机器学习与人工智能. 2021 Fall. Lecture 1. overview.

1. Artificial Intelligence. (wiki).

Intelligence demostrated by machine.

vs. natural intelligence displayed by humans or animals.

Goals. ①. perception 感知. use inputs from sensors
(e.g., speech recognition. facial recognition) 计算视觉.

②. Reasoning deduction /inductive early researchers. systematic search 逻辑推理 (let machine think through some complex puzzles).

③. Control / motion / manipulation robots. (interact w/ physical world)
how to move efficiently. (触觉. 视觉. 处理 (抓起. 拾取)).

④. planning. 智能规划. 导航. 自动驾驶. 车间调度

⑤. learning. computer algorithms that improve automatically through experience
static. dynamic. overtime.

⑥. Natural language process (communication)
allow machines to read and understand human language.

2. Data type. 统计学意义.

Numerical. 具有实际测量意义. (身高. 体重. 血压) quantitative data.

- 离散: 可计数 (有限. / 无限) → 掷硬币直到100次人头朝上的次数. [100, ∞)
- 连续: 不可数. 只能用区间表示.

Categorical: 被描述对象的性质. (可用 numerical data 描述. e.g., 1 = F. 0 = M). 无数学意义
qualitative data

- nominal: 无排序的标签
- ordinal: 有排序的

Boolean — Binary classification 二元分类.
categorical — multiclass →
ordinal — ordinal →
(train multi binary classification)
$P(y=1) = 1 - Pr(Target > 1)$
$P(y=2) = Pr(T>1) - P(T>2)$
movie rating

real — regression
ordering — ranking.
...

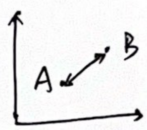
3. K-Nearest Neighbor

(1). Example. loan application.

userID.	$x_1^{(i)}, x_1$ age.	$x_2^{(i)}$ loan amount.	$x_3^{(i)}$ income.	$x_4^{(i)}$ purchase Freq. (y)	y default.
1	25 $x_1^{(1)}$	40k $x_2^{(1)}$	35K	39.	N
2.	35. $x_1^{(2)}$	60k $x_2^{(2)}$	50K	18.	N
3.	23	95k $x_2^{(3)}$	200k	100.	Y.
4.	40.	62K.	170K.	28.	Y.
5.	45	80k	40K	12	N.

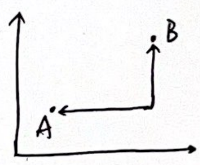
(2). Distance.

(a). Type. (i) Euclidean distance. (欧式)



features ↑. prefer (ii).

(ii) Manhattan distance



(iii). Hamming distance. (2 binary strings)

- XOR $a \oplus b$ (异或). $= (\neg a \wedge b) \vee (a \wedge \neg b)$
- count #1.

$d(11011001, 10011101) = 2$

$\oplus \quad = 01000100.$

(iv). cosine distance

mainly used in collaborative Filtering based recommendation systems.

$$D(x, y) = \cos(\theta) = x \cdot y / \|x\| \|y\|$$

(b). special case: categorical data

(i) transform to use Hamming.

(ii) Cosine.

(iii) Python categorical-similarity-measures. library

(c). Normalization

鞋码+身高 ⇒ Gender

A [179, 42, M]. B [178, 43, M]. C [165, 36, F]. D [177, 42, M]. E [160, 35, F]

Test (167, 43)

$AF = \sqrt{145}$ $BF = \sqrt{121}$ $CF = \sqrt{53}$. $DF = \sqrt{101}$ $EF = \sqrt{103}$

$K=3 \Rightarrow$ C.D.E $\Rightarrow$ Female. Not realistic

$$M_i = \max_{j \in train} x_{j,i} - \min x_{j,i}$$

$$d(x_j, x_k) = \sqrt{\sum_i \left(\frac{x_{j,i}}{M_i} - \frac{x_{ki}}{M_i}\right)^2}$$

sklearn.preprocessing pkg.

(a) standard scaler. $z = (x-\mu)/s$. → 并不会变成 normal dist. 取决于原始分布

(b). min max scale $z_i' = (z_i - z_i^{min}) / (z_i^{max} - z_i^{min})$ (normalize)

or mean

和极值关系大.

归一化 → [0,1].or [-1,1].

标准化 → mean=0. std.dev = 1 ~~normal dist~~.

对数据范围有严格要求 — 归一化 (image processing. a typical NN requires 0-1 scale)

归一化

image 0-255 RGB

数据不稳定. 存在极端值 — 标准化.

distance → 标准化

PCA

~~When in doubt~~

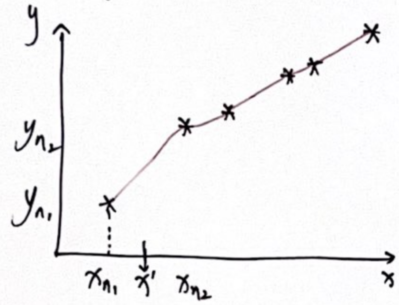
When in doubt, just standardize the data, it shouldn't hurt

机器学习与人工智能. Lecture 2. Regression. 2021F

1. Regression.

Example: stock price prediction (if predict price ↑ or ↓ ⇒ classification).

2. KNN. Regression



continuous the dots (k=2)

K=1. step function. 分段函数.

3. Definition

$x \to$ input. features. independent variables 自变量

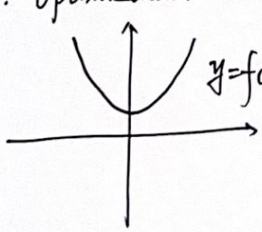
$y \to$ output. value. dependent variables. 因变量

4. Key ideas of linear regression

1, find a linear function (parameters $\vec{w}$. b)

2, minimize residuals (sum of square) for a training dataset

5. Optimization.



$y=f(x)=x^2+1$

$v^* = \min_x f(x) = 1$

$x^* = \arg\min_x f(x) = 0$

6. Data $D = (x^{(i)}, y^{(i)})_{i=1}^{N}$

Assume $x^{(i)} \sim p^*()$, $y^{(i)} = h^*(x^{(i)})$ } unknown.

$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - h_\theta(x^{(i)}))^2$

$\mathcal{H} = \{h_\theta : h_\theta(x) = \theta^T x,\ \theta \in \mathbb{R}^M\}$.

closed-form solution. 数值解.

① Solve $\nabla J(\vec{\theta})=0$ for $\vec{\theta}$

MSE: 均方误差

② Test min/max using second derivative.

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix} \qquad X = \begin{bmatrix} x_1^{(1)} & \cdots & x_M^{(1)} \\ & \vdots & \\ x_1^{(N)} & \cdots & x_M^{(N)} \end{bmatrix} \begin{matrix} \leftarrow \vec{x}^{(1)} \\ \\ \leftarrow \vec{x}^{(N)} \end{matrix}$$

① $J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)}\right)^2 \leftarrow$ ordinary Least square (OLS) 最小二乘法

$= \frac{1}{N}\cdot\frac{1}{2}(X\vec{\theta}-\vec{y})^T(X\vec{\theta}-\vec{y})$.

② Gradient $\nabla J(\theta) = X^TX\vec{\theta} - X^T\vec{y} = 0$.

③ $\vec{\theta}^{MLE} = (X^TX)^{-1}(X^T\vec{y}) = \underset{\theta}{\operatorname{argmin}}\, J(\vec{\theta})$

↳ 满秩或正定时可逆.（特征数 > 样本数, 即列数多于行数, 不满秩. 常引入 Regularization）

computational complexity

$$(\underbrace{\underset{M\times N}{X^T}\ \underset{N\times M}{X}}_{M\times M})^{-1}(\underbrace{X^T\ \underset{N\times 1}{Y}}_{M\times 1})$$

X^TX. $O(M^2N)$

$(\ )^{-1}$ $O(M^{2.373}) \sim M^3$.

X^TY $O(MN)$

$(\ )^{-1}(\ )$ $O(M^2)$

$O(M^2N + M^{2.373})$.

Linear in # of examples N

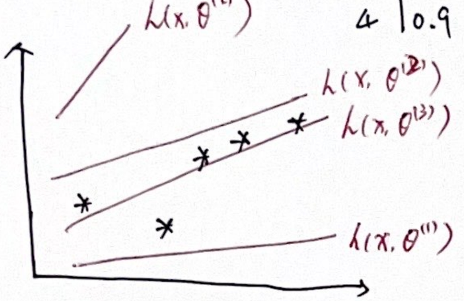
Polynomial in # of features M.

7. Gradient Descent.

(1). Random Guessing.

t	θ_1	θ_2	$J(\theta_1, \theta_2)$
1	0.2	0.2	10.4
2	0.3	0.7	7.2
3	0.6	0.4	1.0
4	0.9	0.7	19.2

when # θ ↑, you cannot directly look at/plot contour plots.



(2). unconstrained optimization.

Given function $J(\vec{\theta})$ $\quad J: \mathbb{R}^m \rightarrow \mathbb{R}$.

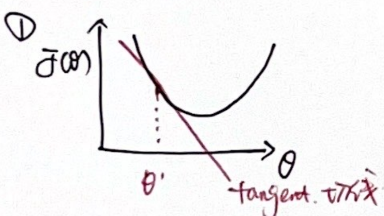
Goal. $\hat{\theta} = \underset{\theta}{\operatorname{argmin}} J(\vec{\theta})$.

J ↳ objective function

θ ↳ parameters.

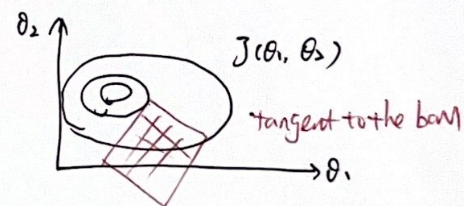
Derivative.

①



slope $= \frac{\partial J(\theta)}{\partial \theta}$ partial derivative

② Tangent plane



$\langle \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_1}, \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_2} \rangle$.

vector

$\boxed{\text{Gradient}}$

$$\nabla J(\theta) = \begin{bmatrix} \frac{\partial J(\vec{\theta})}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\vec{\theta})}{\partial \theta_m} \end{bmatrix}$$

first order partial derivatives

(3) Algorithm

① choose an initial point $\vec{\theta}$

② Repeat. a, compute gradient $\vec{g} = \nabla J(\vec{\theta})$

b, choose a step size $\gamma > 0$ (a real value)

c, update $\vec{\theta} \longleftarrow \vec{\theta} - \gamma \vec{g}$ (taking steps on the contour plots)

③ return $\vec{\theta}$ when stopping criterion is met.

Remarks.

① starting points. a, $\vec{\theta} = 0$ b $\vec{\theta}$ randomly

收敛. ②. stopping. $\|\nabla J(\vec{\theta})\|_2 < \varepsilon$ $\varepsilon = 10^{-8}$ $\|x\|_2 = \sqrt{\sum_{k=1}^{n} |x_k|^2}$

L_2 norm. 2-范数

③ step size. a, fixed value $\gamma = 0.1$

b, exact line search 线搜索

c, backtracking line search 求得一个包含理想的步长的区间. 2方法

d, schedule $\gamma_t = \gamma_0 / (t-1)\gamma_0 + 1$ (gradually shrink the step)

(4) Gradient for Linear Regression.

MSE $J(\theta) = \frac{1}{N}\sum_{i=1}^{N} J^{(i)}(\theta)$ where $J^{(i)}(\theta) = \frac{1}{2}(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})^2$

$\hookrightarrow$ doesn't affect argmin. just for mathematic convenience

$$\frac{\partial J^{(i)}(\theta)}{\partial \theta_j} = \cancel{\tfrac{1}{2}} 2\,(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})\frac{\partial}{\partial \theta_j}(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})$$

$$= (y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})\frac{\partial}{\partial \theta_j}(y^{(i)} - \sum_{m=1}^{M}\theta_m x_m^{(i)})$$

$$= -(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})\,x_j^{(i)}$$

$$\nabla J^{(i)}(\theta) = \begin{bmatrix} \frac{\partial J^{(i)}(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J^{(i)}(\theta)}{\partial \theta_M} \end{bmatrix} = \underbrace{-(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})}_{\text{scalar}}\ \vec{x}^{(i)} \quad \text{vector}$$

$$\nabla J(\theta) = \nabla\left(\frac{1}{N}\sum_{i=1}^{N} J^{(i)}(\theta)\right) = \frac{1}{N}\sum_{i=1}^{N}\nabla J^{(i)}(\theta) = \frac{1}{N}\sum_{i=1}^{N} -(y^{(i)} - \vec{\theta}^T\vec{x}^{(i)})\,\vec{x}^{(i)}$$

↑(error)· you would put a larger weight on the specific data point

(5) why SGD works.

Expectation of Gradients.

$I \sim \text{Uniform}(\{1, 2, \cdots, N\})$

$$E_I[\nabla J_I(\vec{\theta})] = \sum_{i=1}^{N} P(I=i) \nabla J_i(\vec{\theta})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \nabla J_i(\vec{\theta}) = \nabla J(\vec{\theta}).$$

Mini-batch
批梯度下降

8. MLE.

$$\theta_{MLE} = \underset{\theta}{\operatorname{argmax}}\, P(D|\theta)$$

$\downarrow$ log is monotonic

$$= \underset{\theta}{\operatorname{argmax}}\, \log P(D|\theta)$$

$$= \underset{\theta}{\operatorname{argmax}}\, l(\theta) \quad \text{where } l(\theta) \triangleq \log P(D|\theta)$$

log-likelihood.

9. Logistic Regression.

Odds ratio 优势比 从发生比看，一个特定事件与另一个发生的可能性大小。

(1) Binary

$$\sigma(u) = \frac{1}{1+\exp(-u)}$$

$$p(y|\vec{x}) = \begin{cases} \sigma(\theta^T x) & \text{if } y=1 \\ 1-\sigma(\theta^T x) & \text{if } y=0 \end{cases}$$

$y \sim \text{Bernoulli}(\phi)$

objective

$$l(\vec{\theta}) = \sum_{i=1}^{N} \log p(y^{(i)} | x^{(i)}, \vec{\theta}) \longleftarrow P_\theta(y^{(i)} | x^{(i)}).$$

$$J(\theta) = -\frac{1}{N} l(\theta) = \frac{1}{N} \sum_{i=1}^{N} \underbrace{-\log p(y^{(i)} | x^{(i)}, \theta)}_{J^{(i)}(\theta) \text{ for SGD}}.$$

(2) Multinomial $y \in \{P, S, W\}$

$$P(y=p|\vec{x}) = \exp(\vec{\theta}_p \cdot \vec{x}) / Z(\vec{x}, \theta) \qquad Z(\vec{x}, \theta) = \exp(\theta_p \cdot x) + \exp(\theta_S \cdot x) + \exp(\theta_W \cdot x)$$

Derivatives (Logistic Regression)

$$\frac{\partial J^{(i)}(\theta)}{\partial \theta_m} = \frac{\partial}{\partial \theta_m}\left(-\log p(y^{(i)} \mid x^{(i)}, \theta\right)$$

$$= \begin{cases} \partial/\partial\theta_m \; -\log\left[\sigma(\theta^T x^{(i)})\right] & \text{if } y^{(i)}=1 \\ \partial/\partial\theta_m \; -\log\left[1-\sigma(\theta^T x^{(i)})\right] & \text{if } y^{(i)}=0 \end{cases}$$

$$= \cdots$$

$$= -\underbrace{\left(\underbrace{y^{(i)}}_{\text{truth}} - \underbrace{\sigma(\vec{\theta}^T \vec{x}^{(i)})}_{\text{prob of } y=1}\right)}_{\text{scalar}} \underbrace{x_m^{(i)}}_{m^{th}\text{ feature}}$$