Lecture 2: Linear Regression

Feng Li

Shandong University

fli@sdu.edu.cn

September 13, 2023

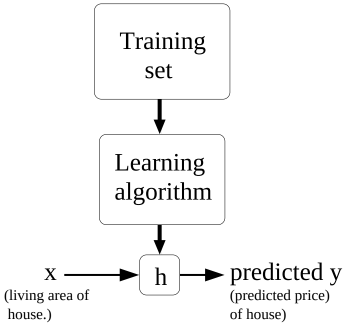
Supervised Learning

- Regression: Predict a continuous value
- Classification: Predict a discrete value, the class

Living area ($feet^2$)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

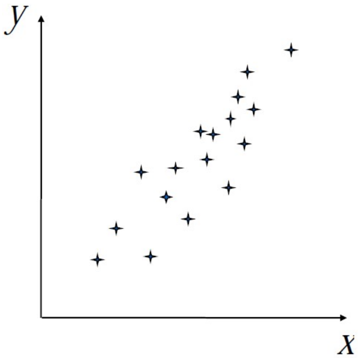
Supervised Learning (Contd.)

- Features: input variables, x;
- Target: output variable, y;
- Training example: $(x^{(i)}, y^{(i)})$, $i = 1, 2, 3, ..., m$
- Hypothesis: $h : \mathcal{X} \rightarrow \mathcal{Y}$.

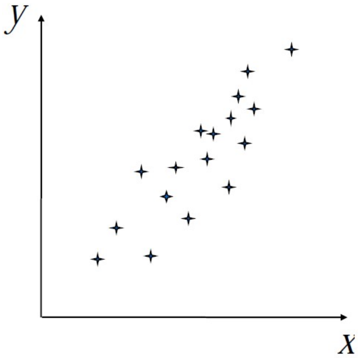


Linear Regression

- Linear hypothesis: $h(x) = \theta_1 x + \theta_0$.
- θ_i ($i = 1, 2$ for 2D cases): Parameters to estimate.
- How to choose θ_i's?

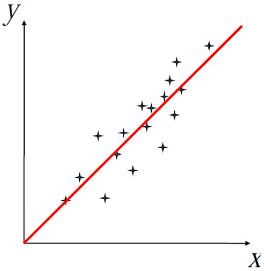


Linear Regression (Contd.)



- Input: Training set $(x^{(i)}, y^{(i)}) \in \mathbb{R}^2$ $(i = 1, ..., m)$
- Goal: Model the relationship between x and y such that we can predict the corresponding target according to a given new feature.

Linear Regression (Contd.)



- The relationship between x and y is modeled as a linear function.
- The linear function in the 2D plane is a straight line.
- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$ (where θ_0 and θ_1 are parameters)

Linear Regression (Contd.)

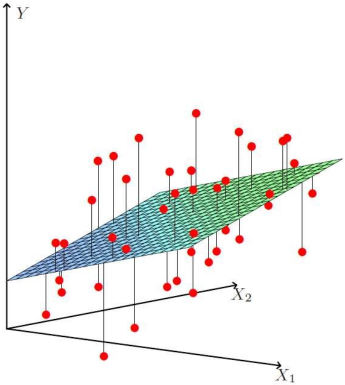
- Given data $x \in \mathbb{R}^n$, we then have $\theta \in \mathbb{R}^{n+1}$
- Thus $h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$, where $x_0 = 1$
- What is the best choice of θ ?

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

where $J(\theta)$ is so-called a cost function

Linear Regression (Contd.)

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



Gradient

Definition

Directional Derivative: The directional derivative of function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ in the direction $u \in \mathbb{R}^n$ is

$$\nabla_u f(x) = \lim_{h \rightarrow 0} \frac{f(x + hu) - f(x)}{h}$$

- $\nabla_u f(x)$ represents the rate at which f is increased in direction u
- When u is the i-th standard unit vector e_i,

$$\nabla_u f(x) = f_i'(x)$$

where $f_i'(x) = \frac{\partial f(x)}{\partial x_i}$ is the partial derivative of $f(x)$ w.r.t. x_i

Gradient (Contd.)

Theorem

For any n-dimensional vector u, the directional derivative of f in the direction of u can be represented as

$$\nabla_u f(x) = \sum_{i=1}^{n} f_i'(x) \cdot u_i$$

Gradient (Contd.)

Proof.

Letting $g(h) = f(x + hu)$, we have

$$g'(0) = \lim_{h \to 0} \frac{g(h) - g(0)}{h} = \lim_{h \to 0} \frac{f(x + hu) - g(0)}{h} = \nabla_u f(x) \qquad (1)$$

On the other hand, by the chain rule,

$$g'(h) = \sum_{i=1}^{n} f_i'(x) \frac{d}{dh}(x_i + hu_i) = \sum_{i=1}^{n} f_i'(x) u_i \qquad (2)$$

Let $h = 0$, then $g'(0) = \sum_{i=1}^{n} f_i'(x) u_i$, by substituting which into (1), we complete the proof. □

Gradient (Contd.)

Definition

Gradient: The gradient of f is a vector function $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\nabla f(x) = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} e_i$$

where e_i is the i-th standard unit vector. In another simple form,

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n} \right]^T$$

Gradient (Contd.)

- $\nabla_u f(x) = \nabla f(x) \cdot u = \|\nabla f(x)\| \|u\| \cos a$ where a is the angle between $\nabla f(x)$ and u
- Without loss of generality, assume u is a unit vector,

$$\nabla_u f(x) = \|\nabla f(x)\| \cos a$$

- When $u = \nabla f(x)$ such that $a = 0$ (and thus $\cos a = 1$, we have the maximum directional derivative of f, which implies that $\nabla f(x)$ is **the direction of steepest ascent** of f.

Gradient Descent (GD) Algorithm

- If the multi-variable function $J(\theta)$ is differentiable in a neighborhood of a point θ, then $J(\theta)$ decreases fastest if one goes from θ in the direction of the negative gradient of J at θ
- Find a local minimum of a differentiable function using gradient descent

Algorithm 1 Gradient Descent

Given a starting point $\theta \in$ **dom** J
repeat
    Calculate gradient $\nabla J(\theta)$;
    Update $\theta \leftarrow \theta - \alpha \nabla J(\theta)$
until convergence criterion is satisfied

- θ is usually initialized randomly
- α is so-called learning rate

GD Algorithm (Contd.)

- Stopping criterion (i.e., conditions to convergence)
  - the gradient has its magnitude less than or equal to a predefined threshold (say ε), i.e.

    $$\|\nabla f(x)\|_2 \leq \varepsilon$$

    where $\|\cdot\|_2$ is ℓ_2 norm, such that the values of the objective function differ very slightly in successive iterations
  - Set a fixed value for the maximum number of iterations, such that the algorithm is terminated after the number of the iterations exceeds the threshold.

GD Algorithm (Contd.)

- In more details, we update each component of θ according to the following rule

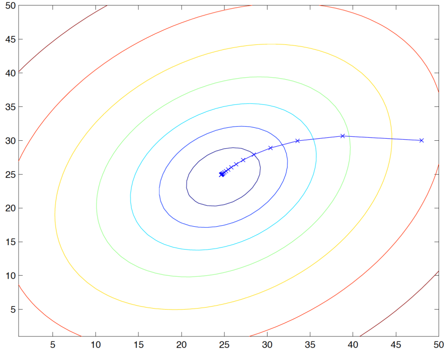
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}, \quad \forall j = 0, 1, \cdots, n$$

- Calculating the gradient for linear regression

$$\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2 \\
&= \frac{\partial}{\partial \theta_j} \frac{1}{2} \sum_{i=1}^{m} (\sum_{j=0}^{n} \theta_j x_j^{(i)} - y^{(i)})^2 \\
&= \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)}) x_j^{(i)}
\end{aligned}$$

GD Algorithm (Contd.)

- An illustration of gradient descent algorithm
- The objective function is decreased fastest along the gradient



GD Algorithm (Contd.)

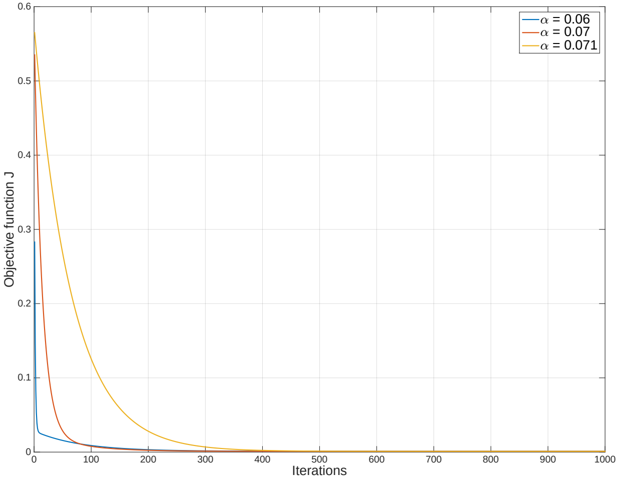
- Another commonly used form

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

- What's the difference?
  - m is introduced to scale the objective function to deal with differently sized training set.
- Gradient ascent algorithm
  - Maximize the differentiable function $J(\theta)$
  - The gradient represents the direction along which J increases fastest
  - Therefore, we have

$$\theta_j \leftarrow \theta_j + \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

Convergence under Different Step Sizes



Stochastic Gradient Descent (SGD)

- What if the training set is huge?
  - In the above batch gradient descent algorithm, we have to run through the entire training set in each iteration
  - A considerable computation cost is induced!
- Stochastic gradient descent (SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization method
  - In each iteration, the parameters are updated according to the gradient of the error with respect to one training sample only

Stochastic Gradient Descent (Contd.)

Algorithm 2 Stochastic Gradient Descent for Linear Regression

Given a starting point $\theta \in \textbf{dom } J$
repeat
  Randomly shuffle the training data;
  for $i = 1, 2, \cdots, m$ **do**
    $\theta \leftarrow \theta - \alpha \nabla J(\theta; x^{(i)}, y^{(i)})$
  end for
until convergence criterion is satisfied

More About SGD

- The objective does not always decrease for each iteration
- Usually, SGD has θ approaching the minimum much faster than batch GD
- SGD may never converge to the minimum, and oscillating may happen
- A variants: Mini-batch, say pick up a small group of samples and do average, which may accelerate and smoothen the convergence

Matrix Derivatives [1]

- A function $f : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}$
- The derivative of f with respect to A is defined as

$$\nabla f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

- For an $n \times n$ matrix, its trace is defined as $\mathrm{tr}A = \sum_{i=1}^{n} A_{ii}$
  - $\mathrm{tr}ABCD = \mathrm{tr}DABC = \mathrm{tr}CDAB = \mathrm{tr}BCDA$
  - $\mathrm{tr}A = \mathrm{tr}A^T$, $\mathrm{tr}(A+B) = \mathrm{tr}A + \mathrm{tr}B$, $\mathrm{tr}aA = a\mathrm{tr}A$
  - $\nabla_A \mathrm{tr}AB = B^T$, $\nabla_{A^T} f(A) = (\nabla_A f(A))^T$
  - $\nabla_A \mathrm{tr}ABA^TC = CAB + C^TAB^T$, $\nabla_A |A| = |A|(A^{-1})^T$
  - Funky trace derivative $\nabla_{A^T} \mathrm{tr}ABA^TC = B^TA^TC^T + BA^TC$

[1] Details can be found in "Properties of the Trace and Matrix Derivatives" by John Duchi

Revisiting Least Square

- Assume

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \qquad Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

- Therefore, we have

$$X\theta - Y = \begin{bmatrix} (x^{(1)})^T\theta \\ \vdots \\ x^{(m)})^T\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix}$$

- $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 = \frac{1}{2}(X\theta - Y)^T(X\theta - Y)$

Revisiting Least Square (Contd.)

- Minimize $J(\theta) = \frac{1}{2}(Y - X\theta)^T(Y - X\theta)$
- Calculate its derivatives with respect to θ

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(Y - X\theta)^T(Y - X\theta) \\
&= \frac{1}{2}\nabla_\theta (Y^T - \theta^T X^T)(Y - X\theta) \\
&= \frac{1}{2}\nabla_\theta \mathrm{tr}(Y^T Y - Y^T X\theta - \theta^T X^T Y + \theta^T X^T X\theta) \\
&= \frac{1}{2}\nabla_\theta \mathrm{tr}(\theta^T X^T X\theta) - X^T Y \\
&= \frac{1}{2}(X^T X\theta + X^T X\theta) - X^T Y \\
&= X^T X\theta - X^T Y
\end{aligned}$$

- Tip: Funky trace derivative $\nabla_{A^T} \mathrm{tr} ABA^T C = B^T A^T C^T + BA^T C$

Revisiting Least Square (Contd.)

- **Theorem:**
  The matrix $A^T A$ is invertible if and only if the columns of A are linearly independent. In this case, there exists only one least-squares solution

$$\theta = (X^T X)^{-1} X^T Y$$

- Prove the above theorem in Problem Set 1.

Probabilistic Interpretation

- The target variables and the inputs are related

$$y = \theta^T x + \epsilon$$

  - ϵ's denote the errors and are independently and identically distributed (i.i.d.) according to a Gaussian distribution $\mathcal{N}(0, \sigma^2)$
- The density of $\epsilon^{(i)}$ is given by

$$f(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

- The conditional probability density function of y

$$y \mid x; \theta \sim \mathcal{N}(\theta^T x, \sigma^2)$$

Probabilistic Interpretation (Contd.)

- The training data $\{x^{(i)}, y^{(i)}\}_{i=1,\cdots,m}$ are sampled identically and independently

$$p(y = y^{(i)} \mid x = x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- Likelihood functoin

$$\begin{aligned} L(\theta) &= \prod_i p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

Probabilistic Interpretation (Contd.)

- Maximizing the likelihood $L(\theta)$
- Since $L(\theta)$ is complicated, we maximize an increasing function of $L(\theta)$ instead

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{i} (y^{(i)} - \theta^T x^{(i)})^2
\end{aligned}
$$

- Apparently, maximizing $L(\theta)$ (thus $\ell(\theta)$) is equivalent to minimizing

$$
\frac{1}{2} \sum_{i}^{m} (y^{(i)} - \theta^T x^{(i)})^2
$$

Thanks!

Q & A