Introduction to Machine Learning

October 10, 2018

1 Lecture 1

linear function

$$y = f_{\mathbf{W}}(\mathbf{X}) = f(\mathbf{X}, \mathbf{W}) = \mathbf{W}^{\mathrm{T}}\mathbf{X}$$

linear classifier (perception model)

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \cdot \mathbf{w} + b < 0$$

linear regression in 1 dimension

$$y^i = \mathbf{W}^{\mathrm{T}}\mathbf{X}^i + \epsilon^i$$

where ϵ is the noise(loss).

Loss function: sum of squared errors

$$L(\mathbf{W}) = \sum_{i=1}^{N} (\epsilon^i)^2$$

$$L(w_0, w_1) = \sum_{i=1}^{N}$$

$$\frac{\partial L(w_0, w_1)}{\partial w_0} = \sum_{i=1}^{N} \frac{\partial [y^i - (w_0 x_0^i + w_1 x_1^i)]^2}{\partial w_0} = -2 \sum_{i=1}^{N} (y^i - (w_0 x_0^i + w_1 x_1^i)) x_0^i = 0$$

$$\sum_{i=1}^{N} y^i x_0^i = w_0 \sum_{i=1}^{N} x_0^i x_0^i + w_1 \sum_{i=1}^{N} x_1^i x_0^i$$

as follow, the partial gradient of w_1 would be

$$\sum_{i=1}^{N} y^i x_1^i = w_0 \sum_{i=1}^{N} x_0^i x_1^i + w_1 \sum_{i=1}^{N} x_1^i x_1^i$$

Therefore

$$\begin{bmatrix} \sum_{i=1}^{N} y^i x_0^i \\ \sum_{i=1}^{N} y^i x_1^i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_0^i x_0^i & \sum_{i=1}^{N} x_0^i x_1^i \\ \sum_{i=1}^{N} x_0^i x_1^i & \sum_{i=1}^{N} x_1^i x_1^i \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \tag{1}$$

Formally, it could conclude that

$$\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w}$$

$$\mathbf{w} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

Here still need to add the trace version(more generalized version)

Why the gradient could be equal to 0

Hessian matrix is a square matrix of second-order partial derivatives of scalar-valued function, or scalar field.

$$\mathbf{H}(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{2}$$

$$\mathbf{H} = \begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial x^2} \end{vmatrix}$$

In the 2 dimension, when $\mathbf{H} > 0$: if $\frac{\partial^2 f}{\partial x^2} > 0$, then point$(x_0, y_0)$ is the local min point. If $\frac{\partial^2 f}{\partial x^2} < 0$, then point$(x_0, y_0)$ is the local max point.

when $\mathbf{H} < 0$, then point(x_0, y_0) is the stationary point.

when $\mathbf{H} = 0$, second order cannot decide the point property, then consider it in higher order Taylor's Expansion.

In the example,$\mathbf{H} = 4(x_0^i)^2(x_1^i)^2 - 4(x_1^i x_0^i)(x_0^i x_1^i) = 0$, and $\frac{\partial^2 f}{\partial x^2} = 4(x_0^i)^2(x_1^i)^2 > 0$. Therefore, it is a local min point for the loss function.

In higher dimension space (multi-variables), $\mathbf{H}(f)$ should be a positive definite matrix($(\nabla\mathbf{x})^{\mathrm{T}}\mathbf{H}(f)\nabla\mathbf{x} \geq 0$ for any $\nabla\mathbf{x}$).

The more detail of Hessian matrix could look up Taylor expansion.

Generalized linear regression

$$L(\mathbf{w}) = \sum_{i=1}^{N} (y^i - \mathbf{w}^{\mathrm{T}} \phi(\mathrm{x}^i))^{\mathrm{T}}$$

where $\phi(\mathbf{x}^i)$ is a polynomial function for $\mathbf{x}^i$

normalization

L1 norm(euclidean) norm:

$$||\mathbf{w}||_2 = \sqrt{\sum_{d=1}^{D} w_d^2} = \sqrt{< \mathbf{w}, \mathbf{w} >}$$

L2 norm(manhattan) norm:

$$||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d|$$

Lp norm, $p > 1$:

$$||\mathbf{w}||_p = (\sum_{d=1}^{D} w_d^p)^{\frac{1}{p}}$$

Ridge regression: L2-regularized linear regression

$$\begin{aligned} L(\mathbf{w}) &= \epsilon^{\mathrm{T}} \epsilon + \lambda \mathbf{w}^{\mathrm{T}} \mathbf{w} = \mathbf{y}^{\mathrm{T}} \mathbf{y} - 2\mathbf{y}^{\mathrm{T}} \mathbf{X} \mathbf{w} + \mathbf{w}^{\mathrm{T}} (\mathbf{X}^{\mathrm{T}} \mathbf{X} + \lambda \mathbf{I}) \mathbf{w} \\ \nabla L(\mathbf{w}^*) &= 0 \\ \mathbf{w}^* &= (\mathbf{X}^{\mathrm{T}} \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^{\mathrm{T}} \mathbf{y} \end{aligned}$$

In some case, the matrix cannot be inversed, then add $\lambda \mathbf{I}$ to make it invertible..

invertible matrix

Lasso regression: L1-regularized linear regression

suitable for small sample, large dimension.

It could shrink some coefficient into 0, helpful for feature selection.

The optimization would be gradient descent, LARS, PGD.

Logistic regression

sigmoid function:

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

Given training set:$\{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)\}, \mathbf{x} \in \mathbb{R}^D, y \in \{0, 1\}$ The ML function would be:

$$\begin{aligned}
p(y^1, \ldots, y^n | \mathbf{x}^1, \ldots, \mathbf{x}^N) &= \prod_{i=1}^{N} P(y^i | \mathbf{x}^i) \\
&= \prod_{i=1}^{N} \sigma(\mathbf{w}^T \mathbf{x}^i)^{y^i} (1 - \sigma(\mathbf{w}^T \mathbf{x}^i))^{1-y^i} \\
logP(\mathbf{y} | \mathbf{X}; \mathbf{w}) &= \sum_{i=1}^{N} y^i log\sigma(\mathbf{w}^T \mathbf{x}^i) + (1 - y^i) log(1 - \sigma(\mathbf{w}^T \mathbf{x}^i))
\end{aligned}$$