Contents

Lec 1 Linear Regression: I

Def 1.1 (supervised learning)

Supervise Learning: Given data X in feature space and labels Y, learn to predict Y from given X. ♣

Label: 可以是 discrete 的 or continuous 的.

对于 **discrete** 的 label, 这类问题称为 **classification**.

对于 **continuous** 的 label, 这类问题称为 **regression**.

1.1 notation and expression

我们使用以下 notation:

Notation $x \in \mathbb{R}^d$: data

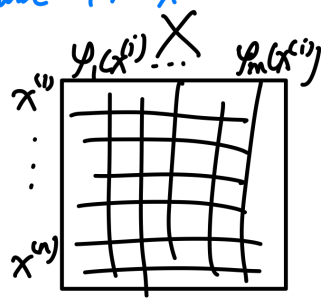
$\varphi(x) \in \mathbb{R}^m$: features for x

$\varphi_j(x) \in \mathbb{R}$: the j^{th} feature for x

$y \in \mathbb{R}$: ctn label

$x^{(n)}$: the n^{th} training example

$y^{(n)}$: the n^{th} training label



Def 1.2 ((generalized) linear regression)

给定 N 个 data points $\{(x^{(n)}, y^{(n)})\}_{n=1,\cdots,N}$ where each $x^{(n)} \in \mathbb{R}^d, y^{(n)} \in \mathbb{R}$,

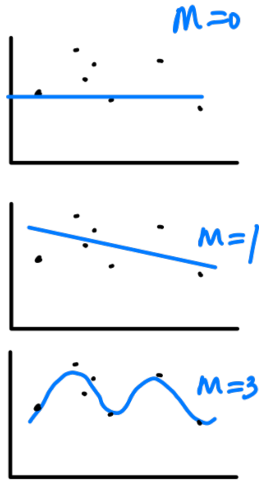
以及预先设定好的 M 个 basis functions $\{\phi_i(x)\}_{i=1,\cdots,M}$ 用以表示 M 个 features,

我们通过建立一个 $h(x, w) : \mathbb{R}^d \times \mathbb{R}^M \to \mathbb{R} = \sum_{i=0}^{M-1} w_i \phi_i(x)$, 使其关于 w 线性, 以找到一组参数 $w \in \mathbb{R}^M$, 使得 $h(x^{(n)}, w)$ 能够近似 $y^{(n)}$ for each n, with respect to the loss function we define to measure the distance between two vectors. ♣

1-d case $D=1$

$$\{x^{(1)},\dots,x^{(N)}\},\ \{y^{(1)},\dots,y^{(N)}\}$$

We want: $h(\underset{\text{input}}{x};\underset{\text{parameter}}{w}) \approx y$



general case

$$h(x;w) = \underbrace{w_0}_{\text{bias term}} + \sum_{j=1}^{M-1} w_j \varphi_j(x)$$

$$= w^T\varphi(x) \quad \varphi_0(x)=1$$

$$(= w\cdot\varphi(x))$$

$w=(w_0,\dots,w_{M-1})^T$, $\varphi(x)=(1,\varphi_1(x),\dots,\varphi_{M-1}(x))^T$

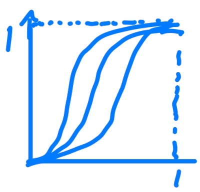
$$w=\begin{bmatrix} w_0 \\ \vdots \\ w_{M-1}\end{bmatrix},\quad \varphi(x)=\begin{bmatrix} 1 \\ \varphi_1(x) \\ \vdots \\ \varphi_{M-1}(x)\end{bmatrix}$$

Remark 注意: linear regression 指的是 $y \in \mathbb{R}$ 和参数 $w \in \mathbb{R}^M$ 之间是 linear 的, 而不是说 y 和 input x 之间是 linear 的. 我们可以选择 nonlinear 的 basis funtions 来 encode x 来表示 features 的特性, 比如我们可以选择:

Ex $\varphi_j(x) = x^j$ (polynomial)

$\varphi_j(x) = \exp\left(-\dfrac{(x-\mu_j)^2}{2s^2}\right)$ (Gaussian)

$\varphi_j(x) = \dfrac{1}{1+\exp\left(-\frac{x-\mu_j}{s}\right)}$ (sigmoid)



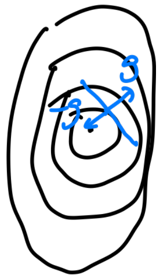
1.2 loss function: sum of squared error

这个 loss function 衡量两个 vectors 之间的距离, 目的是衡量 $y \in \mathbb{R}^N$ 和 $h(x,w) \in \mathbb{R}^N$ 这两个 vectors 的差距. 实际上就是它们 difference 的 L_2-norm 的平方.

We use: sum of squares error

$$E(w)=\underbrace{\frac{1}{2}\sum_{n=1}^{N}\left\|h(x^{(n)},w)-y^{(n)}\right\|_{L2}^{2}}$$

$$=\frac{1}{2}\sum_{n=1}^{N}\left(\sum_{i=0}^{M-1}w_i\varphi_i(x^{(n)})-y^{(n)}\right)^2$$



$$\nabla E(w)=\begin{bmatrix}\frac{\partial E}{\partial w_0}(w)\\ \vdots\\ \frac{\partial E}{\partial w_m}(w)\end{bmatrix}$$

$$\text{where }\frac{\partial E}{\partial w_k}(w)=\frac{\partial}{\partial w_k}\left[\frac{1}{2}\sum_{n=1}^{N}\left(\sum_{i=0}^{M-1}w_i\varphi_i(x^{(n)})-y^{(n)}\right)^2\right]$$

1.3 gradient of sum of squared error

我们下面首先通过求 $\nabla E(w)$ 的每个 entry $\frac{\partial E}{\partial w_k}(w)$ 来写出这个 gradient.

$$\frac{\partial E}{\partial w_k}(w)=\frac{\partial}{\partial w_k}\left[\frac{1}{2}\sum_{n=1}^{N}\left(\sum_{i=0}^{M-1}w_i\varphi_i(x^{(n)})-y^{(n)}\right)^2\right]$$

$$=\frac{1}{2}\sum_{n=1}^{N}\frac{\partial}{\partial w_k}\left(w_0\varphi_0(x^{(n)})+\dots+w_k\varphi_k(x^{(n)})+\underbrace{\dots+w_{m-1}\varphi_{m-1}(x^{(n)})-y^{(n)}}_{\text{const}}\right)^2$$

即 $\frac{d}{dw_k}(aw_k+b)^2=2a(aw_k+b)$

where $a=\varphi_k(x^{(n)})$

$$=\sum_{n=1}^{N}\left[\left(\sum_{i=0}^{M-1}w_i\varphi_i(x^{(n)})\right)-y^{(n)}\right]\varphi_k(x^{(n)})$$

$$=$$

So

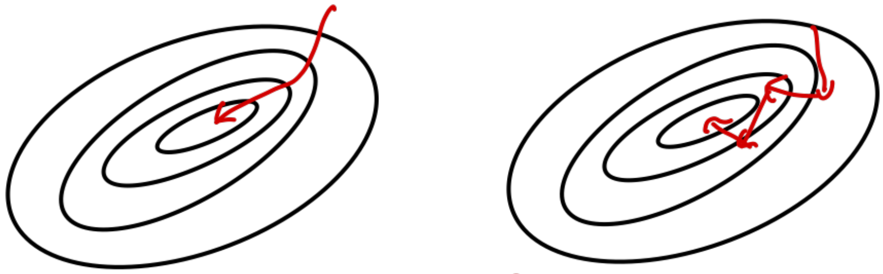
$$\nabla E(w)=\sum_{n=1}^{N}\left\{\underbrace{\left(\left(\sum_{i=0}^{M-1}w_i\varphi_i(x^{(n)})\right)-y^{(n)}\right)}^{(\text{pred}-\text{actual})\text{ for }x^{(n)}}\begin{bmatrix}\varphi_0(x^{(n)})\\ \vdots\\ \varphi_{m-1}(x^{(n)})\end{bmatrix}\right\}$$

$$=\sum_{n=1}^{N}\left(\left(w^T\varphi(x^{(n)})-y^{(n)}\right)\varphi(x^{(n)})\right)$$

$$=\sum_{n=1}^{N}\left(\left(\text{pred}^{(n)}-\text{actual}^{(n)}\right)\text{feature.vector}^{(n)}\right)$$

1.4 batch v.s. stochastic GD

我们通过迭代降低 gradient 来降低 loss function 的值, 从而优化 weight vector.



III. Batch Gradient Desent v.s Stochastic Gradient Descent

repeat till convergence:

$$w := w - \eta \nabla E(w)$$

repeat until conv:

Randomly suffle N samples

for $n = 1, \dots, N$

$$w := w - \eta \nabla E(w|x^{(n)})$$

note: $\nabla E(w) = \sum_{n=1}^{N} (pred^{(n)} - actual^{(n)}) feature^{(n)}$

$\nabla E(w|x^{(n)}) = (pred^{(n)} - actual^{(n)}) feature^{(n)}$

为其中一项

Rmk usually set $\eta_t \propto \frac{1}{t}$ 或 $\eta_t = \eta_1 \frac{1}{1+\frac{t-1}{r}}$

More practically, 我们可以采用 minibatch SGD: 即在 batch GD 和 SGD 之间, 每次选择一小部分 samples, 称为一个 **minibatch**, 在这个 minibatch 上进行 GD.

Lec 2 Linear Regression: II

2.1 vectorization

我们可以把每个 $x^{(n)}$ 的 features 写成一个 row vector, 并 stack up N 个 row vectors, 成为一个 $N \times M$ 的 matrix Φ. 从而:

$$h(x, w) = \Phi w$$

vectorization 的好处是: 1. 便于手算; 2. computer 可以进行并行计算.

$$\text{let } \Phi = \begin{pmatrix} \varphi_0(x^{(1)}) & \cdots & \varphi_{M-1}(x^{(1)}) \\ \vdots & & \vdots \\ \varphi_0(x^{(N)}) & \cdots & \varphi_{M-1}(x^{(N)}) \end{pmatrix} \Rightarrow \underbrace{h(x,w) = \overset{\Phi}{\Phi} \overset{\in\mathbb{R}^M}{w}}_{\in\mathbb{R}^{N\times M}}$$

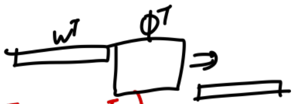
$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left(w^T\varphi(x^{(n)}) - y^{(n)}\right)^2$$

$E: \mathbb{R}^M \to \mathbb{R}$

$$= \frac{1}{2}\|\Phi w - y\|^2$$

$$= \frac{1}{2}(\Phi w - y)^T(\Phi w - y)$$

$$= \frac{1}{2}(w^T\Phi^T - y^T)(\Phi w - y)$$



$$= \frac{1}{2}\Big(w^T\Phi^T\Phi w - \underbrace{w^T\Phi^T y - y^T\Phi w}_{\text{the same, scalar}} + y^T y\Big)$$

$$= \frac{1}{2}w^T\Phi^T\Phi w - w^T\Phi^T y + \frac{1}{2}y^T y$$

(then recall chain rule to differentiate $f: \mathbb{R}^m \to \mathbb{R}^n$)

计算得 linear regression 的 loss function 为:

$$E(w) = \frac{1}{2}w^T\Phi^T\Phi w - w^T\Phi^T y + \frac{1}{2}y^T y$$

2.2 closed-form solution

如果

$$\nabla E(w) = y$$

有一个 closed form solution, 那么这个 solution 一定是一个 local min/max, 从而 possibly 成为一个 global min.

为了计算 closed form solution, 我们首先要给出 $\nabla E(w)$ 的 matrix form 表达式.

这里首先引入 linear form 和 quadratic form ($\mathbb{R}^m \to \mathbb{R}$) 的 gradient:

linear form:

$$f(x) = b^T x \Rightarrow \nabla_x f(x) = b \quad \text{since } \frac{\partial f(x)}{\partial x_k} = b_k$$

quadratic form: $\left(= \sum_{i,j=1}^{n} x_i A_{ij} x_j\right)$

$$f(x) = x^T A x \Rightarrow \nabla_x f(x) = 2Ax \quad \text{since } \frac{\partial f}{\partial x_k}(x) = 2\sum_{j=1}^{n} A_{kj} x_j = 2(Ax)_k$$

我们发现: $E(w)$ 就是一个 w 的 quadratic form, 一个 w 的 linear form 和一个 const 的组合. 从而可以求出:

$$\nabla E(w) = \frac{1}{2}\nabla\underbrace{(w^T\Phi^T\Phi w)}_{\substack{\text{quadratic form} \\ w^T(\Phi^T\Phi)w \\ \xrightarrow{\text{diff}} 2\Phi^T\Phi}} - \nabla\underbrace{(w^T\Phi^T y)}_{\substack{\text{linear form} \\ w^T(\Phi^T y) \\ = w\cdot(\Phi^T y) \\ = (\Phi^T y)^T w \\ \xrightarrow{\text{diff}} \Phi^T y}} + \frac{1}{2}\nabla\underbrace{(y^T y)}_{\substack{\text{const} \\ \text{diff: } 0}}$$

$$= \Phi^T\Phi w - \Phi^T y$$

$$= \Phi^T(\underbrace{\Phi w}_{h(x,w)} - y) = \Phi^T(\text{pred\_y} - \text{actual\_y})$$

从而我们得到 closed form solution (if exists):

$$\nabla E(w) = \Phi^T(\Phi w - y) := 0$$

$$\Rightarrow \Phi^T\Phi w = \Phi^T y$$

$$\Rightarrow w_{ML} = (\Phi^T\Phi)^{-1}\Phi^T y \quad \text{(if exists)}$$

因而 closed form exists iff $\Phi^T\Phi$ 可逆, iff Φ 可逆.

并且 recalll in linear algebra: $\text{rank}(\Phi^T\Phi) = \text{rank}(\Phi)$. 因而, **closed form exists iff** $M >= N$ **且** $\text{rank}(\Phi) = N$.