Lecture 5: Logistic Regression

CIS 4190/5190
Fall 2022
Supervised Learning

Data $Z = \{(x_i, y_i)\}_{i=1}^n$

$\hat{\beta}(Z) = \arg\min_{\beta} L(\beta; Z)$

$L$ encodes $y_i \approx f_{\beta}(x_i)$

Model $f_{\hat{\beta}(Z)}$
Regression

Data $Z = \{(x_i, y_i)\}_{i=1}^n$

Label is a real value $y_i \in \mathbb{R}$

Model $f_{\hat{\beta}(Z)}$

$\hat{\beta}(Z) = \arg\min_{\beta} L(\beta; Z)$

$L$ encodes $y_i \approx f_{\beta}(x_i)$
Classification

Data $Z = \{(x_i, y_i)\}_{i=1}^n$

$\hat{\beta}(Z) = \arg \min_\beta L(\beta; Z)$

$L$ encodes $y_i \approx f_\beta(x_i)$

Model $f_{\hat{\beta}(Z)}$

Label is a **discrete value** $y_i \in Y = \{c_1, \ldots, c_k\}$
(Binary) Classification

- **Input:** Dataset $Z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
- **Output:** Model $y_i \approx f_\beta(x_i)$

Example: Malignant vs. Benign Ocular Tumor

Image: https://eyecancer.com/uncategorized/choroidal-metastasis-test/
Loss Minimization View of ML

• Three design decisions
  • Model family: What are the candidate models $f$? (E.g., linear functions)
  • Loss function: How to define “approximating”? (E.g., MSE loss)
  • Optimizer: How do we optimize the loss? (E.g., gradient descent)

• How do we adapt to classification?
Linear Functions for (Binary) Classification

• **Input:** Dataset \( Z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \)

• **Regression:**
  • Labels \( y_i \in \mathbb{R} \)
  • Predict \( y_i \approx \beta^T x_i \)

• **Classification:**
  • Labels \( y_i \in \{0, 1\} \)
  • Predict \( y_i \approx 1(\beta^T x_i \geq 0) \)
  • \( 1(C) \) equals 1 if \( C \) is true and 0 if \( C \) is false
  • How to learn \( \beta \)? **Need a loss function!**
Loss Functions for Linear Classifiers

• (In)accuracy:

\[ L(\beta; Z) = \frac{1}{n} \sum_{i=1}^{n} 1(y_i \neq f_{\beta}(x_i)) \]

• Computationally intractable
• Often, but not always the “true” loss (e.g., imbalanced data)

\[ L(\beta; Z) = \frac{6}{50} \]
Loss Functions for Linear Classifiers

• Distance:

\[ L(\beta; Z) = \frac{1}{n} \sum_{i=1}^{n} \text{dist}(x_i, f_\beta) \cdot 1(f_\beta(x_i) \neq y_i) \]

• If \( L(\beta; Z) = 0 \), then 100% accuracy
• Variant of this loss results in SVM
• But, we will consider a more general strategy

\[ L(\beta; Z) = 1.2 \]
Maximum Likelihood Estimation

• Our first **probabilistic** viewpoint on learning (from statistics)

• Given $x_i$, suppose $y_i$ is drawn i.i.d. from distribution $p_{Y|X}(Y = y \mid x; \beta)$ with parameters $\beta$ (or density, if $y_i$ is continuous):

\[
y_i \sim p_{Y|X}(\cdot \mid x_i; \beta)
\]

• Typically write $p_{\beta}(Y = y \mid x)$ or just $p_{\beta}(y \mid x)$
  • Called a **model** (and $\{p_{\beta}\}_\beta$ is the **model family**)
  • Will show up convert $p_{\beta}$ to $f_{\beta}$ later

$Y$ is random variable, not vector
Maximum Likelihood Estimation

• **Likelihood**: Given model $p_\beta$, the probability of dataset $Z$ :

$$L(\beta; Z) = p_\beta(y_i \mid X) = \prod_{i=1}^{n} p_\beta(y_i \mid x_i)$$

• **Negative Log-likelihood (NLL)**: Negative log of the likelihood function (computationally better behaved):

$$\ell(\beta; Z) = -\log L(\beta; Z) = -\sum_{i=1}^{n} \log p_\beta(y_i \mid x_i)$$
Intuition on the Likelihood

High likelihood
(Low NLL)

Low likelihood
(High NLL)
**Example:** Linear Regression

- Assume that the conditional density is

  $$p_\beta(y_i \mid x_i) = N(y_i; \beta^\top x_i, 1) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{\|\beta^\top x_i - y_i\|^2}{2}}$$

- $N(\mu, \sigma^2)$ is the density of the normal (a.k.a. Gaussian) distribution with mean $\mu$ and variance $\sigma^2$

- Note that $p_\beta(y_i \mid x_i)$ is maximized if $y_i = \beta^\top x_i$
Example: Linear Regression

• Then, the likelihood is

\[
L(\beta; Z) = \prod_{i=1}^{n} p_\beta(y_i \mid x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{(\beta^\top x_i - y_i)^2}{2}}
\]

• The NLL is

\[
\ell(\beta; Z) = -\sum_{i=1}^{n} \log p_\beta(y_i \mid x_i) = \frac{n \log(2\pi)}{2} + \sum_{i=1}^{n} (\beta^\top x_i - y_i)^2
\]

\[\text{constant} \quad \text{MSE!}\]
Example: Linear Regression

• Loss minimization for maximum likelihood estimation:

\[ \hat{\beta}(Z) = \arg \min_{\beta} \ell(\beta; Z) \]

• Note: Maximizing the likelihood equivalent to minimizing the NLL
Example: Linear Regression

• What about the model family?

\[
f_\beta(x) = \underset{y}{\arg\max} \ p_\beta(y \mid x)
\]

\[
= \arg\max_y \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{\|\beta^T x - y\|^2}{2}}
\]

\[
= \beta^T x
\]

• Recovers linear functions!
Loss Minimization View of ML

• Three design decisions
  • **Model family**: What are the candidate models $f$? (E.g., linear functions)
  • **Loss function**: How to define “approximating”? (E.g., MSE loss)
  • **Optimizer**: How do we optimize the loss? (E.g., gradient descent)
Maximum Likelihood View of ML

- **Two** design decisions
  - **Likelihood:** Probability \( p_\beta(y \mid x) \) of data \((x, y)\) given parameters \( \beta \)
  - **Optimizer:** How do we optimize the NLL? (E.g., gradient descent)

- **Corresponding Loss Minimization View:**
  - **Model family:** Most likely label \( f_\beta(x) = \arg \max_y p_\beta(y \mid x) \)
  - **Loss function:** Negative log likelihood (NLL) \( \ell(\beta; Z) = -\sum_{i=1}^n \log p_\beta(y_i \mid x_i) \)
What about classification?

- Consider the following choice:

  $$p_\beta(Y = 0 \mid x_i) \propto e^{-\frac{\beta^T x_i}{2}} \quad \text{and} \quad p_\beta(Y = 1 \mid x_i) \propto e^{\frac{\beta^T x_i}{2}}$$

- Then, we have

  $$p_\beta(Y = 1 \mid x_i) = \frac{e^{\frac{\beta^T x_i}{2}}}{\frac{\beta^T x_i}{2} + e^{-\frac{\beta^T x_i}{2}}} = \frac{1}{1 + e^{-\beta^T x_i}}$$

- Furthermore, $$p_\beta(Y = 0 \mid x_i) = 1 - p_\beta(Y = 1 \mid x_i)$$
What about classification?

• Consider the following choice:

\[ p_\beta(Y = 0 \mid x_i) \propto e^{-\frac{\beta^T x_i}{2}} \quad \text{and} \quad p_\beta(Y = 1 \mid x_i) \propto e^{\frac{\beta^T x_i}{2}} \]

• Then, we have

\[ p_\beta(Y = 1 \mid x_i) = \frac{e^{\frac{\beta^T x_i}{2}}}{e^{\frac{\beta^T x_i}{2}} + e^{-\frac{\beta^T x_i}{2}}} = \sigma(\beta^T x_i) \]

• Furthermore, \( p_\beta(Y = 0 \mid x_i) = 1 - \sigma(\beta^T x_i) \)

Sigmoid function

\[ \sigma(z) = \frac{1}{1 + e^{-z}} \]
Logistic/Sigmoid Function

\[ p_\beta(Y = 1 \mid x_i) = \sigma(\beta^T x_i) \]
Logistic Regression Model Family

\[ f_\beta(x) = \arg \max_y p_\beta(y \mid x_i) \]

\[ = \arg \max_y \begin{cases} 
\sigma(\beta^T x_i) & \text{if } y = 1 \\
1 - \sigma(\beta^T x_i) & \text{if } y = 0 
\end{cases} \]

\[ = \begin{cases} 
1 & \text{if } \sigma(\beta^T x_i) \geq \frac{1}{2} \\
0 & \text{otherwise} 
\end{cases} \]
Logistic Regression Model Family

\[ f_\beta(x) = \arg \max_y p_\beta(y \mid x_i) \]

= \arg \max_y \begin{cases} 
\sigma(\beta^T x_i) & \text{if } y = 1 \\
1 - \sigma(\beta^T x_i) & \text{if } y = 0 
\end{cases}

= \begin{cases} 
1 & \text{if } \sigma(\beta^T x_i) \geq \frac{1}{2} \\
0 & \text{otherwise} 
\end{cases}

= \begin{cases} 
1 & \text{if } \beta^T x_i \geq 0 \\
0 & \text{otherwise} 
\end{cases}

= 1(\beta^T x_i \geq 0)

• Recovers linear classifiers!
Logistic Regression Algorithm

• Then, we have the following NLL loss:

\[
\ell(\beta; Z) = - \sum_{i=1}^{n} \log p_\beta(y_i | x_i) \\
= - \sum_{i=1}^{n} 1(y_i = 1) \cdot \log(\sigma(\beta^T x_i)) + 1(y_i = 0) \cdot \log(1 - \sigma(\beta^T x_i)) \\
= - \sum_{i=1}^{n} y_i \cdot \log(\sigma(\beta^T x_i)) + (1 - y_i) \cdot \log(1 - \sigma(\beta^T x_i))
\]

• Logistic regression minimizes this loss:

\[
\hat{\beta}(Z) = \arg \min_{\beta} \ell(\beta; Z)
\]
Intuition on the Objective

• Loss for example $i$ is

$$\begin{cases} 
- \log(\sigma(\beta^\top x_i)) & \text{if } y_i = 1 \\
- \log(1 - \sigma(\beta^\top x_i)) & \text{if } y_i = 0 
\end{cases}$$
Intuition on the Objective

• Loss for example $i$ is

\[
\begin{cases}
-\log(\sigma(\beta^T x_i)) & \text{if } y_i = 1 \\
-\log(1 - \sigma(\beta^T x_i)) & \text{if } y_i = 0
\end{cases}
\]
Intuition on the Objective

• If $y_i = 1$:
  • If $p_\beta(Y = 1 \mid x_i) = 1$, then loss $= 0$
  • As $p_\beta(Y = 1 \mid x_i) \to 0$, loss $\to \infty$

\[
y_i \cdot \log(\sigma(\beta^T x_i)) + (1 - y_i) \cdot \log(1 - \sigma(\beta^T x_i))
\]
Intuition on the Objective

• If $y_i = 1$:
  • If $p_\beta(Y = 1 \mid x_i) = 1$, then loss = 0
  • As $p_\beta(Y = 1 \mid x_i) \to 0$, loss $\to \infty$

• If $y_i = 0$:
  • If $p_\beta(Y = 0 \mid x_i) = 1$, then loss = 0
  • As $p_\beta(Y = 0 \mid x_i) \to 0$, loss $\to \infty$

$$y_i \cdot \log(\sigma(\beta^T x_i)) + (1 - y_i) \cdot \log(1 - \sigma(\beta^T x_i))$$
Optimization for Logistic Regression

• To optimize the NLL loss, we need its gradient:

\[
\nabla_{\beta} \ell(\beta; Z) = -\sum_{i=1}^{n} y_i \cdot \nabla_{\beta} \log(\sigma(\beta^T x_i)) + (1 - y_i) \cdot \nabla_{\beta} \log(1 - \sigma(\beta^T x_i))
\]

\[
= -\sum_{i=1}^{n} y_i \cdot \frac{\nabla_{\beta} \sigma(\beta^T x_i)}{\sigma(\beta^T x_i)} - (1 - y_i) \cdot \frac{\nabla_{\beta} \sigma(\beta^T x_i)}{1 - \sigma(\beta^T x_i)}
\]

\[
\sigma'(z) = \sigma(z)(1 - \sigma(z))
\]

\[
= -\sum_{i=1}^{n} y_i \cdot \frac{\sigma(\beta^T x_i)(1 - \sigma(\beta^T x_i))}{\sigma(\beta^T x_i)} \cdot x_i - (1 - y_i) \cdot \frac{\sigma(\beta^T x_i)(1 - \sigma(\beta^T x_i))}{1 - \sigma(\beta^T x_i)} \cdot x_i
\]

\[
= -\sum_{i=1}^{n} y_i - \sigma(\beta^T x_i)) \cdot x_i - (1 - y_i) \cdot \sigma(\beta^T x_i) \cdot x_i
\]

\[
= -\sum_{i=1}^{n} (y_i - \sigma(\beta^T x_i)) \cdot x_i
\]
Optimization for Logistic Regression

• Gradient of NLL:

\[ \nabla_\beta \ell(\beta; Z) = \sum_{i=1}^{n} (\sigma(\beta^T x_i) - y_i)x_i \]

• Surprisingly similar to the gradient for linear regression!
  • Only difference is the \( \sigma \)

• Gradient descent works as before
  • No closed-form solution for \( \hat{\beta}(Z) \)
Feature Maps

• Can use feature maps, just like linear regression
Multi-Class Classification

• What about more than two classes?
  • Disease diagnosis: healthy, cold, flu, pneumonia
  • Object classification: desk, chair, monitor, bookcase
  • In general, consider a finite space of labels $\mathcal{Y}$
Multi-Class Classification

• **Naïve Strategy:** One-vs-rest classification
  • **Step 1:** Train $|Y|$ logistic regression models, where model $p_{\beta_y}(Y = 1 \mid x)$ is interpreted as the probability that the label for $x$ is $y$
  • **Step 2:** Given a new input $x$, predict label $y = \arg \max_{y'} p_{\beta_{y'}}(Y = 1 \mid x)$
Multi-Class Logistic Regression

• **Strategy:** Include separate $\beta_y$ for each potential label $y \in \mathcal{Y}$

• Let $p_\beta(y \mid x) \propto e^{-\beta_y^T x}$, i.e.

$$p_\beta(y \mid x) = \frac{e^{-\beta_y^T x}}{\sum_{y' \in \mathcal{Y}} e^{-\beta_{y'}^T x}}$$

• We define $\text{softmax}(z_1, \ldots, z_k) = \begin{bmatrix} \frac{e^{-z_1}}{\sum_{i=1}^k e^{-z_i}} & \cdots & \frac{e^{-z_k}}{\sum_{i=1}^k e^{-z_i}} \end{bmatrix}$

• Then, $p_\beta(y \mid x) = \text{softmax}(\beta_{y_1}^T x, \ldots, \beta_{y_k}^T x)_y$

  • Thus, sometimes called **softmax regression**
Multi-Class Logistic Regression

• Model family
  
  \[ f_\beta(x) = \arg\max_y p_\beta(y \mid x) = \arg\max_y \frac{e^{-\beta^\top \tilde{x}_y}}{\sum_{y' \in y} e^{-\beta_{y'}^\top \tilde{x}}} = \arg\min_y \beta_y^\top \tilde{x} \]

• Optimization
  
  • Gradient descent on NLL
  • Simultaneously update all parameters \( \{\beta_y\}_{y \in Y} \)
Regularized Logistic Regression

• We can add $L_1$ or $L_2$ regularization to the NLL loss, e.g.:

$$\ell(\beta; Z) = - \sum_{i=1}^{n} y_i \cdot \log(\sigma(\beta^T x_i)) + (1 - y_i) \cdot \log(1 - \sigma(\beta^T x_i)) + \lambda \cdot \|\beta\|_2^2$$

• Is there a more “natural” way to derive the regularized loss?
Regularization as a Prior

- So far, we have not assumed any distribution over the parameters $\beta$
  - What if we assume $\beta \sim N(0, \sigma^2 I)$ (the $d$ dimensional normal distribution)?

- Consider the modified likelihood

\[
L(\beta; Z) = p_{Y_{\beta|X}}(Y, \beta | X) = p_{Y|X,\beta}(Y | X, \beta) \cdot N(\beta; 0, \sigma^2 I)
\]

\[
= \left( \prod_{i=1}^{n} p_\beta(y_i | x_i) \right) \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|\beta\|^2}{2\sigma^2}}
\]
Regularization as a Prior

- So far, we have not assumed any distribution over the parameters $\beta$
  - What if we assume $\beta \sim N(0, \sigma^2 I)$ (the $d$ dimensional normal distribution)?

- Consider the modified NLL

$$
\ell(\beta; Z) = -\sum_{i=1}^{n} \log p_\beta(y_i | x_i) + \log \sigma \sqrt{2\pi} + \frac{\|\beta\|_2^2}{2\sigma^2}
$$

- Obtain $L_2$ regularization on $\beta$!
  - With $\lambda = \frac{1}{2\sigma^2}$
  - If $\beta_i \sim \text{Laplace}(0, \sigma^2)$ for each $i$, obtain $L_1$ regularization
Additional Role of Regularization

• For logistic regression, regularization is necessary

• In $p_{\beta}$, if we replace $\beta$ with $c \cdot \beta$, where $c \gg 1$ (and $c \in \mathbb{R}$), then:
  • The decision boundary does not change
  • The probabilities $p_{\beta}(y \mid x)$ become more confident

\[
\begin{align*}
  p_{\beta}(y \mid x) & \approx 0.6 \\
  p_{\beta}(Y = 1 \mid x) & \approx 0.6 \\
  p_{10\beta}(y \mid x) & \\
  p_{10\beta}(Y = 1 \mid x) & \approx 1
\end{align*}
\]
Additional Role of Regularization

- For logistic regression, regularization is **necessary**

- Regularization ensures that $\beta$ does not become too large
  - Without regularization (i.e., $\lambda = 0$), gradient descent diverges (i.e., $\beta \rightarrow \pm \infty$)
  - Also prevents overconfidence