Statistics 101C - Binary Classification: Surrogate Loss functions

Shirong Xu

University of California, Los Angeles

shirong@stat.ucla.edu

October 10, 2023

Question

- **Question**: If you are asked to minimize or maximize a function $f : \mathbb{R} \to \mathbb{R}$, what would you do?

Question

- **Question**: If you are asked to minimize or maximize a function $f : \mathbb{R} \to \mathbb{R}$, what would you do?
- **Answer**: Take the derivative with respect to x and solve the equation

$$\frac{d}{dx} f(x) = 0$$

Question

- **Question**: If you are asked to minimize or maximize a function $f : \mathbb{R} \to \mathbb{R}$, what would you do?
- **Answer**: Take the derivative with respect to x and solve the equation

$$\frac{d}{dx} f(x) = 0$$

- **Question**: What is the equation $\frac{d}{dx} f(x) = 0$ is hard to solve?

Question

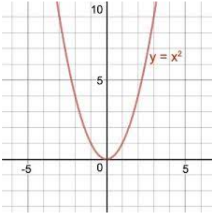
- **Question**: If you are asked to minimize or maximize a function $f : \mathbb{R} \to \mathbb{R}$, what would you do?
- **Answer**: Take the derivative with respect to x and solve the equation

$$\frac{d}{dx} f(x) = 0$$

- **Question**: What is the equation $\frac{d}{dx} f(x) = 0$ is hard to solve?
- **Answer**: Gradient ascent (For maximization) or Gradient descent (For minimization)

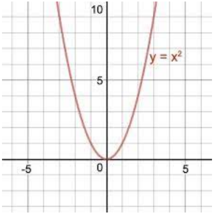
Gradient Descent: An Example

- Suppose we want to minimize $f(x) = x^2$



Gradient Descent: An Example

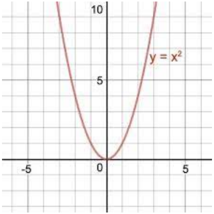
- Suppose we want to minimize $f(x) = x^2$



- First, we randomly choose an initial point $x^{(0)} = 3$

Gradient Descent: An Example

- Suppose we want to minimize $f(x) = x^2$

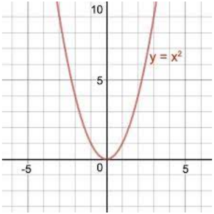


- First, we randomly choose an initial point $x^{(0)} = 3$
- Second, we calculate the derivative at the point $x^{(0)} = 3$

$$f'(3) = 6$$

Gradient Descent: An Example

- Suppose we want to minimize $f(x) = x^2$

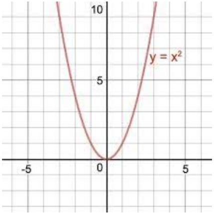


- First, we randomly choose an initial point $x^{(0)} = 3$
- Second, we calculate the derivative at the point $x^{(0)} = 3$

$$f'(3) = 6$$

- Update the new value $x^{(1)} = x^{(0)} - \lambda \cdot f'(3)$, λ is the step size.

- Suppose we want to minimize $f(x) = x^2$



- First, we randomly choose an initial point $x^{(0)} = 3$
- Second, we calculate the derivative at the point $x^{(0)} = 3$

$$f'(3) = 6$$

- Update the new value $x^{(1)} = x^{(0)} - \lambda \cdot f'(3)$, λ is the step size.
- Repeat the above process until convergence.

Gradient Descent

- **Motivation of Gradient descent**: gradient provides information to minimize the objective function.

- **General form of gradient descent**: Let $f : \mathbb{R}^p \to \mathbb{R}$ be a p-variate function. Then gradient descent has the form

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} - \lambda \nabla f(\boldsymbol{x}),$$

where $\nabla f(\boldsymbol{x}) = \left(\frac{\partial}{\partial x_1} f(\boldsymbol{x}), \frac{\partial}{\partial x_2} f(\boldsymbol{x}), \ldots, \frac{\partial}{\partial x_p} f(\boldsymbol{x}) \right)$

Gradient Descent

- **Motivation of Gradient descent**: gradient provides information to minimize the objective function.

- **General form of gradient descent**: Let $f : \mathbb{R}^p \to \mathbb{R}$ be a p-variate function. Then gradient descent has the form

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} - \lambda \nabla f(\boldsymbol{x}),$$

where $\nabla f(\boldsymbol{x}) = \left(\frac{\partial}{\partial x_1} f(\boldsymbol{x}), \frac{\partial}{\partial x_2} f(\boldsymbol{x}), \dots, \frac{\partial}{\partial x_p} f(\boldsymbol{x}) \right)$

- **Question**: When does gradient descent stop?

Gradient Descent

- **Motivation of Gradient descent**: gradient provides information to minimize the objective function.

- **General form of gradient descent**: Let $f : \mathbb{R}^p \to \mathbb{R}$ be a p-variate function. Then gradient descent has the form

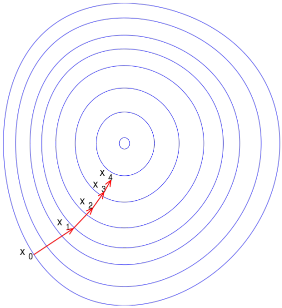
$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} - \lambda \nabla f(\boldsymbol{x}),$$

  where $\nabla f(\boldsymbol{x}) = \left(\frac{\partial}{\partial x_1} f(\boldsymbol{x}), \frac{\partial}{\partial x_2} f(\boldsymbol{x}), \ldots, \frac{\partial}{\partial x_p} f(\boldsymbol{x}) \right)$

- **Question**: When does gradient descent stop?
- **Answer**: When $\nabla f(\boldsymbol{x}) \approx \mathbf{0}$

Application of Gradient Descent

- Gradient descent is usually employed, when optimization problem is non-convex or does not have **analytic solution** or high-dimensional $\boldsymbol{x}$, for example Deep neural networks (Non-convex)



- Convex Optimization Problem + Gradient Descent $\Rightarrow$ Optimal Solution

Classification

- A typical dataset in classification $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$.
  - $\boldsymbol{x}_i$: the covariate vector of i-th instance
  - $y_i \in \{-1, 1\}$: binary label of i-th instance

Classification

- A typical dataset in classification $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$.
  - $\boldsymbol{x}_i$: the covariate vector of i-th instance
  - $y_i \in \{-1, 1\}$: binary label of i-th instance

- **Question**: Can we directly minimize the averaged 0-1 loss?

$$\text{Training Error} : \frac{1}{n} \sum_{i=1}^{n} I(f(\boldsymbol{x}_i) \neq y_i)$$

Classification

- A typical dataset in classification $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.
  - $\boldsymbol{x}_i$: the covariate vector of i-th instance
  - $y_i \in \{-1, 1\}$: binary label of i-th instance
- **Question**: Can we directly minimize the averaged 0-1 loss?

$$\text{Training Error} : \frac{1}{n}\sum_{i=1}^{n} I(f(\boldsymbol{x}_i) \neq y_i)$$

- **Answer**: No, the 0-1 loss function is non-convex and discontinuous, so (sub)gradient methods cannot be applied.

Classification - Surrogate Loss

- We can replace the 0-1 loss by other loss functions

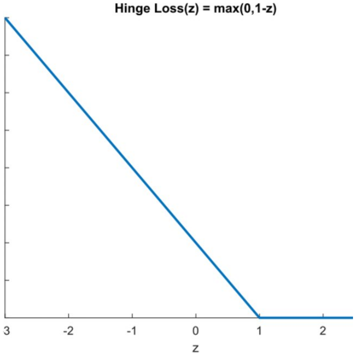
$$\frac{1}{n}\sum_{i=1}^{n} I(f(\boldsymbol{x}_i) \neq y_i) \Rightarrow \frac{1}{n}\sum_{i=1}^{n} L(f(\boldsymbol{x}_i), y_i) = \frac{1}{n}\sum_{i=1}^{n} \phi(f(\boldsymbol{x}_i)y_i)$$

  - Hinge loss: $\phi(x) = \max\{0, 1 - x\}$
  - Logistic loss $\phi(x) = \log(1 + \exp(-x))$

Classification - Hinge Loss

- Definition of Hinge loss:

$$L_{hinge}(f(\boldsymbol{x}_i), y_i) = \begin{cases} 1 - f(\boldsymbol{x}_i)y_i & \text{if } f(\boldsymbol{x}_i)y_i \leq 1 \\ 0, & \text{if } f(\boldsymbol{x}_i)y_i > 1 \end{cases}$$



Classification - Hinge Loss

- Let $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$. The expected hinge loss: hinge risk.

$$
\begin{aligned}
R_{hinge}(f) =& \mathbb{E}_{\boldsymbol{X},Y}\left[L_{hinge}(f(\boldsymbol{X}), Y)\right] \\
=& \mathbb{E}_{X}\Big[\eta(\boldsymbol{X})(1 - f(\boldsymbol{X}))_{+} + \big(1 - \eta(\boldsymbol{X})\big)(1 + f(\boldsymbol{X}))_{+}\Big]
\end{aligned}
$$

Classification - Hinge Loss

- Let $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1 | \boldsymbol{X} = \boldsymbol{x})$. The expected hinge loss: hinge risk.

$$\begin{aligned} R_{hinge}(f) =& \mathbb{E}_{\boldsymbol{X},Y}\left[L_{hinge}(f(\boldsymbol{X}), Y)\right] \\ =& \mathbb{E}_X\left[\eta(\boldsymbol{X})(1 - f(\boldsymbol{X}))_+ + \big(1 - \eta(\boldsymbol{X})\big)(1 + f(\boldsymbol{X}))_+\right] \end{aligned}$$

- Suppose that $f(\boldsymbol{X}) \in [-1, 1]$, for any $\boldsymbol{X}$, we have

$$\begin{aligned} & \eta(\boldsymbol{X})(1 - f(\boldsymbol{X})) + \big(1 - \eta(\boldsymbol{X})\big)(1 + f(\boldsymbol{X})) \\ =& \eta(\boldsymbol{X}) - 2\eta(\boldsymbol{X})f(\boldsymbol{X}) + 1 + f(\boldsymbol{X}) - \eta(\boldsymbol{X}) \\ =& f(\boldsymbol{X})\big(1 - 2\eta(\boldsymbol{X})\big) + 1. \end{aligned}$$

Classification - Hinge Loss

- Let $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$. The expected hinge loss: hinge risk.

$$\begin{aligned} R_{hinge}(f) =& \mathbb{E}_{\boldsymbol{X},Y}\left[L_{hinge}(f(\boldsymbol{X}), Y)\right] \\ =& \mathbb{E}_X\left[\eta(\boldsymbol{X})(1 - f(\boldsymbol{X}))_+ + \big(1 - \eta(\boldsymbol{X})\big)(1 + f(\boldsymbol{X}))_+\right] \end{aligned}$$

- Suppose that $f(\boldsymbol{X}) \in [-1, 1]$, for any $\boldsymbol{X}$, we have

$$\begin{aligned} & \eta(\boldsymbol{X})(1 - f(\boldsymbol{X})) + \big(1 - \eta(\boldsymbol{X})\big)(1 + f(\boldsymbol{X})) \\ =& \eta(\boldsymbol{X}) - 2\eta(\boldsymbol{X})f(\boldsymbol{X}) + 1 + f(\boldsymbol{X}) - \eta(\boldsymbol{X}) \\ =& f(\boldsymbol{X})\big(1 - 2\eta(\boldsymbol{X})\big) + 1. \end{aligned}$$

- The optimal function f^*_{hinge} minimizing $R_{hinge}(f)$
  - If $1 - 2\eta(\boldsymbol{X}) > 0$, hinge loss is minimized at $f(\boldsymbol{X}) = -1$
  - If $1 - 2\eta(\boldsymbol{X}) < 0$, hinge loss is minimized at $f(\boldsymbol{X}) = 1$

Classification - Hinge Loss

- Recall that the optimal classifier (Bayes classifier) of Binary loss is defined as

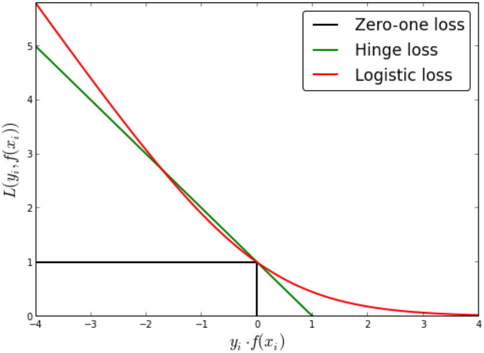
$$f^*(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2) = \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x}) > 1/2 \\ 0 & \text{if } \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x}) < 1/2 \end{cases}$$

- **Observation**: f^*_{hinge} is exactly the Bayes classifier and the hinge loss is a **convex function**, which makes it possible to minimize the training error.

Classification - Logistic Loss

- Definition of Logistic loss:

$$L_{log}(f(\boldsymbol{x}_i), y_i) = \log\left(1 + \exp(-f(\boldsymbol{x}_i)y_i)\right)$$



Classification - Logistic Loss

- The logistic risk:

$$
\begin{aligned}
R_{log}(f) &= \mathbb{E}_{\boldsymbol{X},Y}\left[\log\left(1+\exp(-f(\boldsymbol{X})Y)\right)\right] \\
=&\mathbb{E}_{\boldsymbol{X}}\left[\eta(\boldsymbol{X})\log\left(1+\exp(-f(\boldsymbol{X}))\right)+(1-\eta(\boldsymbol{X}))\log\left(1+\exp(f(\boldsymbol{X}))\right)\right]
\end{aligned}
$$

Classification - Logistic Loss

- The logistic risk:

$$
\begin{aligned}
R_{log}(f) &= \mathbb{E}_{\boldsymbol{X},Y}\Big[\log\Big(1+\exp(-f(\boldsymbol{X})Y)\Big)\Big] \\
=&\mathbb{E}_{\boldsymbol{X}}\Big[\eta(\boldsymbol{X})\log\Big(1+\exp(-f(\boldsymbol{X}))\Big) + (1-\eta(\boldsymbol{X}))\log\Big(1+\exp(f(\boldsymbol{X}))\Big)\Big]
\end{aligned}
$$

- Take the derivative with respect to f, it follows that

$$
\begin{aligned}
&-\eta(\boldsymbol{X})\frac{\exp(-f(\boldsymbol{X}))}{1+\exp(-f(\boldsymbol{X}))} + (1-\eta(\boldsymbol{X}))\frac{\exp(f(\boldsymbol{X}))}{1+\exp(f(\boldsymbol{X}))} \\
=&-\eta(\boldsymbol{X})\frac{1}{1+\exp(f(\boldsymbol{X}))} + (1-\eta(\boldsymbol{X}))\frac{\exp(f(\boldsymbol{X}))}{1+\exp(f(\boldsymbol{X}))} \\
=&\frac{\exp(f(\boldsymbol{X}))}{1+\exp(f(\boldsymbol{X}))} - \eta(\boldsymbol{X}) = 0 \longleftrightarrow f^*_{log}(\boldsymbol{X}) = \log\frac{\eta(\boldsymbol{X})}{1-\eta(\boldsymbol{X})}
\end{aligned}
$$

Connection between binary loss and surrogate losses

- The Bayes classifier $f^*(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$
- The optimal classifier of Hinge risk $f^*_{hinge}(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$
- The optimal classifier of Logistic risk $f^*_{log}(\boldsymbol{x}) = \log \frac{\eta(\boldsymbol{X})}{1-\eta(\boldsymbol{X})}$

Connection between binary loss and surrogate losses

- The Bayes classifier $f^*(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$
- The optimal classifier of Hinge risk $f^*_{hinge}(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$
- The optimal classifier of Logistic risk $f^*_{log}(\boldsymbol{x}) = \log \frac{\eta(\boldsymbol{X})}{1-\eta(\boldsymbol{X})}$
- **Question**: what is the connection between these optimal classifiers?

Connection between binary loss and surrogate losses

- The Bayes classifier $f^*(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$
- The optimal classifier of Hinge risk $f^*_{hinge}(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$
- The optimal classifier of Logistic risk $f^*_{log}(\boldsymbol{x}) = \log \frac{\eta(\boldsymbol{X})}{1-\eta(\boldsymbol{X})}$
- **Question**: what is the connection between these optimal classifiers?
- **Answer**: They are consistent in sign.

Classification

- A typical dataset in classification $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.
  - $\boldsymbol{x}_i$: the covariate vector of i-th instance
  - $y_i \in \{0,1\}$: binary label of i-th instance

- Bayes classifier f^*:

$$f^*(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x}) > 1/2 \\ 0 & \text{if } \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x}) < 1/2 \end{cases}$$

- Minimal risk $R(f^*)$:

$$R(f^*) = \mathbb{E}\Big[f^*(\boldsymbol{X}) \neq Y\Big] = \mathbb{E}\Big[\min(\eta(\boldsymbol{X}), 1 - \eta(\boldsymbol{X}))\Big],$$

where $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$.

How can we construct classifier?

- **Discriminative models**
  - Discriminative modeling studies the $P(Y|\boldsymbol{X})$
  - Examples: Logistic regression (LR) and Support Vector Machine (SVM)

How can we construct classifier?

- **Discriminative models**
  - Discriminative modeling studies the $P(Y|\boldsymbol{X})$
  - Examples: Logistic regression (LR) and Support Vector Machine (SVM)
- **Generative models**
  - Generative models studies the joint probability distribution $\mathbb{P}(\boldsymbol{X}, Y)$
  - Examples: linear discriminant analysis and quadratic discriminant analysis

Logistic Regression

- **Logistic regression** estimates the conditional probability probability:

$$\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X}\Big)$$

- In logistic regression, it is assumed that

$$\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big) = \frac{\exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x})}{1 + \exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x})},$$

where
  - $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ is a p-dimensional predictor
  - β_0 and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ are unknown parameters
  - $\boldsymbol{\beta}^T \boldsymbol{x} = \sum_{l=1}^{p} \beta_i x_i$

Why Logistic Regression

- The **odds ratio**: the probability that $Y = 1$ divided by the probability that $Y = 0$ conditional on $\boldsymbol{X} = \boldsymbol{x}$.

$$\exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}) = \frac{\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}{1 - \mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)} = \frac{\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}{\mathbb{P}\Big(Y = 0 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}$$

Why Logistic Regression

- The **odds ratio**: the probability that $Y = 1$ divided by the probability that $Y = 0$ conditional on $\boldsymbol{X} = \boldsymbol{x}$.

$$\exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}) = \frac{\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}{1 - \mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)} = \frac{\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}{\mathbb{P}\Big(Y = 0 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}$$

- The **log-odds**: linear with respect to $\boldsymbol{\beta}$

$$\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x} = \log\left(\frac{\mathbb{P}\Big(Y = 1 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}{\mathbb{P}\Big(Y = 0 \Big| \boldsymbol{X} = \boldsymbol{x}\Big)}\right)$$

  - The log-odds take any value in $\mathbb{R}$
  - The log-odds equals a linear combination of the predictors.
  - β_i can then be interpreted as the average change in the log-odds ratio given by a one-unit increase in x_i

Estimation in Logistic Regression

- Likelihood function $L(\beta_0, \boldsymbol{\beta})$:

$$L(\beta_0, \boldsymbol{\beta}) = \prod_{i=1}^{n} \Big(\mathbb{P}\big(Y = 1 \big| \boldsymbol{X} = \boldsymbol{x}\big)\Big)^{y_i} \Big(\mathbb{P}\big(Y = 0 \big| \boldsymbol{X} = \boldsymbol{x}\big)\Big)^{1-y_i}$$

Estimation in Logistic Regression

- Likelihood function $L(\beta_0, \boldsymbol{\beta})$:

$$L(\beta_0, \boldsymbol{\beta}) = \prod_{i=1}^{n} \left(\mathbb{P}(Y = 1 \middle| \boldsymbol{X} = \boldsymbol{x})\right)^{y_i} \left(\mathbb{P}(Y = 0 \middle| \boldsymbol{X} = \boldsymbol{x})\right)^{1-y_i}$$

- Logarithm of $L(\beta_0, \boldsymbol{\beta})$:

$$\begin{aligned}\log L(\beta_0, \boldsymbol{\beta}) &= \sum_{i=1}^{n} \left[y_i \log\left(\mathbb{P}(Y = 1 \middle| \boldsymbol{X} = \boldsymbol{x})\right) + (1 - y_i) \log\left(\mathbb{P}(Y = 0 \middle| \boldsymbol{X} = \boldsymbol{x})\right)\right] \\ &= \sum_{i=1}^{n} \left[y_i(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}) - \log\left(1 + \exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x})\right)\right]\end{aligned}$$

Estimation in Logistic Regression

Estimate β_0 and $\boldsymbol{\beta}$ (Gradient Ascent):

$$\beta_0^{(t+1)} \leftarrow \beta_0^{(t)} + \lambda \sum_{i=1}^{n} \left[y_i - \frac{\exp(\beta_0^{(t)} + \boldsymbol{\beta}^{(t)T}\boldsymbol{x})}{1 + \exp(\beta_0^{(t)} + \boldsymbol{\beta}^{(t)}\boldsymbol{x})} \right]$$

$$\boldsymbol{\beta}^{(t+1)} \leftarrow \boldsymbol{\beta}^{(t)} + \lambda \sum_{i=1}^{n} \left[y_i - \frac{\exp(\beta_0^{(t)} + \boldsymbol{\beta}^{(t)T}\boldsymbol{x})}{1 + \exp(\beta_0^{(t)} + \boldsymbol{\beta}^{(t)T}\boldsymbol{x})} \right] \boldsymbol{x}_i$$

Classification

Once we get estimated parameters, we have

$$\widehat{\eta}(\boldsymbol{x}) = \frac{\exp(\widehat{\beta}_0 + \widehat{\boldsymbol{\beta}}^T \boldsymbol{x})}{1 + \exp(\widehat{\beta}_0 + \widehat{\boldsymbol{\beta}}^T \boldsymbol{x})}$$

Then we make predictions by

$$\widehat{f}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \widehat{\eta}(\boldsymbol{x}) > 1/2 \\ 0, & \text{if } \widehat{\eta}(\boldsymbol{x}) < 1/2 \end{cases}$$

If $\widehat{\eta}(\boldsymbol{x}) = 1/2$, then just randomly assign a label to it.

Example: Data Generation

Dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{5}000$, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$.

- Features are generated from uniform distribution $x_{il} \sim Unif(0,2), l = 1,2,3,4$.
- $\beta_0 = 0.5$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_2)$ with $\beta_i \sim Unif(-1,1)$ for $i = 1,2,3,4$.
- Model:

$$y_i \sim Bernoulli(\frac{\exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x})}{1 + \exp(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x})})$$

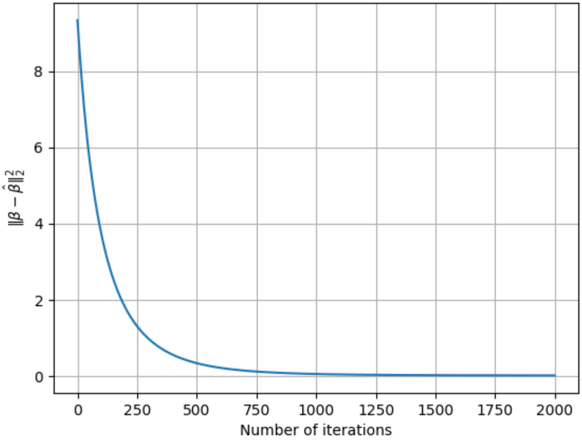
Python Codes

```
import numpy as np
np.random.seed(2)
n,p = 5000,4  # Set training datasize and dimension of features
X = np.random.uniform(-1,1,[n,p]) # Generation of features
beta = np.random.uniform(0,2,4) # Generation of parameters
beta_0 = 0.5 # Set the intercept term to 0.5
logOdd = (X * beta).sum(axis=1)+beta_0 # Log-odds
Prob = np.exp(logOdd)/(1+np.exp(logOdd)) # Probability
Y = np.array(Prob - np.random.uniform(0,1,n)>0,dtype=int) # Generate labels
```

Python Codes

```python
Beta_0_hat = 0. # Initialization of intercept term
Beta_hat = np.zeros(p) # Initialization of beta
lamb = 0.1 # Learning rate
Error = [] # Error set
for i in range(2000):# Iterations of gradient ascent
    logOdd_hat = (X * Beta_hat).sum(axis=1)+Beta_0_hat
    Beta_0_hat = Beta_0_hat + lamb * np.mean(Y - np.exp(logOdd_hat)/(1+np.exp(logOdd_hat)))
    Beta_hat = Beta_hat + lamb * ((Y - np.exp(logOdd_hat)/(1+np.exp(logOdd_hat))) * X.T).mean(axis=1)
    Error.append(np.linalg.norm(Beta_hat-beta)**2)
import matplotlib.pyplot as plt
plt.plot(np.arange(0,2000),Error)
plt.xlabel('Number of iterations')
plt.ylabel('$\Vert \\beta - \hat{\\beta}\Vert_2^2$')
plt.grid()
```

Example: gradient ascent for logistic regression



Application of Logistic Regression to Banknote dataset

```
library(mclust)
data(banknote)
set.seed(123)
i <- 1:dim(banknote)[1]
i.train <- sample(i, 130, replace = FALSE)
bn.train <- banknote[i.train,]
bn.test <- banknote[-i.train,]
ml <- glm(Status Length + Right + Left + Top, data =
bn.train,family = "binomial")
summary(ml)
pred.log.odds <- predict(ml)
pred.probs <- predict(ml, type = 'response')
my.thres <- 0.5
pred.log.odds.test <- predict(ml, bn.test[,-1])
pred.probs.test <- predict(ml, bn.test[,-1], type = 'response')
predicted.counterfeit <- pred.probs.test > my.thres
```

- table('Reference' = bn.test[,1] == 'counterfeit', "Predicted" = predicted.counterfeit)

```
            Predicted
Reference FALSE TRUE
    FALSE     2   31
    TRUE     31    6
```