machine learning 📈 formulae and expressions

linear regression

hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_n$$

parameters:

$$\theta_0, \theta_1, \dots, \theta_n$$

cost function:

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

goal:

$$\min_{\theta_0,\theta_1,\dots,\theta_n} J(\theta_0, \theta_1, \dots, \theta_n)$$

partial derivative gradient descent for multivariate linear regression:

repeat until convergence {

$$\theta_j := \theta_j - a\frac{\partial}{\partial\theta_j}J(\theta)$$

$$\theta_j := \theta_j - a\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

(update θ_j for $j = 0, \dots, n$ simultaneously)

}

normal equation:

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

regularized multivariate linear regression:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

regularized gradient descent for multivariate linear regression: (for all j)

repeat until convergence {

$$\theta_0 \coloneqq \theta_0 - a\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_0^{(i)}$$

$$\theta_j \coloneqq \theta_j - a\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

(update θ_j for $j = \cancel{0}, 1, 2, 3 \ldots, n$ simultaneously)

}

alternative notation for θ_j update: $\theta_j \coloneqq \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$

regularized normal equation:

$$\theta = \left(X^T X + \lambda\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right)^{-1} X^T y$$

logistic regression

hypothesis:

$$h_\theta(x) = g(\theta^T X) \quad \rightarrow \quad g(z) = \frac{1}{1+e^{-z}}$$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T X}} \quad \rightarrow \quad \text{sigmoid/logistic function}$$

interpretation:

$$h_\theta(x) = P(y = 1 | x : \theta)$$

cost function:

$$\mathrm{cost}\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = \frac{1}{2}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$\mathrm{cost}\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(h_\theta\left(x^{(i)}\right)\right) & \text{if } y = 1 \\ -\log\left(1 - h_\theta\left(x^{(i)}\right)\right) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{cost}\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)$$

$$= \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

partial derivative gradient descent for multivariate logistic regression:

repeat until convergence {

$$\theta_j := \theta_j - a\frac{\partial}{\partial\theta_j}J(\theta)$$

$$\theta_j := \theta_j - a\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

(update θ_j for $j = 0, \ldots, n$ simultaneously)

}

$h_\theta(x) = \theta^T X$ to $h_\theta(x) = \frac{1}{1+e^{-\theta^T X}}$

regularized multivariate logistic regression:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{i=1}^{u}\theta_j^2$$

regularized gradient descent for multivariate logistic regression: (for all j)

repeat until convergence {

$$\theta_0 := \theta_0 - a\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_0^{(i)}$$

$$\theta_j := \theta_j - a\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

(update θ_j for $j = \cancel{0}, 1, 2, 3 \ldots, n$ simultaneously)

}

$h_\theta(x) = \theta^T X$ to $h_\theta(x) = \frac{1}{1+e^{-\theta^T X}}$

neural networks

backpropagation algorithm

gradient computation

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{i=1}^{K} y_k^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right)_k + \left(1 - y_k^{(i)}\right)\log\left(1 - \left(h_\theta\left(x^{(i)}\right)\right)_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_1}\sum_{j=1}^{s_1+1}\left(\theta_{ji}^{(l)}\right)^2$$

$$\min_\theta J(\theta)$$

gradient checking

$$\frac{\partial}{\partial\theta}J(\theta) \approx \frac{J(\theta+\varepsilon) - (\theta-\varepsilon)}{2\varepsilon}$$

train · validation · test error

training error:

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)^2$$

cross validation error:

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta\left(x_{cv}^{(i)}\right), y_{cv}^{(i)}\right)^2$$

test error:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(h_\theta\left(x_{test}^{(i)}\right), y_{test}^{(i)}\right)^2$$

precision:

$$\frac{\text{true positives}}{\#\ \textbf{predicted}\ \text{positives}} = \frac{\text{true positives}}{\text{true positives} + \textbf{false positives}}$$

recall:

$$\frac{\text{true positives}}{\#\ \textbf{actual}\ \text{positives}} = \frac{\text{true positives}}{\text{true positives} + \textbf{false negatives}}$$

support vector machines

optimization objective:

$$\min_\theta \frac{1}{m} C \sum_{i=1}^{m} y^{(i)} \operatorname{cost}_1(\theta^T x^{(i)}) + \left(1 - y^{(i)}\right) \operatorname{cost}_0\left(\theta^T x^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

hypothesis output:

$$h_\theta x = \begin{cases} 1 \text{ if } \theta^T x \text{ is } \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

decision boundary:

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}(\theta_1^2 + \theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

kernels

similarity:

$$f_1 = \text{similarity}(x, \ell^{(1)}) = \exp\left(-\frac{\left\|x - \ell^{(1)}\right\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}\left(x_j - \ell_j^{(1)}\right)^2}{2\sigma^2}\right)$$

k-means algorithm

optimization objective:

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_c{}^{(i)}\right\|^2$$

anomaly detection algorithm

gaussian distribution:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n}\frac{1}{\sqrt{2\pi}\sigma^2}\exp\left(-\frac{\left(x_j - \mu_j\right)^2}{2\sigma_j^2}\right)$$

multivariate gaussian distribution:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}\,|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\,\Sigma^{-1}(x-\mu)\right)$$

recommender systems

optimization objective:

to learn $\theta^{(j)}$ (parameter for user j):

$$\min_{\theta^{(j)}}\frac{1}{2}\sum_{i:r(i,j)=1}\left(\left(\theta^{(j)}\right)^T\left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

to learn $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(n_u)}$:

$$\min_{\theta^{(j)},\dots,\theta^{(n_u)}}\frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}\left(\left(\theta^{(j)}\right)^T\left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

simultaneous gradient descent update:

(for $k = 0$)

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T \left(x^{(i)}\right) - y^{(i,j)} \right) x_k^{(i)}$$

(for $k \neq 0$)

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T \left(x^{(i)}\right) - y^{(i,j)} \right) x_k^{(i)} + \lambda\, \theta_k^{(j)} \right)$$

collaborative filtering optimization algorithm:

given $\theta^{(1)}, \dots, \theta^{(n_u)}$, to learn $x^{(i)}$:

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T \left(x^{(i)}\right) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)}\right)^2$$

given $\theta^{(1)}, \dots, \theta^{(n_u)}$, to learn $x^{(1)}, \dots, x^{(n_m)}$:

$$\min_{x^{(i)}, \dots, x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T \left(x^{(i)}\right) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(i)}\right)^2$$

collaborative filtering algorithm:

given features $x^{(1)}, \dots, x^{(n_m)}$, estimate parameters $\theta^{(1)}, \dots, \theta^{(n_u)}$:

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2$$

given parameters $\theta^{(1)}, \dots, \theta^{(n_u)}$, estimate features $x^{(1)}, \dots, x^{(n_m)}$:

$$\min_{x^{(1)}, \dots, x^{(m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(j)}\right)^2$$

minimizing $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ simultaneously:

$$J\left(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}\right)$$

$$= \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2$$

collaborative filtering algorithm update:

$$x_k^{(i)} := x_k^{(i)} - \alpha \left(\sum_{j:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right) \theta_k^{(j)} + \lambda\, x_k^{(i)} \right)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right) x_k^{(i)} + \lambda\, \theta_k^{(j)} \right)$$