col)`

- Tracker class의 멤버변수 `self.belief`
  - `self.belief.getProb(row, col)`
  - `self.belief.setProb(row, col, p)`
  - `self.belief.addProb(row, col, p)`

Car Tracking Problem

- `util.pdf(mean, std, value)`
  - (mean, std)의 정규분포 PDF를 통해 value의 확률 계산
  - 이 PDF는 sum-to-1의 확률을 return하지 않으나 확률 처럼 사용해 belief state를 계산
  - 모든 계산을 마치고 normalize 수행
    - `self.belief.normalize()`



- mean: agent와 각 tile간의 거리
- std: `Const.SONAR_STD`
- value: 다른 자동차와의 거리 (with noise)

Emission Probability

- 다른 자동차가 멈춰 있는 환경 ($C_t = C_{t-1}$)

- 오직 observation만을 바탕으로 belief state tracking

- C_t의 posterior probability를 계산하는 **ExactInference 클래스의 `observe` 함수 구현**

$$\mathbb{P}(C_t | D_1 = d_1, \cdots, D_t = d_t) \propto \mathbb{P}(C_t | D_1 = d_1, \cdots, D_{t-1} = d_{t-1}) p(d_t | c_t)$$

Emission Probability

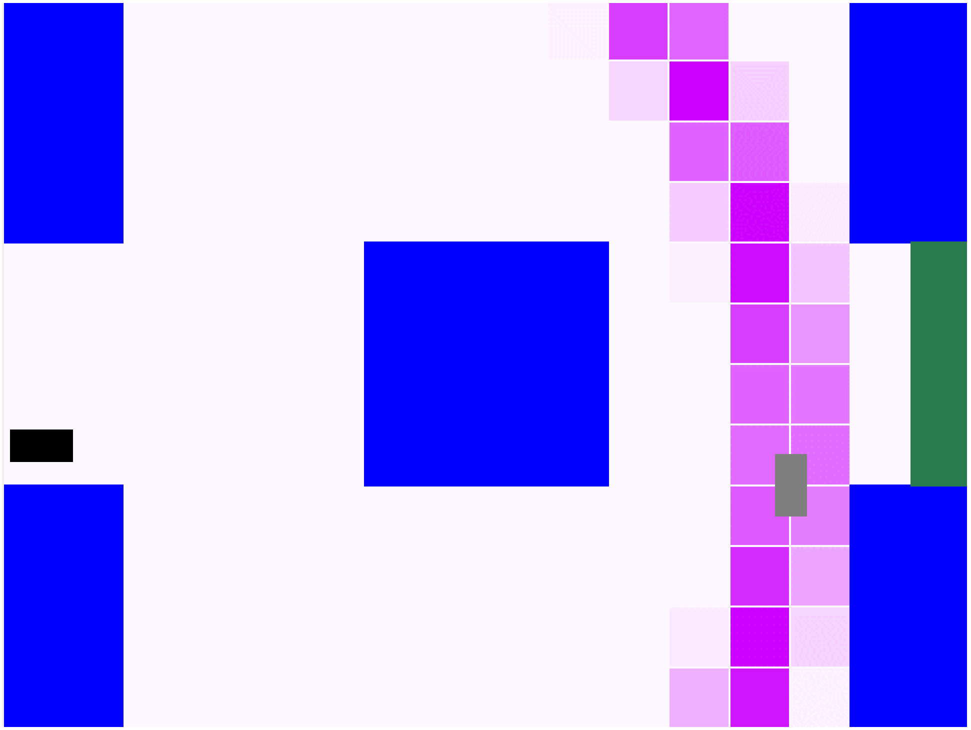
- Pseudocode

```
def observe(agentX, agentY, observedDist):
  for each tile:
      tile의 row, col을 x, y 좌표로 변환
      dist ← agent와 tile의 euclidean 거리 계산
      belief[tile] ← belief[tile] X p(observedDist|dist, std)
  belief를 normalize
```

- 각 프레임마다 observe 함수 호출

Emission Probability

- 다음 명령어를 통해 **결과 확인**
  - `python drive.py -p -d -k 1 -i exactInference` (자동 운전 X)
  - `python drive.py -a -p -d -k 1 -i exactInference`



Transition Probability

- 다른 자동차들이 transition probability $p(c_t|c_{t-1})$에 따라 이동

- Dictionary 형태의 `self.transProb` 사용
  - key = (srcTile, destTile)
  - value = transition probability

- Observation 및 transition probability를 모두 고려해 belief state tracking

- C_t의 posterior probability를 계산하는 **ExactInference 클래스의 `elapseTime` 함수 구현**

$$\mathbb{P}(C_{t+1} = c_{t+1}|D_1 = d_1, \cdots, D_t = d_t) \propto \sum_{c_t} \mathbb{P}(C_t = c_t|D_1 = d_1, \cdots, D_t = d_t) p(c_{t+1}|c_t)$$

Transition Probability

- Pseudocode

def elapseTime():

  belief_{t+1} 초기화

  for each srcTile, destTile:

    $\text{belief}_{t+1}[\text{destTile}] \pm \text{belief}_{t}[\text{srcTile}] \text{ X } p(\text{destTile}|\text{srcTile})$

  belief_{t+1}을 normalize

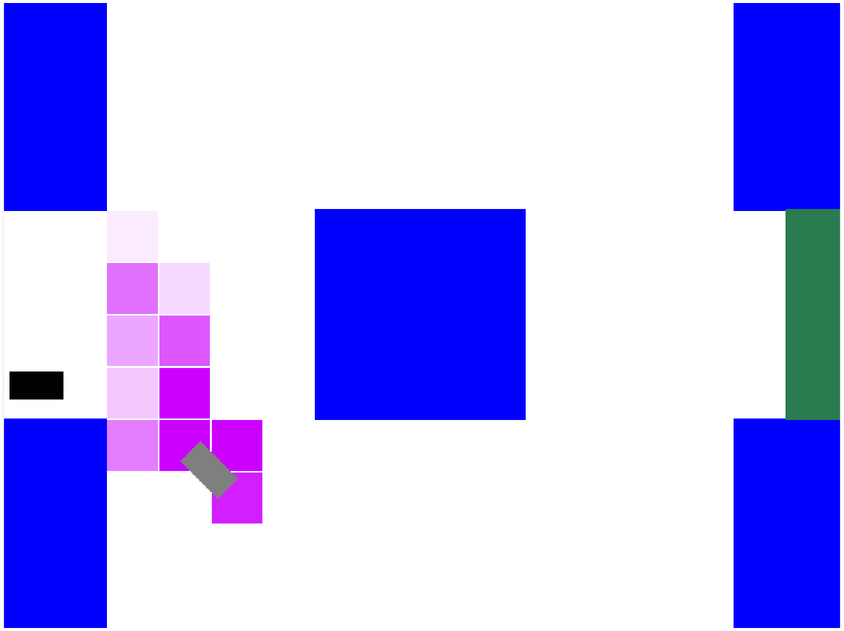
  $\text{belief}_{t} \leftarrow \text{belief}_{t+1}$

- 각 프레임마다 1. elapseTime 및 2. observe 함수 호출

Transition Probability

- 다음 명령어를 통해 **결과 확인**
  - `python drive.py -a -d -k 1 -i exactInference`
    - Emission probability 문제 대비 -p flag가 제거됨

-k 1



-k 3