

# Clustering Algorithm

구름

도시공학과 일반대학원

한양대학교

**1. 추천 시스템**

2. 강화 학습

# 연관 규칙 (Association Rule)

구매 내역을 분석하여 아이템간의 연관성을 분석하는 알고리즘으로 장바구니 분석에 활용

Item	A고객	B고객	C고객	D고객	E고객	F고객	G고객	H고객	I고객	J고객
과자	1	0	0	1	1	0	0	1	1	0
맥주	1	1	1	1	0	1	1	1	1	0
우유	1	0	0	0	0	0	0	1	0	0
커피	0	0	0	0	0	0	0	0	0	0
치즈	0	1	0	1	0	0	1	0	0	0
라면	0	0	1	0	1	1	0	1	1	0

지지도 (support)

$$S(A, B) = \frac{N(A \cap B)}{N} \quad S(A) = \frac{N(A)}{N} = P(A)$$

신뢰도(confidence)

$$C(A \rightarrow B) = \frac{S(A, B)}{S(A)}$$

향상도(lift)

$$L(A \rightarrow B) = \frac{C(A, B)}{S(B)} = \frac{S(A, B)}{S(A) \cdot S(B)}$$

$$S(\text{과자}, \text{맥주}) = 0.4$$

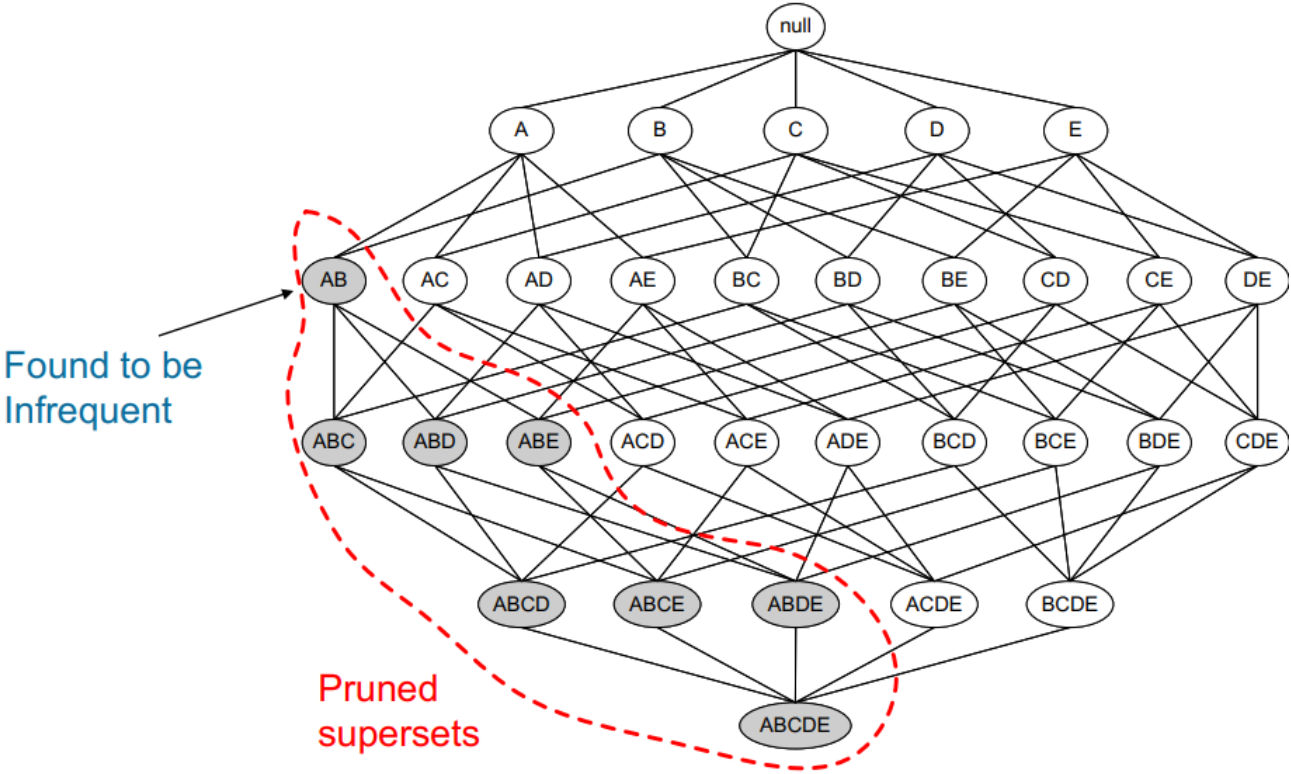
$$C(\text{과자} \rightarrow \text{맥주}) = 0.8$$

$$L(\text{과자} \rightarrow \text{맥주}) = 1$$

# A priori algorithm

연관분석의 경우 item갯수가 늘어나면 계산 복잡도가 높아져서

※ item 개수 n일 경우 조합 가능한 부분집합의 개수는  $2^n - 1$  개, **연관규칙 수는  $3^n - 2^{(n+1)} + 1$  개**  
발생 빈도가 일정 수준(임계치) 이상인 아이템만 이용하여 연관분석을 수행하는 알고리즘



# 유사도 함수를 이용한 추천 방식

상품을 벡터로 표현할 수 있다면 다차원 공간에서 각 상품의 좌표를 비교하여 유사 상품을 추천하는 방식

## 유클리디안 유사도 (Euclidean similarity)

$$\|\mathbf{p} - \mathbf{q}\| = \sqrt{(\mathbf{p} - \mathbf{q}) \cdot (\mathbf{p} - \mathbf{q})} = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p} \cdot \mathbf{q}}.$$

## 코사인 유사도 (cosine similarity)

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

## 피어슨 유사도 (pearson similarity)

$$\text{pearson\_sim}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u) \cdot (r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}}$$

# TF-IDF (Term Frequency - Inverse Document Frequency)

[https://m.blog.naver.com/mk\\_crew/222070694314](https://m.blog.naver.com/mk_crew/222070694314)

문서의 단어를 카운트하여 문서간 유사도를 추출하는 알고리즘

**TF** (Term Frequency)  $tf(d, t) =$  특정 문서  $d$  에서 특정 단어  $t$ 의 등장 횟수

**DF** (Document Frequency)  $df(t) =$  특정 단어  $t$ 가 등장한 문서의 수

**IDF** (Inverse Document Frequency)  $idf(d, t) = \log \left( \frac{n(\text{문서수})}{1 + df(t)} \right)$

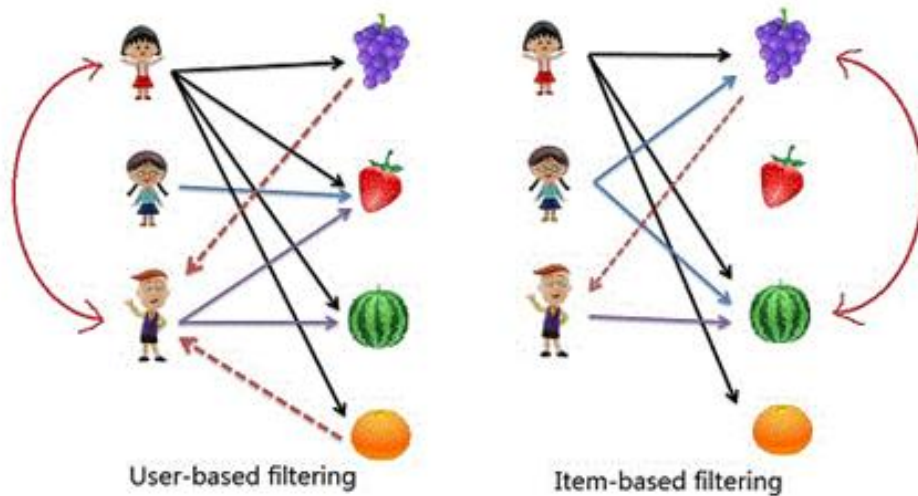
-	과일이	길고	노란	먹고	바나나	사과	싫은	저는	좋아요	단어	IDF(역 문서 빈도)
문서1	0	0	0	1	0	1	1	0	0	과일이	$\ln(4/(1+1)) = 0.693147$
문서2	0	0	0	1	1	0	1	0	0	길고	$\ln(4/(1+1)) = 0.693147$
문서3	0	1	1	0	2	0	0	0	0	노란	$\ln(4/(1+1)) = 0.693147$
문서4	1	0	0	0	0	0	0	1	1	먹고	$\ln(4/(2+1)) = 0.287682$
-	과일이	길고	노란	먹고	바나나	사과	싫은	저는		바나나	$\ln(4/(2+1)) = 0.287682$
문서1	0	0	0	0.287682	0	0.693147	0.287682	0		사과	$\ln(4/(1+1)) = 0.693147$
문서2	0	0	0	0.287682	0.287682	0	0.287682	0		싫은	$\ln(4/(2+1)) = 0.287682$
문서3	0	0.693147	0.693147	0	0.575364	0	0	0		저는	$\ln(4/(1+1)) = 0.693147$
문서4	0.693147	0	0	0	0	0	0	0.6931		좋아요	$\ln(4/(1+1)) = 0.693147$

Cos Similarity 통해  
Document 유사성 파악

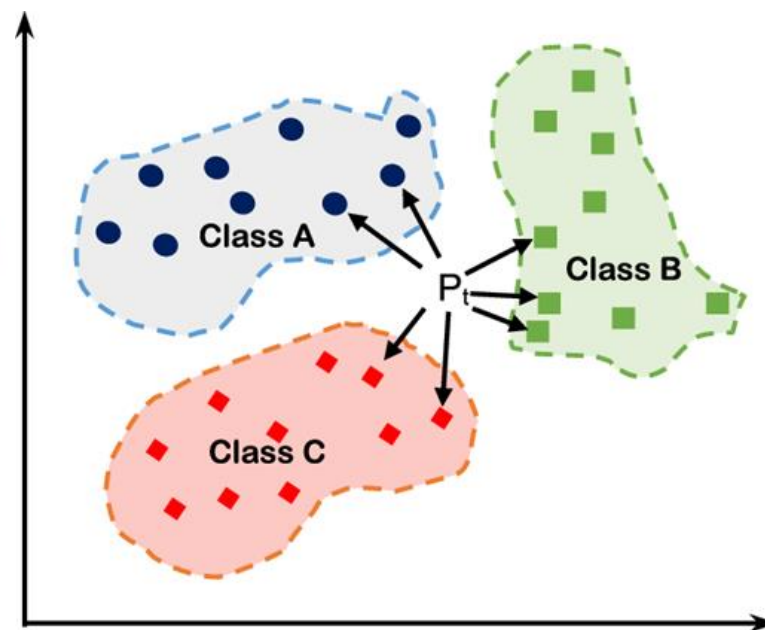
## 협업 필터링 (collaborative filtering)

<https://eda-ai-lab.tistory.com/527>

### collaborative filtering

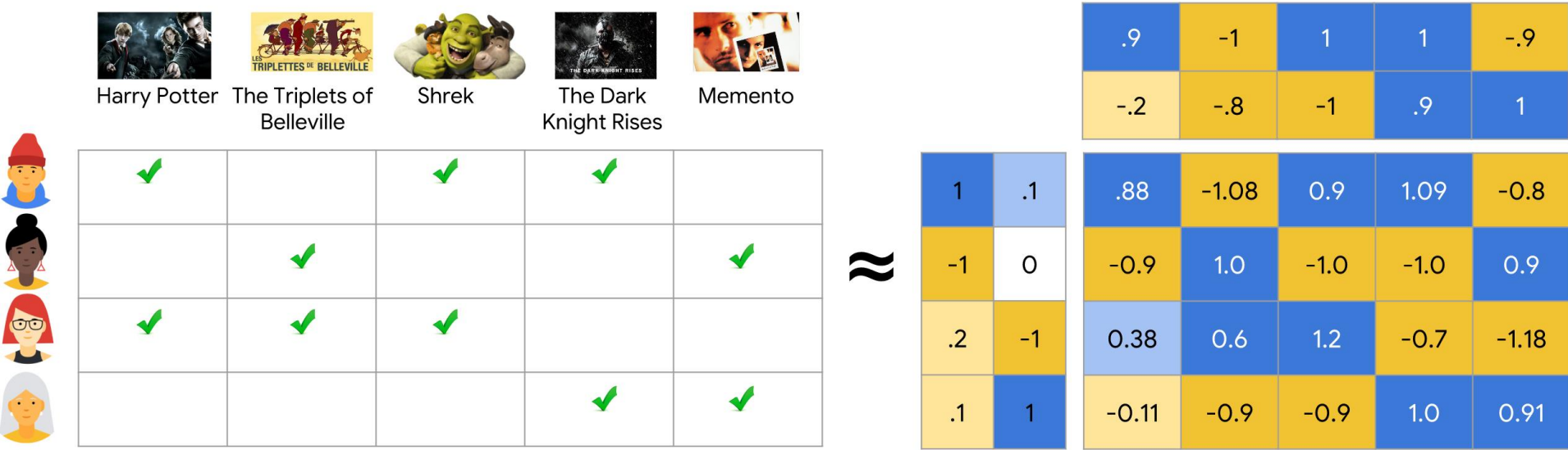


### K-Nearest Neighbor



# Matrix Factorization (SGD, ALS)

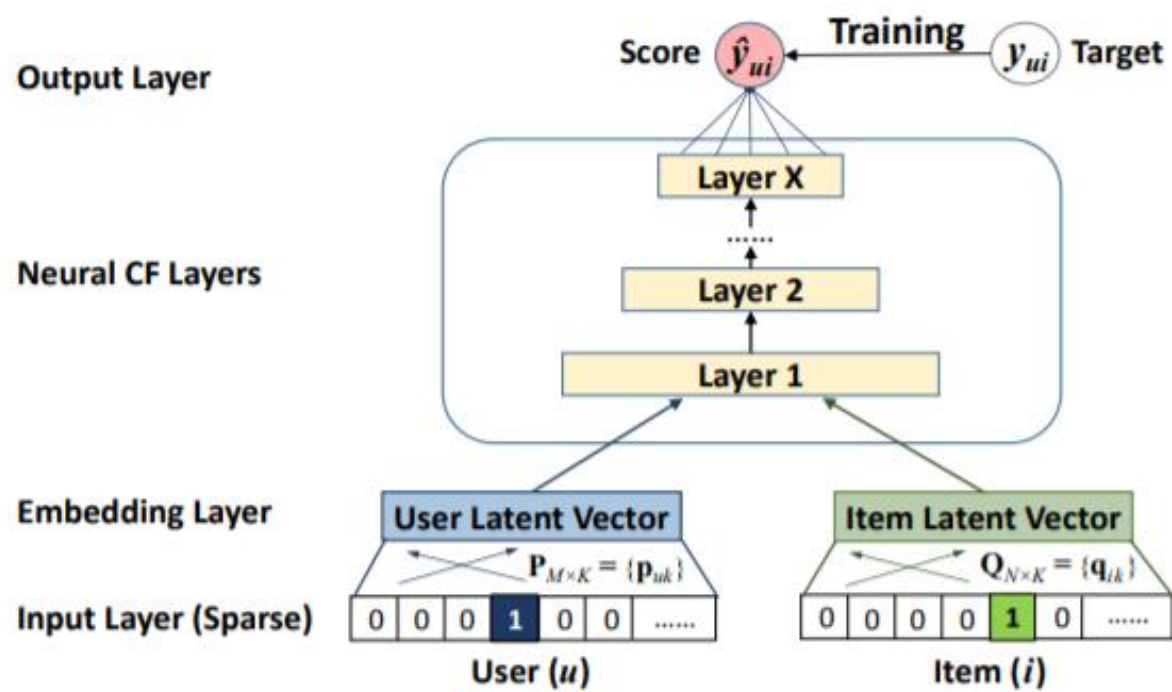
<https://developers.google.com/machine-learning/recommendation/collaborative/matrix>





# Neural Collaborative Filtering

<https://arxiv.org/pdf/1708.05031.pdf>



# 협업필터링을 활용한 행정동 – 연령/성별 분포 차이 분석

<https://jumin.mois.go.kr/#>

전체 읍면동별 연령대/성별 인구 분포 Matrix를 Neural Collaborative Filtering 로 분해하여 분석

 행정안전부

주민등록 인구통계

> 주민등록 인구 및 세대현황

> **연령별 인구현황**

> 주민등록 인구 기타현황

연령별 인구현황

통계표

그래프

행정구역

전국

시·군·구

등록구분

전체

조회기간

☒ 월간 ☐ 연간

2022년

10월

~

2022년

10월

※ 매월 말일 작성 / 공표일시 : 매월 1일 12시 이후(공표일이 주말, 공휴일인 경우에는 다음 평일에 공표)

구분

☒ 계 ☒ 남·여 구분

정렬순서

행정기관코드

오름차순

연령 구분 단위

10세

만 연령구분

0

100이상

검색

초기화

☒ 현재화면 ☐ 전체시군구현황 ☐ 전체읍면동현황

csv 파일 다운로드

xlsx 파일 다운로드

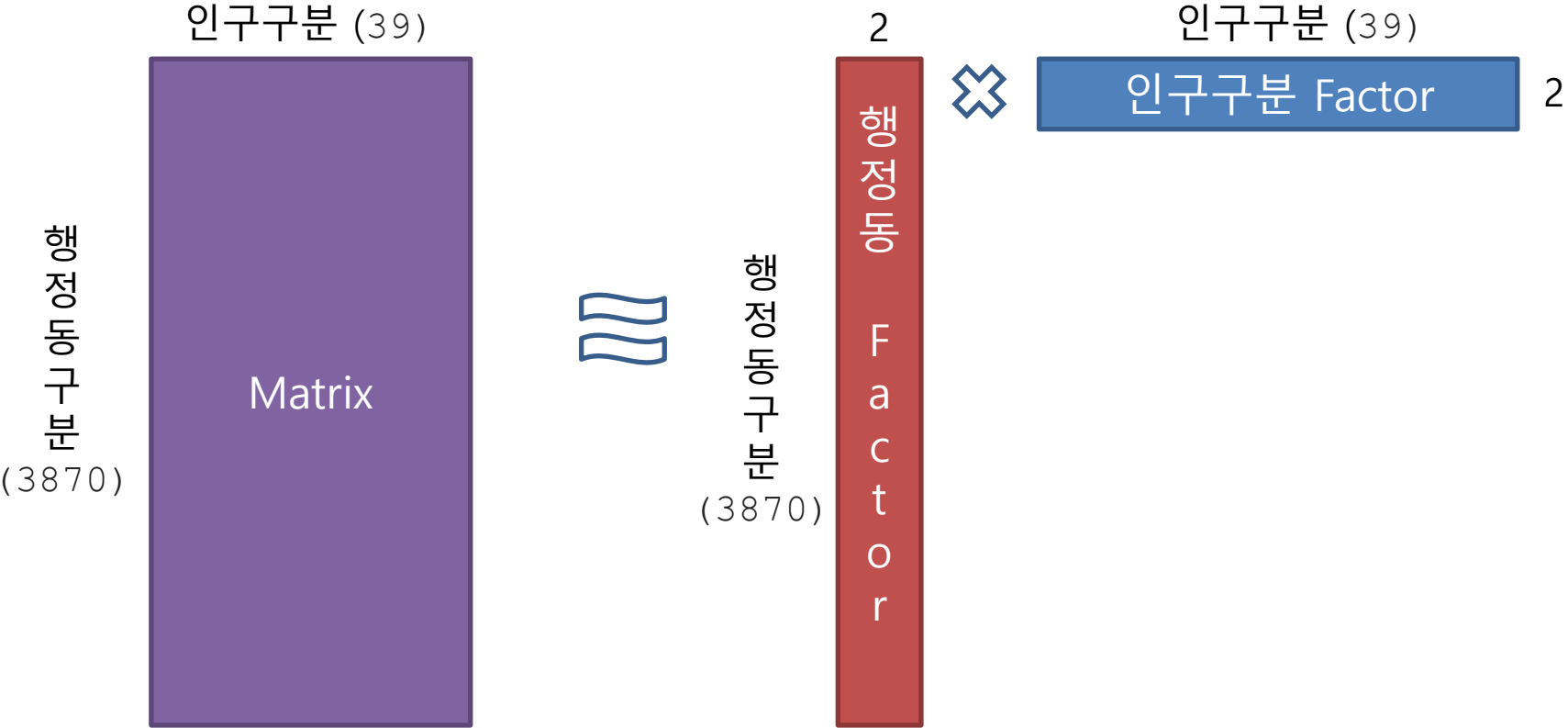
10

# 분석하고자 하는 Matrix 데이터를 확인

## 행정동 구분

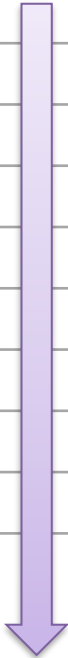
	여 _10~19 세	여 _20~29 세	여 _30~39 세	여 _40~49 세	여 _50~59 세	여 _60~69 세	여 _70~79 세	여 _80~89 세	여 _90~99 세	여 _100 세 이 상
행정구역										
서울특별시 (1100000000)										
서울특별시 종로구 (1111000000)										
서울특별시 종로구 청운효자동(1111051500)										
서울특별시 종로구 사직동(1111053000)										
서울특별시 종로구 삼청동(1111054000)										
서울특별시 종로구 부암동(1111055000)										
서울특별시 종로구 평창동(1111056000)										
서울특별시 종로구 무악동(1111057000)										
서울특별시 종로구 교남동(1111058000)										
서울특별시 종로구 가회동(1111060000)										
서울특별시 종로구 종로1.2.3.4가동(1111061500)										
서울특별시 종로구 종로5.6가동(1111063000)										
서울특별시 종로구 이화동(1111064000)										
서울특별시 종로구 혜화동(1111065000)										
서울특별시 종로구 창신제1동(1111067000)										
서울특별시 종로구 창신제2동(1111068000)										
서울특별시 종로구 창신제3동(1111069000)										
서울특별시 종로구 숭인제1동(1111070000)										
서울특별시 종로구 숭인제2동(1111071000)										
서울특별시 중구 (1114000000)										
서울특별시 중구 소공동(1114052000)										
서울특별시 중구 회현동(1114054000)										
서울특별시 중구 명동(1114055000)										
서울특별시 중구 필동(1114057000)										
서울특별시 중구 장충동(1114058000)										
서울특별시 중구 광희동(1114059000)										
서울특별시 중구 을지로동(1114060500)										

Matrix 를 2차원의 행정동 factor와 인구구분 factor로 분해 목표



Neural Network에 입력할 학습용 데이터셋을 아래와 같이 변환하여 생성

	0	1	2	3	4	5
0						
1			3059			
2						
3						
4						
5						
6						
7						
8						
9						
10						



행정동 입력 값

0	1	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

인구구분 입력 값

0	0	1	0	0	0
---	---	---	---	---	---

행렬값

3059
------

## Neural Network 구조 – 가중치가 Factor Matrix가 되도록 입력 층 구조화

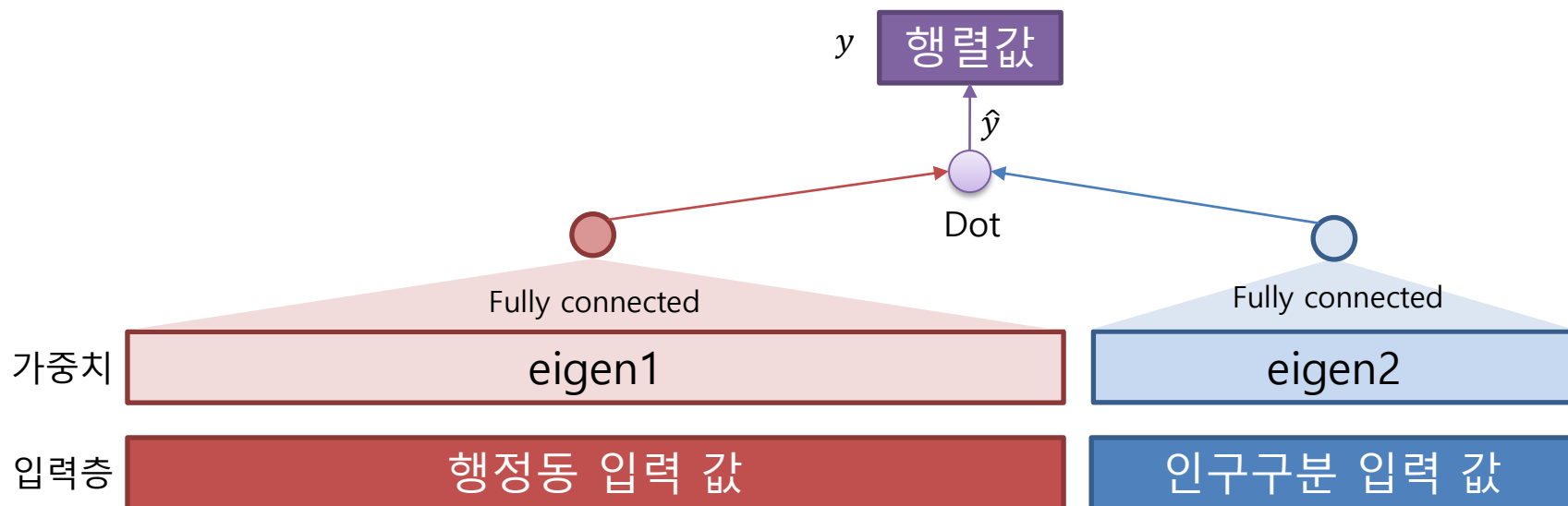
```
input1 = layers.Input(shape=(location.shape[0]))
input2 = layers.Input(shape=(pcount.shape[0]))

eigen1 = layers.Dense(2, use_bias=False, activation='linear')(input1)
eigen2 = layers.Dense(2, use_bias=False, activation='linear')(input2)

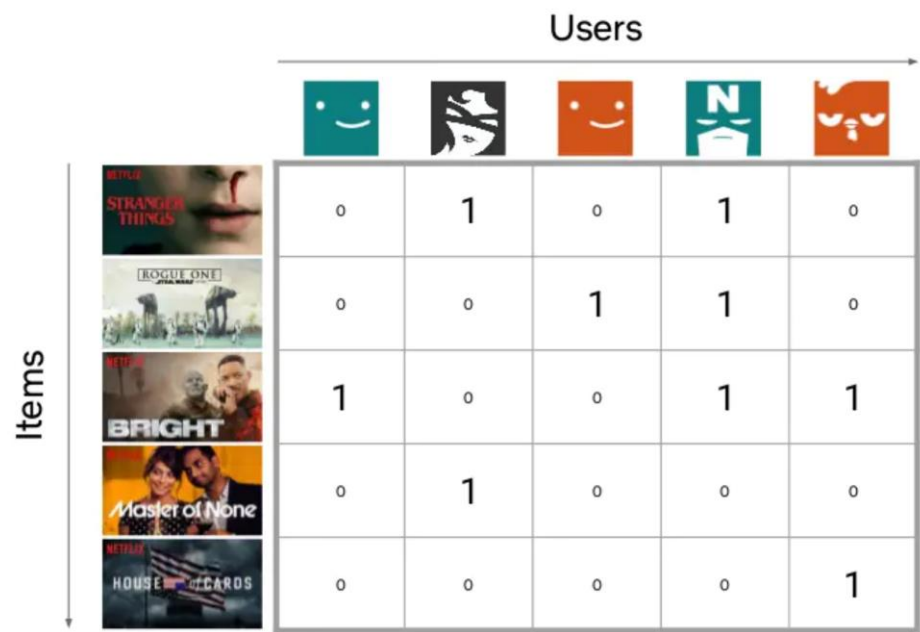
net = layers.Dot(axes=(1, 1))([eigen1, eigen2])

model = Model(inputs=[input1, input2], outputs=net)

model.compile(loss='mse', optimizer='adam', metrics=['acc'])
model.summary()
```



# Netflix 추천 시스템 (Collaborative Filtering)

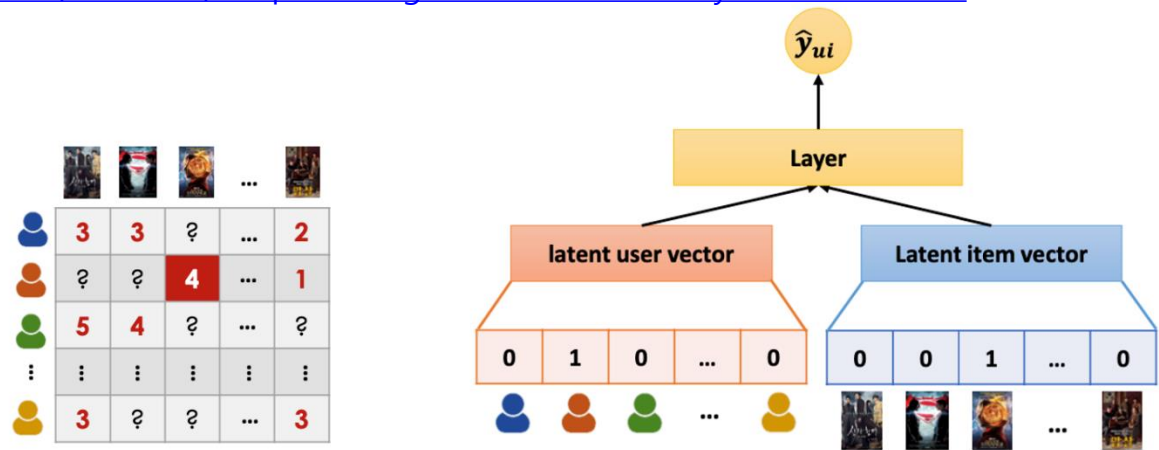


## A Matrix Factorization view

$$R \approx UV^T$$

$$\min_{u,v} \sum_{i,j \in R} (r_{ij} - u_i^T v_j) + \lambda (\|u_i\|^2 + \|v_j\|^2)$$

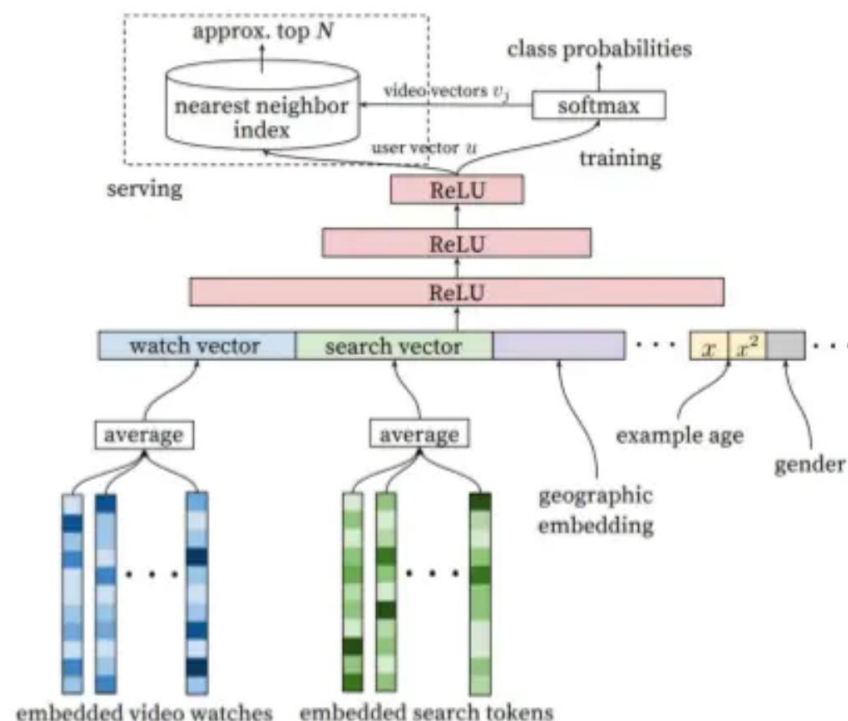
<https://www.slideshare.net/moustaki/deep-learning-for-recommender-systems-86752234>



# YouTube Recommendations

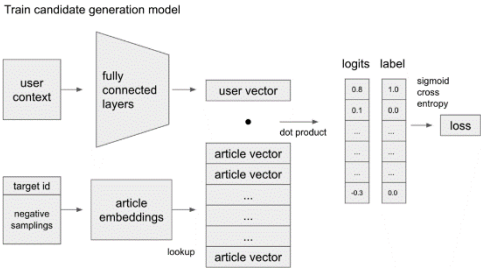
- Two stage ranker: candidate generation (shrinking set of items to rank) and ranking (classifying actual impressions)
- Two feed-forward, fully connected, networks with hundreds of features

[Covington et. al., 2016]





# 당근마켓 추천시스템



## B 사용자의 최근 본 글



생활/가공식품, 60,000 원  
노만코펜하겐 모르모르 그릇 판매 합니다.  
요리가 취미여서 한동안 서울 성동구 옥수동



디지털/가전, 26,000원  
브리츠 BZ-TWS5  
브리츠 블루투스 무선 이어폰입니다. 완전 새제품  
서울 강남구 압구정동



가구/인테리어, 60,000 원  
스피드락 철제선반  
1200\*400 선반6개짜리  
2개 1000\*400 선반5개  
서울 성동구 금호동1가



디지털/가전, 45,000원  
[미개봉] bsw 비엔나 에스프레소 머신  
bsw 비엔나 에스프레소  
서울 중구 약수동

## 서울 성동구 옥수동 지역 추천 글



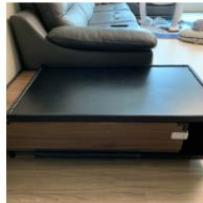
가구/인테리어, 800,000 원  
수납장판매합니다  
홍대수제목공방 가구가있  
는품경에서구입한 대형수  
서울 서초구 잠원동



가구/인테리어, 10,000 원  
원목책장팔아요  
원목책장팔아요 책상 사  
이즈 120\*200\*20 신답  
서울 동대문구 답십리제1  
동



가구/인테리어, 30,000 원  
멋진 책상 팝니다(2개)  
2개 일괄 구매시 5만 5천  
원 입니다 면 너비 70cm  
서울 성동구 하왕십리동



가구/인테리어, 50,000 원  
리바트 티테이블  
리바트에서 작년 3월에  
구입했구요 상판은 생활  
서울 성동구 금호1가동

## A 사용자의 최근 본 글



유아동/유아도서, 15,000 원  
게스 아기옷 6m-9m 3가  
지 일괄  
일괄 15000원이고 따로  
서울 중구 동화동



유아동/유아도서, 5,000 원  
아기 우주복  
저희아기인 우주복 많이 안  
입게 되어 한번정도 입고  
서울 용산구 한강로동



유아동/유아도서, 3,000 원  
돌전후 아기 올인원  
깨끗해요  
서울 용산구 한강로동



유아동/유아도서, 3,000 원  
폴로 베레모 (돌전후)  
깨끗해요  
서울 용산구 한강로동



유아동/유아도서, 콤비yt-180 보행기,점퍼  
루 팔아요  
콤피 사고서 2번사용한  
서울 성북구 돈암제2동



유아동/유아도서, 10,000 원  
킨더팜 유모차라이너 내  
놔요~  
첫째 2개월, 둘째 5개월  
서울 성동구 마장동



유아동/유아도서, 12,000 원  
아기 미끄럼틀  
쿠쿠토이즈 아기 미끄럼  
틀 팝니다. 첫 아기가 들  
서울 성동구 옥수동



유아동/유아도서, 20,000 원  
신생아아기침대 Angel  
Kid  
신생아아기침대 Angel  
서울 광진구 자양제3동

1. 추천 시스템

**2. 강화 학습**

# Neural Collaborative Filtering

<https://github.com/rlcode/reinforcement-learning-kr.git>

☰ README.md

## REINFORCEMENT LEARNING

RLCode팀이 직접 만든 강화학습 예제들을 모아놓은 Repo 입니다. [영문 \(English\)](#)

Maintainers - [이웅원](#), [이영무](#), [양혁렬](#), [이의령](#), [김건우](#)

### 목차 (Table of Contents)

Grid World - 비교적 단순한 환경인 그리드월드에서 강화학습의 기초를 쌓기

- [정책 이터레이션 \(Policy Iteration\)](#)
- [가치 이터레이션 \(Value Iteration\)](#)
- [몬테카를로 \(Monte Carlo\)](#)
- [살사 \(SARSA\)](#)
- [큐러닝 \(Q-Learning\)](#)
- [Deep SARSA](#)
- [REINFORCE](#)

CartPole - 카트폴 예제를 이용하여 여러가지 딥러닝을 강화학습에 응용한 알고리즘들을 적용해보기

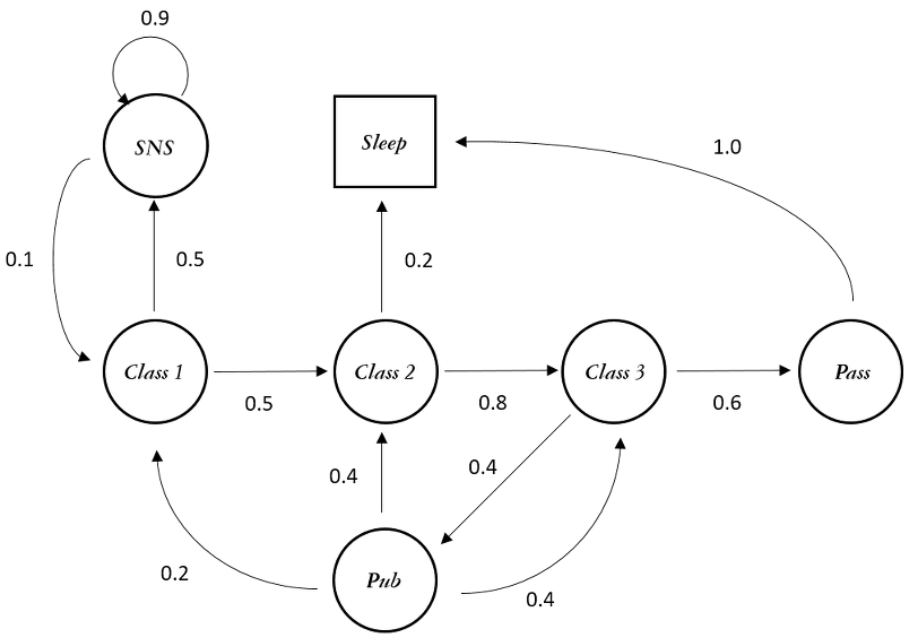
- [Deep Q Network](#)
- [Actor Critic \(A2C\)](#)

Atari [브레이크아웃](#) - 딥러닝을 응용하여 좀더 복잡한 Atari [브레이크아웃](#) 게임을 마스터하는 에이전트 만들기

- [Deep Q Network](#)

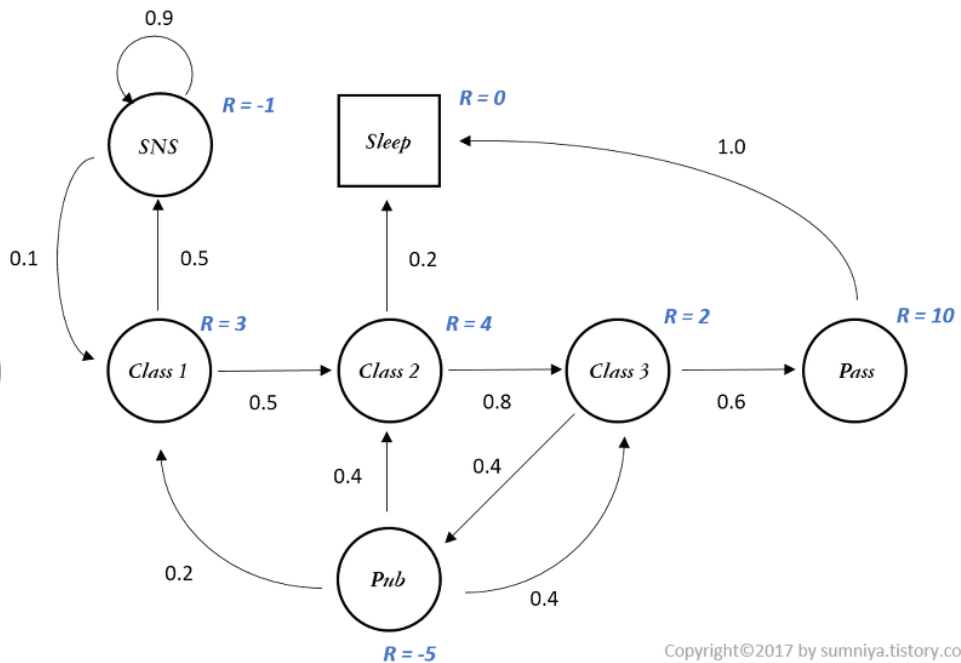
# Markov Decision Process

<https://sumniya.tistory.com/3>



Copyright©2017 by sumniya.tistory.com

Markov Process(MP, Markov Chain)



Copyright©2017 by sumniya.tistory.com

Markov Reward Process(MRP)



Markov Decision Process(MDP)

Copyright©2017 by sumniya.tistory.com

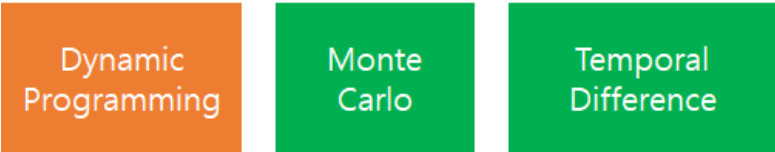
# 강화학습

<https://iridescentboy.tistory.com/m/39>

## 강화학습 문제 – MDP로 정의되는 문제

강화학습 문제를 푼다 = MDP를 푼다 =  
최적 Policy를 구한다 = BOE를 푼다

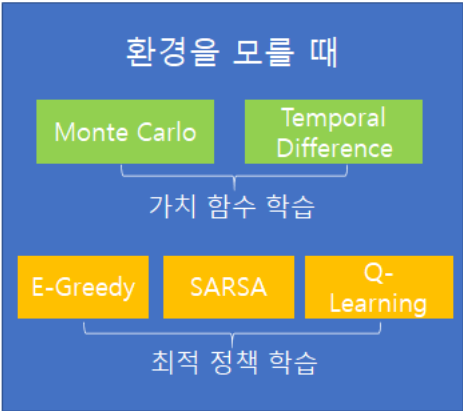
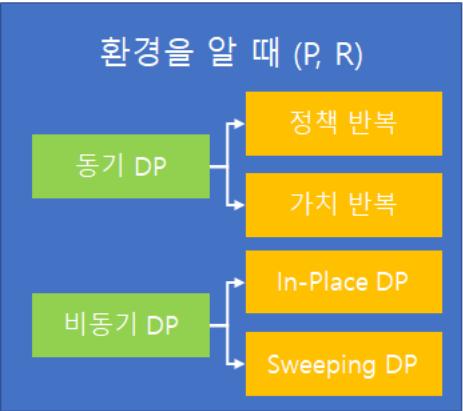
문제 풀이 기법



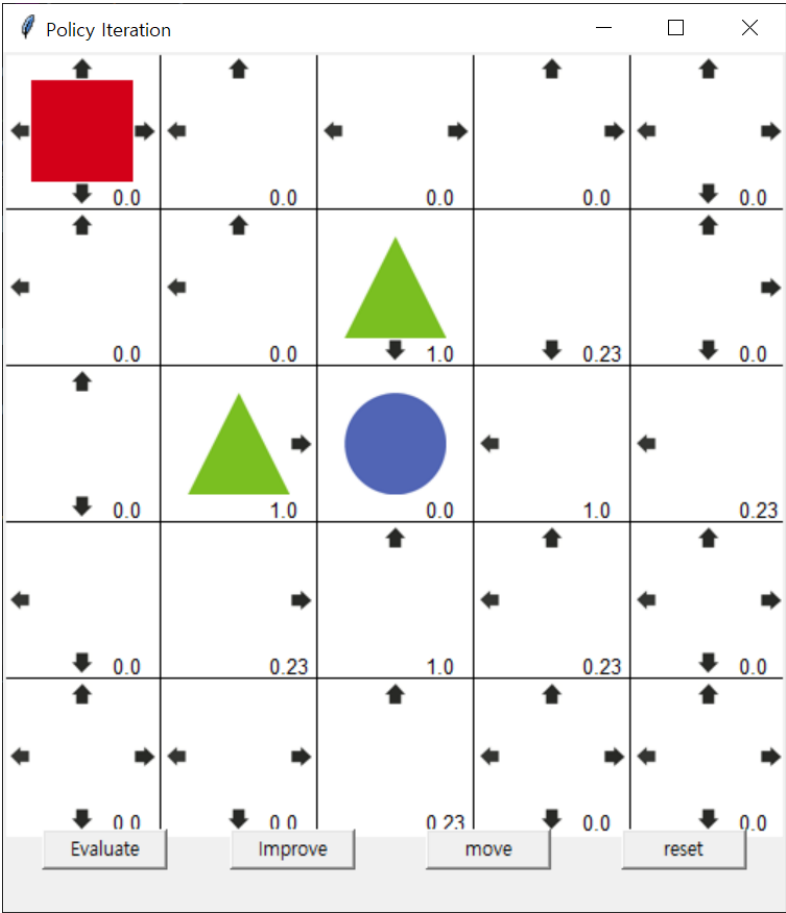
주어진 환경을 알때  
= Reward, State  
Transition을 알때

주어진 환경을 모를 때

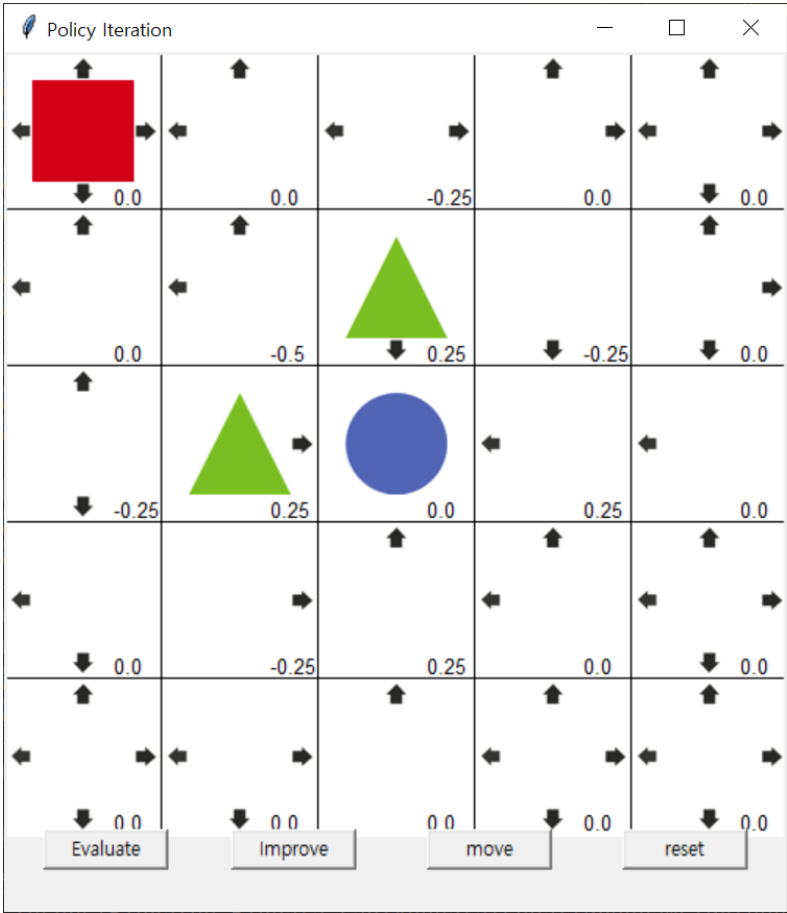
## MDP



# 정책 반복



# 가치 반복

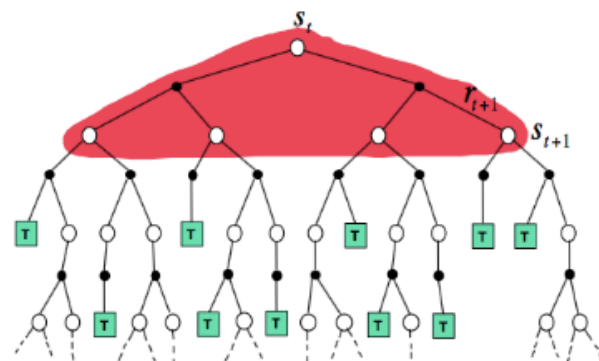


# MonteCarlo

<https://lilianweng.github.io/posts/2018-02-19-rl-overview/>

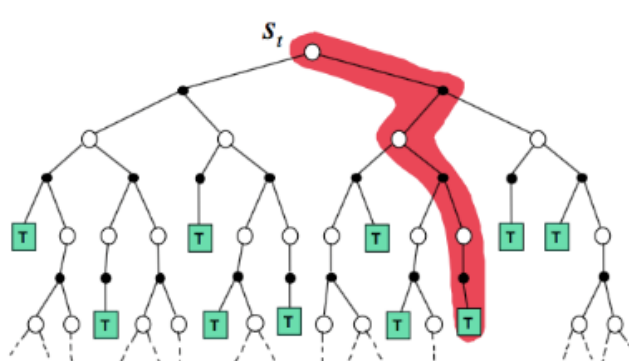
## Dynamic Programming

$$V(S_t) \leftarrow \mathbb{E}_{\pi} [R_{t+1} + \gamma V(S_{t+1})]$$



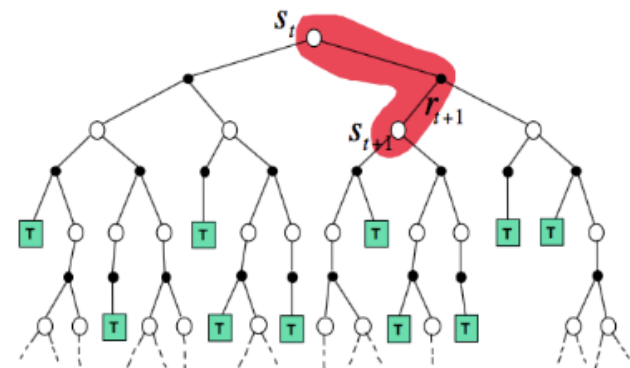
## Monte-Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha (G_t - V(S_t))$$



## Temporal-Difference

$$V(S_t) \leftarrow V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$



# SARSA vs Q-learning

<https://towardsdatascience.com/automating-pac-man-with-deep-q-learning-an-implementation-in-tensorflow-ca08e9891d9c>

## Revisiting Bellman equations

**Sarsa:**  $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha (R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$

$$q_{\pi}(s, a) = \sum_{s', r} p(s', r | s, a) \left( r + \gamma \sum_{a'} \pi(a' | s') q_{\pi}(s', a') \right)$$

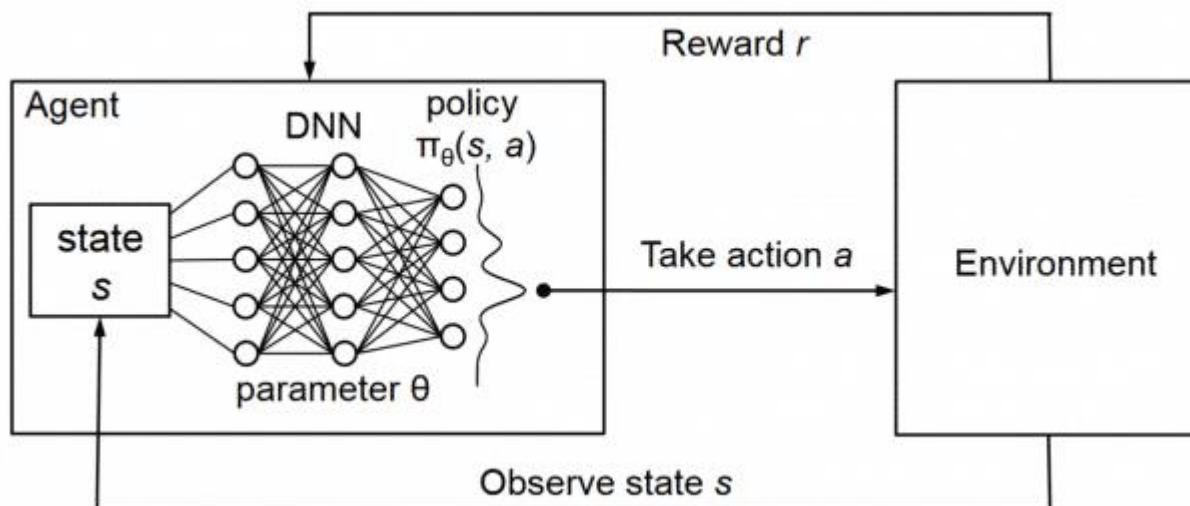
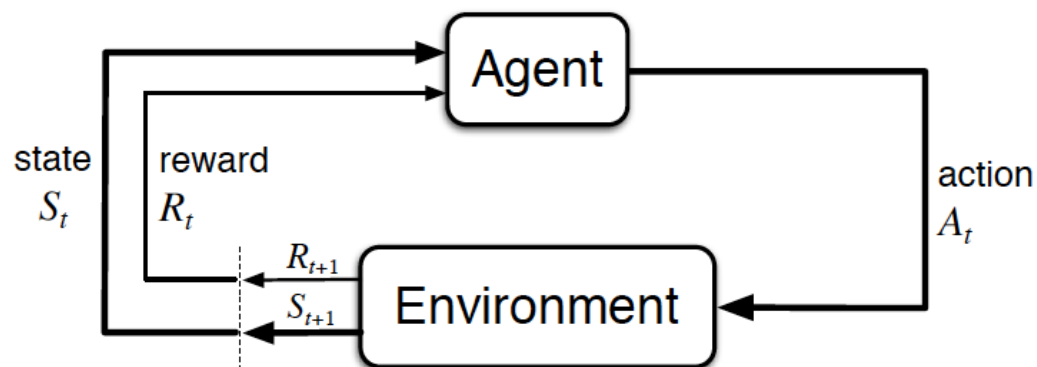
**Q-learning:**  $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha (R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t))$

$$q_*(s, a) = \sum_{s', r} p(s', r | s, a) \left( r + \gamma \max_{a'} q_{\pi}(s', a') \right)$$



# Deep Reinforcement Learning

<https://arxiv.org/pdf/1312.5602.pdf>



<http://people.csail.mit.edu/hongzi/content/publications/DeepRM-HotNets16.pdf>