_/_/_

19 Dec. MACHINE LEARNING ALGORITHMS.

No need to explicitly program, the machine learns by itself.

- Supervised learning: Huge, labelled data.
- Unsupervised learning: Clustering; unlabelled data.
- Semi-supervised learning: Some are labelled, some not.
- Reinforcement learning: Carrot + stick method?.
- Deep (Generative) learning: Language model, generate something, layers of Neural Nets.

① Supervised Learning:

Regression problem: Outputs a real number.
Classification problem: Binary or Multiclass.

$\mathcal{H}$ - Curly H : All possible hypotheses(?).
$h \in \mathcal{H}$, h is the ~~of~~ needed hypothesis.

31 Dec Mapping:

$\text{inputs} \rightarrow \boxed{\text{MODEL}} \rightarrow \text{outputs}$

$\downarrow$

Mathematical model:
family of math equations.

Training a model: finding parameters that predict outputs 'well' from inputs for a training dataset of input/output pairs.

_/_/_

Parameters: $y = mx + b$.
Here, m & b influence the relationship b/w y & x.

Regression: finding the best possible relationship b/w x & y.

Need for a generalized model: a model that isn't made specifically for the dataset alone, but works for more data with good results.

* Capital bold X - matrix.
* Bold variable - ~~scalar value~~ vector.
* normal variable - scalar value; single value.

* Parameters: ϕ ; Model: $y = f[x_i, \phi]$.

* Loss Function:
Measures how bad the model is.

Find the parameter that minimize the loss:

$$\hat{\phi} = \underset{\phi}{\mathrm{argmin}} \, [L[\phi]].$$

Testing a model: Separate test data with input/output pairs.
↳ checks how well it generalizes.

Supervised: mapping from one input to multiple output [classification model].
mapping from multiple input to one output [regression model].

$$L[\phi] = \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2.$$

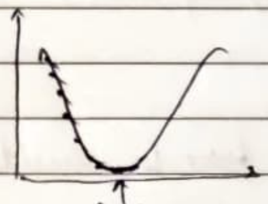
↳ least square loss function.

Gradient Descent Algorithm:

Take / Compute the slope at a point;

Equate to 0;

Check next point, if it is greater than current point, current point is a minimum.



(minima).

16 Jan.

* Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \dots$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix}$$ - parameters / weights.

$$h(x) = \sum_{i=0}^{n} \theta_i^T x_i .$$

* Cost Function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x_i) - y_i)^2$$

Nerva –
Play –

Take house price dataset and write code for Gradient Descent with only 2 features, in python [Find convergence rate, experiment with different learning rate.] _/_/_

* Gradient Descent Algorithm:

$$\theta_j := \theta_j - \alpha \nabla_\theta J(\theta).$$

where $\nabla = \frac{\partial}{\partial \theta_j}$.

α = learning rate.

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2}(h_\theta(x) - y)^2.$$

[Partial deriv cause θ_j ~~has~~ is multivariate].

$$\frac{\partial}{\partial \theta_j} J(\theta) = 2 \cdot \frac{1}{2} (h_\theta(x) - y) \frac{\partial}{\partial \theta_j} (h_\theta(x) - y).$$

$$= (h_\theta(x) - y) \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right).$$

$$\nabla_\theta J(\theta) = (h_\theta(x) - y) x_j.$$

in case → Multiple variables, then it is Hessian matrix.

$$\Rightarrow \theta_j := \theta_j + \alpha(y_i - h_\theta(x_i)) x_j.$$

↳ LHS - update rule: Least Mean Square Update Rule.
Also called: Widrow - Hoff Learning Rule.

Repeat until convergence; when the error term $(y_i - h_\theta(x_i)) x_j$ does not change, hence θ_j does not change.

28 Jan

* Logistic Regression:

Email classifier [Binary]:

$y = \{Spam, Ham\}$.

$$h_\theta(x) = g(\theta^T x).$$

$$= \left[\frac{1}{1+e^{-\theta^T x}}\right]. \quad \text{(Sigmoid)}.$$

$$g(z) = \frac{1}{1+e^{-z}}.$$

$g(z)$ is always bounded between 0 & 1.

$$g'(z) = \frac{d}{dz} \frac{1}{1+e^{-z}}.$$

z vs z ^^?

$$= \frac{1}{(1+e^{-z})^2} e^{-z}.$$

$$= \left[\frac{1}{(1+e^{-z})}\right]\left[1 - \frac{1}{(1+e^{-z})}\right].$$

$$g'(z) = g(z)(1-g(z)).$$

Assume that:

$$P(y=1 \mid x;\theta) = h_\theta(x).$$
$$P(y=0 \mid x;\theta) = (1-h_\theta(x)).$$

$$\boxed{P(y \mid x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y}}$$

Likelihood: $L(\theta) = P(y \mid x;\theta)$

$$= \prod_{i=1}^{n} P(y^i \mid x^i;\theta).$$

HW: Compare the ~~standard~~ stochastic descent rule of linear regression and stochastic ascent rule of logistic regression. What similarities do you observe? Is it a coincidence?

$$L(\theta) = P(y \mid x; \theta)$$

$$= \prod_{i=1}^{n} P(y_i \mid x_i ; \theta).$$

$$= \prod_{i=1}^{n} (h_\theta(x_i))^{y_i} (1 - h_\theta(x_i))^{1-y_i}$$

log likelihood:

$$l(\theta) = \log L(\theta).$$

$$l(\theta) = \sum_{i=1}^{n} y_i \log(h_\theta(x_i)) + (1-y_i) \log(1 - h_\theta(x_i)).$$

Maximize ~~the~~ likelihood:

$$\frac{\partial\, l(\theta)}{\partial \theta_j} = \left[y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1 - g(\theta^T x)} \right] \frac{\partial}{\partial \theta_j} g(\theta^T x).$$

$y \in \{0, 1\}$. $\Rightarrow h_\theta(x) = g(z) = g(\theta^T x)$.

$$\frac{\partial\, l(\theta)}{\partial \theta_j} = [\quad] \, g(\theta^T x)(1 - g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x.$$

$$= (y(1 - g(\theta^T x)) - (1-y) g(\theta^T x)) x_j$$

$$= (y - h_\theta(x)) x_j.$$

Stochastic Gradient Ascent

$$\theta_j := \theta_j + \alpha (y_i - h_\theta(x_i)) x_{ij}$$

[From $\theta = \theta + \alpha \nabla_\theta l(\theta)$].

_/_/_

* Bayesian Decision Theory:

Classification : class w_1 , class w_2.

Accept product (class w_1) ; Reject product (class w_2).

Decision Rule:

REJECTED

$P(w_1) > P(w_2) \Rightarrow w_1$.
$P(w_1) < P(w_2) \Rightarrow w_2$.
} Sometimes good, but not always.

$P(w_1)$, $P(w_2)$ are called Apriori probabilities.
Decision is based on some feature x.

Probability density function of variable x given w_1 or w_2:

$P(x|w_1)$; $p(x|w_2)$. $\rightarrow$ Class conditional probabilities

Decision is based on: $p(w_1|x)$; $p(w_2|x)$.

Joint probability: $P(w_i, x) = P(w_i|x)P(x)$. OR $P(x|w_i)P(w_i)$. [Product rule].

$$\Rightarrow \boxed{P(w_i|x) = \frac{P(x|w_i)P(w_i)}{P(x)}}$$

where $p(x) = \sum P(x|w_i)P(w_i)$.

Decision Rule:

$P(w_1|x) > P(w_2|x) \Rightarrow w_1$.
$P(w_1|x) < P(w_2|x) \Rightarrow w_2$.

(Jan 31 1996)

Exam portion : Moodle - first 4 chapters of textbook.

Exam time 9-11 ?? Why not 10-12 ?

$\boxed{P(x|w_1)P(w_1) > P(x|w_2)P(w_2) \Rightarrow w_1 .}$

Here, the decision is made based on apriori probabilities and class conditional probabilities.

6 Feb

* Parametric methods:

- Maximum Likelihood estimation (MLE):

Baye's Theorem: $\underset{\text{Posterior}}{P(\theta|D)} \propto \underset{\text{likelihood}}{P(D|\theta)} P(\theta)$ → joint?

$P(X|\theta)$ ← likelihood.

$$l(\theta|X) = P(X|\theta) = \prod_{t=1}^{N} P(x_t|\theta).$$

$X = \{x_t\}_{t=1}^{N}$ ← Draw some (N) data from a known distribution, $P(X|\theta)$.

$x_t \sim P(x|\theta)$. ; $\theta = \begin{bmatrix} \mu \\ \sigma \end{bmatrix}$. $\theta_1 = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}$ ... $\theta_p = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix}$.

log likelihood function :

$$L(\theta|X) = \log l(\theta|X)$$
$$= \sum_{t=1}^{N} \log P(X_t|\theta).$$

Bernoulli Distribution : Ber(p). ; Binomial(n, p).

$$P(x) = p^{x}(1-p)^{1-x} \quad ; \quad x \in \{0,1\}.$$

$$L(p|X) = \log \prod_{t=1}^{N} P(x_t)(1-p)^{(1-x_t)}.$$

$$= \sum_{t=1}^{N} x_t \log p + (N - \sum_t x_t) \log(1-p).$$

Galyani

Hoikod?

_/_/_

Maximize : Estimate $\hat{p} = \cancel{\sum_t / \frac{x^t}{N}}$

$$\hat{p} = \frac{\sum_t x^t_t}{N}.$$

It is called estimate.

* Bias and Variance:

X f θ. Bias of a graph distribution, not mean value.

Estimator of θ ← $d = f(X)$. → $\hat{P}(\theta)$.

Error $= (d(X) - \theta)^2$.

$$r(d, \theta) = E\left[(d(X) - \theta)^2\right].$$

Mean Square Error.

Bias of an estimator : $b_\theta(d) = \underline{E[d(X)]} - \theta$.

What does unbiased estimator mean?

For all θ, $b_\theta = 0$, we call it unbiased estimator.

→ number of samples

$$E[m] = E\left[\frac{\sum_t x_t}{N}\right] = \frac{1}{N} \frac{\sum_t x_t}{N} = \frac{N\mu}{N} = \mu.$$

→ concept used similar to.

Law of large numbers : When sample size ↑, sample mean ⇝ actual mean.

Variance: $Var(m) = Var\left[\frac{\sum_t x_t}{N}\right] = \frac{\sigma^2}{N}.$

$Var = E[x^2] - (E[x])^2.$

Mean Square Error:

~~$r(d,\theta) = E[(d(X)-\theta)]^2$~~

$$r(d,\theta) = E[(d(x)-\theta)^2].$$

$$= \underbrace{E[(d - E[d])^2]}_{\text{variance}} + \underbrace{(E(d)-\theta)^2}_{\text{bias}}.$$

*

$r(d,\theta) = Var(d) + b_\theta(d)^2$.

Error = Variance + bias^2.