Week 6
Logistic Regression and SVM

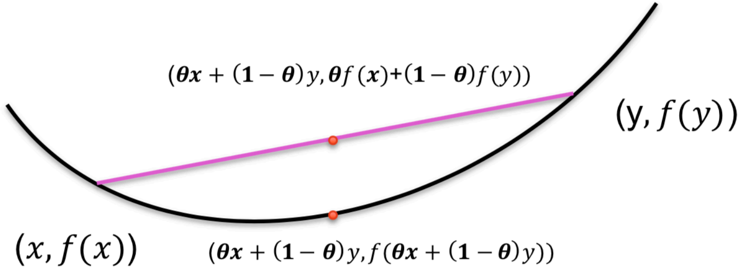
Overview

1. Gradient Descent Algorithm
2. Logistic Regression

Gradient Descent

For convex function, we can find global minimum, same as local minimum.
For non-convex function, we can find local minimum but global minimum is not guaranteed,

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$



The objective function is also known as loss, cost, fitness, utility, energy, etc. function.

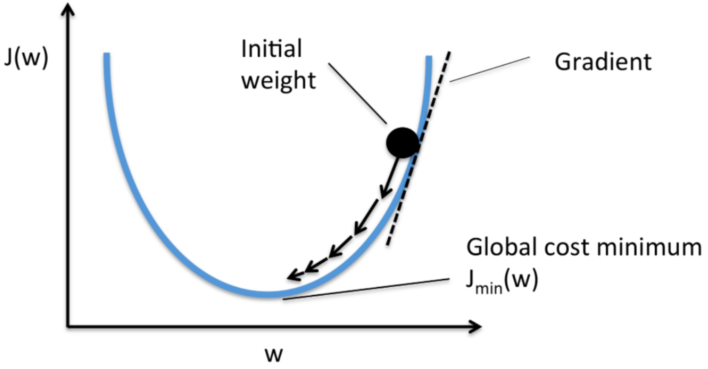
Objectives include: maximizes likelihood (ML), minimizes negative log-likelihood (NLL), maximizes a posterior (MAP).

e.g. The objective function of SVM is convex

$$f(x) = \frac{1}{2}\|x\|^2 + C\sum_{i=1}^{n} max\{0, 1 - b_i a_i^\top x\}$$

Optimization with constraints: lagrangian relaxation and KKT conditions.

Gradient Descent



Cost function: Sum of Squared Errors

$$J(w) = \frac{1}{2}\sum_i (y^{(i)} - \phi(z^{(i)}))^2$$

Step 1: Take partial derivative of the cost function with respect to each weight wj (gradient)

$$\frac{\partial J}{\partial w_j} = -\sum_i (y^{(i)} - \phi(z^{(i)}))x_j^{(i)}$$

Step 2: Update weights by taking a step away from the gradient

$$w := w + \Delta w$$

Step 3: The weight change is defined as the negative gradient multiplied by the learning rate

$$\Delta w = -\eta \Delta J(w)$$

Gradient Descent

$$\begin{aligned}
\frac{\partial J}{\partial w_j} &= \frac{\partial}{\partial w_j}\frac{1}{2}\sum_i (y^{(i)} - \phi(z^{(i)}))^2 \\
&= \frac{1}{2}\frac{\partial}{\partial w_j}\sum_i (y^{(i)} - \phi(z^{(i)}))^2 \\
&= \frac{1}{2}\sum_i 2(y^{(i)} - \phi(z^{(i)}))\frac{\partial}{\partial w_j}(y^{(i)} - \phi(z^{(i)})) \\
&= \sum_i (y^{(i)} - \phi(z^{(i)}))\frac{\partial}{\partial w_j}(y^{(i)} - \sum_i (w_j^{(i)} x_j^{(i)})) \\
&= \sum_i (y^{(i)} - \phi(z^{(i)}))(-x_j^{(i)}) \\
&= -\sum_i (y^{(i)} - \phi(z^{(i)}))x_j^{(i)}
\end{aligned}$$

Gradient Descent

Batch gradient descent:

Compute the gradient for the **entire** training dataset.

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta, \mathcal{X}^{(1:end)})$$

Stochastic gradient descent:

Compute the gradient for **each** training example.

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta, \mathcal{X}^{(i)})$$

Mini-batch gradient descent:

Compute the gradient for every **mini-batch** training examples.

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta, \mathcal{X}^{(i:i+n)})$$

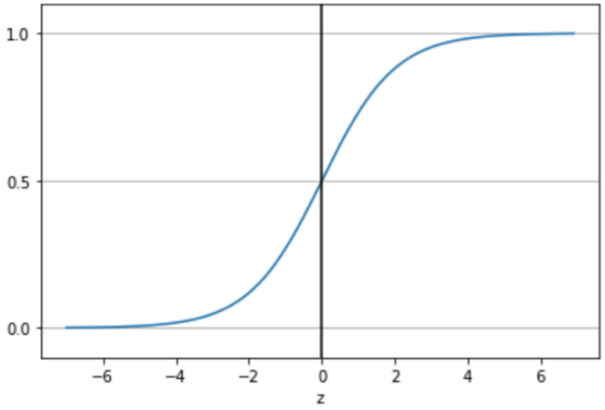
Gradient Descent Optimisation:

Adagrad, Adam, Momentum, Adadelta
Adamax, Nseterov, Rmsprop

Logistic Regression

Sigmoid Function

$$\sigma\left(f(x)\right) = \frac{1}{1+e^{-f(\boldsymbol{x})}} = \frac{e^{f(\boldsymbol{x})}}{1+e^{f(\boldsymbol{x})}}$$



Logistic: Binary classification

Bernoulli distribution pdf

$$f(\boldsymbol{x}) = p^x(1-p)^{1-x} = \begin{cases} p, & if\ x=1 \\ 1-p, & if\ x=0 \end{cases}$$

$p(y=1|\boldsymbol{x})$ Prob(target)

$p(y=0|\boldsymbol{x}) = 1 - p(y=1|\boldsymbol{x})$ Prob(non-target)

Logistic Regression

How to find optimal Beta? => Maximum Likelihood Estimation

1. Likelihood function $L(\boldsymbol{\beta}) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$

2. Take negative log-likelihood

$$\mathcal{L}(\boldsymbol{\beta}) = -logL(\boldsymbol{\beta}) = -log\left(\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}\right)$$

$$= -\sum_{i=1}^{n} log\left(p_i^{y_i}(1-p_i)^{1-y_i}\right)$$

$$= -\sum_{i=1}^{n}\left(log\left(p_i^{y_i}\right) + \log\left((1-p_i)^{1-y_i}\right)\right)$$

$$= -\sum_{i=1}^{n}\left(y_i log(p_i) + (1-y_i)\log(1-p_i)\right)$$

$$= -\sum_{i=1}^{n}\left(y_i log(p_i) + \log(1-p_i) - y_i \log(1-p_i)\right)$$

$$= -\sum_{i=1}^{n}\left(y_i log\left(\frac{p_i}{1-p_i}\right) + log(1-p_i)\right)$$

3. Simplify NLL

$$\mathcal{L}(\boldsymbol{\beta}) = -logL(\boldsymbol{\beta}) = -\sum_{i=1}^{n}\left(\underline{y_i log\left(\frac{p_i}{1-p_i}\right) + \log(1-p_i)}\right)$$

$$= -\sum_{i=1}^{n}\left(y_i \boldsymbol{\beta}^T \boldsymbol{x}_i - log\left(1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right)\right)$$

$$= \sum_{i=1}^{n}\left(log\left(1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right) - y_i \boldsymbol{\beta}^T \boldsymbol{x}_i\right)$$

$$p_i = \sigma\left(f(\boldsymbol{x}_i)\right) = \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}$$

$$log\left(\frac{p_i}{1-p_i}\right) = log\left(\frac{\frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}}{1-\frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}}\right) = log\left(e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right) = \boldsymbol{\beta}^T \boldsymbol{x}_i$$

$$log(1-p_i) = log\left(1-\frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}\right) = log\left(\frac{1}{1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}\right) = -log\left(1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right)$$

Logistic Regression

How to find optimal Beta? => Maximum Likelihood Estimation

4. Compute gradient with respect to Beta

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{\partial \sum_{i=1}^{n} \left(log\left(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right) - y_i \boldsymbol{\beta}^T \boldsymbol{x}_i \right)}{\partial \boldsymbol{\beta}}$$

$$= \sum_{i=1}^{n} \left\{ \frac{\partial \left(log\left(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right) \right)}{\partial \boldsymbol{\beta}} - \frac{\partial \left(y_i \boldsymbol{\beta}^T \boldsymbol{x}_i \right)}{\partial \boldsymbol{\beta}} \right\}$$

$$= \sum_{i=1}^{n} \left\{ \frac{1}{\left(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right)} \times \frac{\partial \left(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right)}{\partial \boldsymbol{\beta}} - y_i \boldsymbol{x}_i \right\}$$

$$= \sum_{i=1}^{n} \left\{ \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}}{\left(1 + e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\right)} \boldsymbol{x}_i - y_i \boldsymbol{x}_i \right\} = \sum_{i=1}^{n} \{ p_i \boldsymbol{x}_i - y_i \boldsymbol{x}_i \}$$

$$= \sum_{i=1}^{n} \{ p_i - y_i \} \boldsymbol{x}_i$$

5. Gradient descent and update Beta

$$\boldsymbol{\beta}^{(t+1)} \leftarrow \boldsymbol{\beta}^{(t)} + \alpha \frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)} + \alpha \sum_{i=1}^{n} \{ p_i - y_i \} \boldsymbol{x}_i$$