

# Unsupervised Learning, Recommenders, Reinforcement Learning - Andrew Ng

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Unsupervised Learning</b>	<b>1</b>
2.1	Define . . . . .	1
2.2	Cluster algorithm . . . . .	2
2.2.1	Define . . . . .	2
2.2.2	K-means algorithm . . . . .	2
2.3	Anomaly detection algorithm . . . . .	2
2.3.1	Density estimation . . . . .	2
2.3.2	Gaussian distribution . . . . .	2
2.3.3	Algorithm . . . . .	3
2.3.4	Evaluation? . . . . .	3
2.3.5	Other . . . . .	4
<b>3</b>	<b>Recommender systems</b>	<b>4</b>
3.1	Basis . . . . .	4
3.2	Collaborative filtering algorithm . . . . .	5
3.2.1	Cost function of collaborative algorithm . . . . .	5
3.2.2	Binary labels: favs, likes and clicks. Let's take the course :D . . . . .	5
3.2.3	How to find related items ? . . . . .	5
3.2.4	Limitations . . . . .	5
3.3	Content-based filtering . . . . .	5
3.3.1	Define . . . . .	5
3.3.2	How ? . . . . .	5
3.3.3	Issue . . . . .	6
<b>4</b>	<b>Reinforcement Learning(Let's take the course :D)</b>	<b>6</b>
4.1	Notation . . . . .	6

## 1 Introduction

- Course 3 trong Machine Learning Specialization - Andrew Ng

## 2 Unsupervised Learning

### 2.1 Define

- Find some interesting structures about this data( not given y)

## 2.2 Cluster algorithm

### 2.2.1 Define

- A form of machine learning that finds patterns using unlabeled data (x)
- Applications: Grouping similar news, market segmentation, DNA analysis, Astronomical data analysis.

### 2.2.2 K-means algorithm

- Randomly initialize K cluster centroids
- Repeat: Assign points to closest cluster centroids and move cluster centroids by averaging

#### 2.2.2.1 Optimization objective

- $c^i$  : index of cluster to which example  $x^i$  is currently assigned
- $u_k$ : cluster centroid k
- $u_{(c^i)}$ : cluster centroid of cluster to which example  $x^i$  has been assigned
- 

$$J(c^1, \dots, c^m, u_1, \dots, u_k) = \frac{1}{m} \sum_{i=1}^m L_2(x^i - u_{(c^i)})$$

=> Ta update các parameters c và u sao cho hàm loss nhỏ nhất. Nghĩa là ta phải phân các điểm dữ liệu vào đúng cluster (tuning c) và tìm đúng cluster centroid (tuning u). Sao cho đến cuối cùng trung bình khoảng cách L2 từ các điểm dữ liệu đến cluster centroid của nó là bé nhất

- Unlike other algorithms of Supervised Learning, with above J, we don't optimize by gradient descent. Doing steps of the algorithms did do this:
  - Assign points to closest cluster centroids: update c
  - move cluster centroid by averaging : update u

#### 2.2.2.2 Issue with K-means

- Initially, we have to randomly initialize K cluster centroids. But this could lead to different results (different J)
- In practice, we have to run/train K-means multiple times. Ex: Run/train 100 times and pick one that has J function lowest.

## 2.3 Anomaly detection algorithm

- Using a unlabeled dataset to detect or raise a red flag if there is an unusual or an anomalous event.

### 2.3.1 Density estimation

- We detect the anomaly using probability. If a point has a low probability, it is considered an anomaly, and vice versa

### 2.3.2 Gaussian distribution

- 

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- Mục đích là dựa vào data xây dựng một Gaussian distribution => According data to find  $\sigma$  and  $\mu$  and use them to evaluate a new data point.

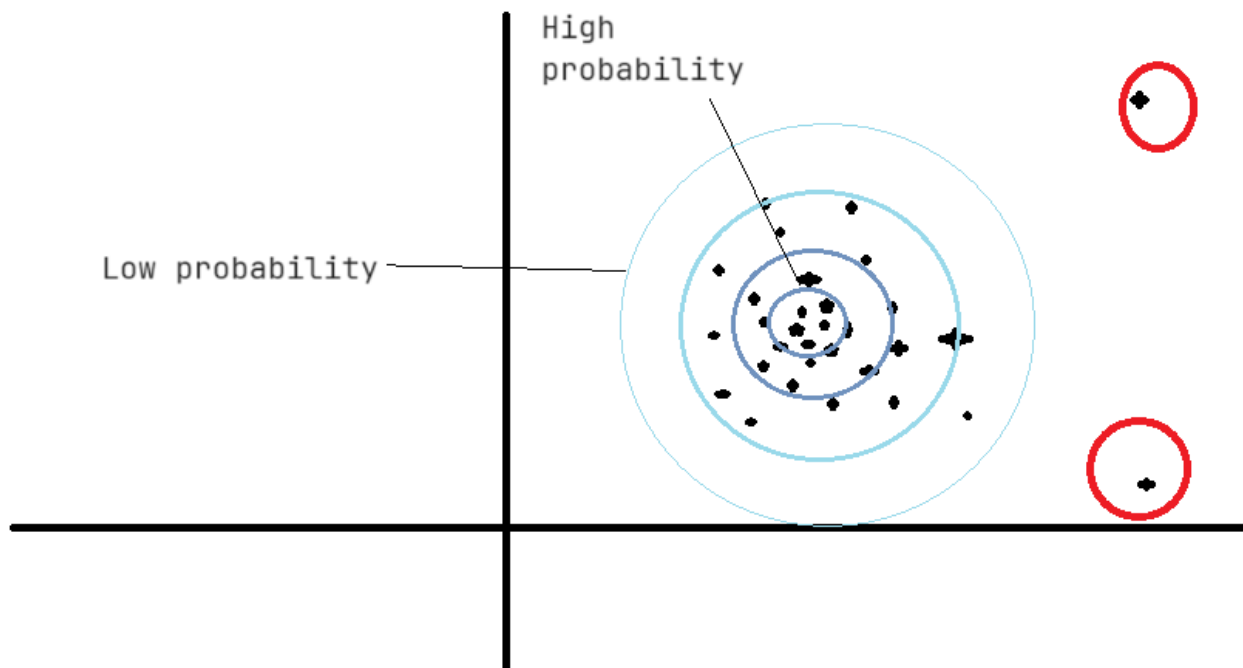


Figure 1: Estimation

•

$$\mu = \frac{1}{m} \sum_{i=1}^{i=m} x^i$$

-

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{i=m} (x^i - \mu^i)^2$$

### 2.3.3 Algorithm

- Giả thiết naive bayes: Với mỗi điểm dữ liệu có nhiều features, ta tính xác suất bằng cách tích các xác suất của mỗi features (coi các feature độc lập với nhau)
- $p(x) = p(x_1) * \dots * p(x_n)$
- Đầu tiên lựa chọn feature => Tính các tham số  $\mu_1, \dots, \mu_n, \sigma_1, \dots, \sigma_n$  (với n là số feature) => Tính  $p(x)$  và đánh giá xem nó có phải là anomaly?  $p(x) < \epsilon$ ?

### 2.3.4 Evaluation?

- In practice, we should have the dataset that is labeled to evaluate the system easily.
- Ex: If we have 10020 examples, 10000 examples are usual and 20 examples are unusual.
  - We use 8000 examples usual to train model and use 2000 examples usual and 20 examples unusual to validate (may have test dataset).
  - Remember that we don't use examples unusual to train.
- Because the dataset is unbalanced, we should use metrics like Precision, recall, F1, and use cross-validation.
- According to this evaluation, we may tune  $\epsilon$  and select features properly.

### 2.3.5 Other

- Vì chúng ta dựa vào Gaussian distribution để đánh giá xác suất của một điểm dữ liệu, do đó thật tốt nếu các feature đều có phân phối Gaussian.
- Nếu không hãy sử dụng các hàm chuyển đổi về Gaussian distribution như: log, căn bậc 2, bậc 3.

## 3 Recommender systems

### 3.1 Basis

	User 1	User 2	User 3	User 4	User 5	User 6	Action
Mov 1	5	1	?	1	4	2	0.9
Mov 2	1	5	1	?	2	4	0.1
Mov 3	4	3	4	2	?	?	0.85
Mov 4	?	1	?	2	5	1	0.95
Mov 5	?	?	5	1	4	2	0.7

Figure 2: Based on features

- Trên các một cách thức thực hiện recommend cơ bản. Với user j ta tìm  $w^j * x^j + b^j$ , dựa vào ratings của user j ta tìm  $w^j, b^j$ . x là feature
- Notation:
  - $r(i,j)$  là một ma trận có số hàng là số movie, số cột là số feature. 1 if user j rated movie i and otherwise
  - $y(i,j)$  là rating given by user j on movie i
  - $w^j, b^j$  là parameters for user j
  - $x^i$  là feature vector of movie i
  - $m^j$  is the number of movies rated
  - Cost function with user j:

$$J_j(w, b) = \frac{1}{m^j} \sum_{i:r(i,j)=1} (w^j * x^i + b^j - y(i, j))^2 + regu(w)$$

- Cost function:

$$J(w, b) = \frac{1}{2} \sum_{j=1}^{j=nu} \sum_{i:r(i,j)=1} (w^j * x^i + b^j - y(i, j))^2 + \sum_{j=1}^{j=nu} regu(w)$$

### 3.2 Collaborative filtering algorithm

- Actually, we don't have value of feature x1, x2,... Hence, we need to learn it. (learn x)

#### 3.2.1 Cost function of collaborative algorithm

$$J(w, b, x) = \frac{1}{2} \sum_{i,j:r(i,j)=1} (w^j * x^i + b^j - y(i, j))^2 + \sum_{i=1}^{i=n_m} regu(x) + \sum_{j=1}^{j=n_u} regu(w)$$

#### 3.2.2 Binary labels: favs, likes and clicks. Let's take the course :D

#### 3.2.3 How to find related items ?

- Find item k similar to item x:

$$\sum_{l=1}^n (x^k - x^i)^2$$

#### 3.2.4 Limitations

- With new items that few users have rated
- With new users that just have rated a few items

### 3.3 Content-based filtering

#### 3.3.1 Define

- A technique recommend items to you based on features of user and item to find good match
- Ex:  $x_u$  have age,gender,country,...  $x_m$  have year,genre,reviews,average rating,...

#### 3.3.2 How ?

- To predict the rating of user j on the movie i, we implement:

$$w^j * x^i$$

- We consider  $v_u^j : w^j, v_m^i : x^i$ . And  $v_u^j$  is computed from  $x_u^j$ ,  $v_m^i$  is computed from  $x_m^i$ .
- Note:  $x_u$  can be different  $x_m$  in shape but  $v_u$  have to be similar  $v_m$  in shape.
- How to compute v ?
  - Cost function for NN:

$$J = \sum_{i,j:r(i,j)=1} (v_u^j * v_m^i - y(i, j))^2 + NNreguterm$$

- Note: We don't need to train NN separately. Because NN can combine.
- Find related items: Use  $L2(v_k - v_i)$

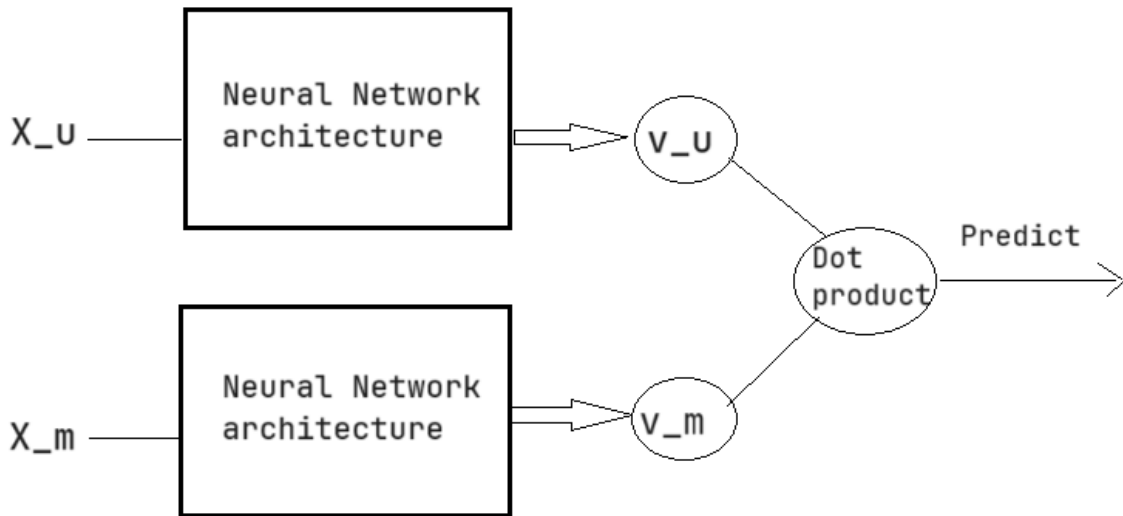


Figure 3: NN architecture

### 3.3.3 Issue

- Ex: We have 10 million ads to recommend. When a new user accesses the website we want to recommend ads for them. For each new user, we have to fit  $x_u$  of them with 10 million, so inefficient. Remember we have pre-computed  $x_m$  but we don't know  $x_u$ .
- Hence, we should choose some from the large catalog. It is retrieval
  - ex: top 50 ads in the their country. For each genre, we choose top 20,....

## 4 Reinforcement Learning(Let's take the course :D)

### 4.1 Notation

- Tại mỗi state  $s$  bất kì ta có thể có nhiều lựa chọn hành động. Mục tiêu là làm sao để chọn được hành động mang lại reward cao nhất. Ý tưởng: figure out tất giá trị reward khi chọn các hành động tại state  $s$ . Sau đó chọn hành động mang lại reward cao nhất
- $Q(s,a)$ : Là giá trị reward mà tại state  $s$  chúng ta chọn hành động  $a$
- Giá trị của  $Q(s,a)$  không chỉ là phần thưởng lập tức nhận được khi chọn hành động  $a$  mà còn là tương lai.
- Bellman Equation:
 
$$Q(s, a) = R(s, a) + \gamma \max_{a'} Q(s', a')$$
- Công thức này có nghĩa là tại state  $s$  ta chọn hành động  $a$  và cộng thêm giá trị tối ưu ở trạng thái  $s'$  nhận được ( $s'$  nhận được khi  $s$  take a action)
- Ta sử dụng NN architecture để figure out value of  $Q(s,a)$

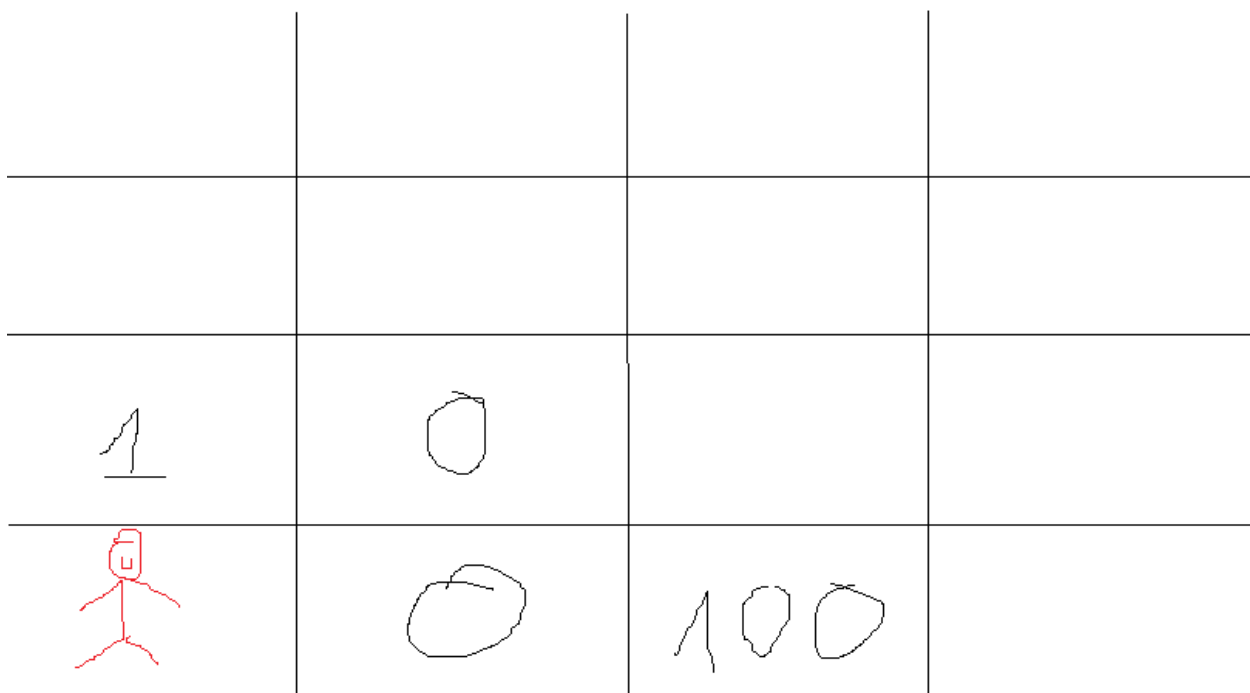
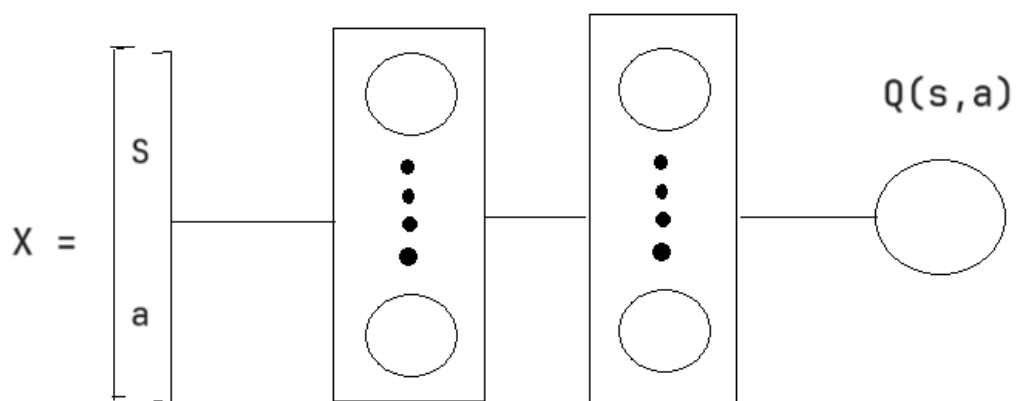


Figure 4: Ex



0

Figure 5: NN architecture

- Tóm lại là xây dựng một mạng neural network với input là một vector  $(s,a)$  ( $s,a$  có thể gồm nhiều phần tử) để tính toán ra  $Q(s,a)$ .
- Vậy lấy dataset như thế nào? interact environment và áp dụng bellman equation
  - Khởi tạo NN. Do đó  $Q(s,a)$  ban đầu là ngẫu nhiên
  - Interact với môi trường để lấy  $(s,a,R(s,a),s')$  tuple
  - Sử dụng nó để làm dataset. Với  $x = (s,a)$ ,  $y = R(s,a) + \gamma \max_{a'} Q(s',a')$