Lecture 7
Logistic Regression

Luigi Freda
ALCOR Lab
DIAG
University of Rome "La Sapienza"

January 28, 2018
Outline

1. Intro
   - Logistic Regression
   - Decision Boundary

2. Maximum Likelihood Estimation
   - Negative Log-Likelihood

3. Optimization Algorithms
   - Gradient Descent
   - Newton’s Method
   - Iteratively Reweighted Least Squares (IRLS)

4. Regularized Logistic Regression
   - Concept
Outline

1 Intro
   - Logistic Regression
     - Decision Boundary

2 Maximum Likelihood Estimation
   - Negative Log-Likelihood

3 Optimization Algorithms
   - Gradient Descent
   - Newton’s Method
   - Iteratively Reweighted Least Squares (IRLS)

4 Regularized Logistic Regression
   - Concept
Linear Regression

linear regression

- $y \in \mathbb{R}$, $x \in \mathbb{R}^D$ and $w \in \mathbb{R}^D$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$y(x) = w^T x + \epsilon = \sum_{j=1}^{D} w_j x_j + \epsilon$$

$$p(y|x, \theta) = \mathcal{N}(w^T x, \sigma^2)$$

polynomial regression

- we replace $x$ by a non-linear function $\phi(x) \in \mathbb{R}^{d+1}$

$$y(x) = w^T \phi(x) + \epsilon$$

$$p(y|x, \theta) = \mathcal{N}(w^T \phi(x), \sigma^2)$$

- $\mu(x) = w^T \phi(x)$ (basis function expansion)
- $\phi(x) = [1, x, x^2, ..., x^d]$ is the vector of polynomial basis functions

N.B.: in both cases $\theta = (w, \sigma^2)$ are the model parameters
Logistic Regression
From Linear to Logistic Regression

Can we generalize linear regression ($y \in \mathbb{R}$) to **binary classification** ($y \in \{0, 1\}$)?

We can follow two steps:

1. Replace $y \sim \mathcal{N}(\mu(x), \sigma^2(x))$ with $y \sim \text{Ber}(y|\mu(x))$ (we want $y \in \{0, 1\}$)
2. Replace $\mu(x) = w^T x$ with $\mu(x) = \text{sigm}(w^T x)$ (we want $0 \leq \mu(x) \leq 1$)

Where

- $\text{Ber}(y|\mu(x)) = \mu(x)^{(y=1)}(1 - \mu(x))^{(y=0)}$ is the **Bernoulli distribution**
- $\mathbb{I}(e) = 1$ if $e$ is true, $\mathbb{I}(e) = 0$ otherwise (**indicator function**)
- $\text{sigm}(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)} = \frac{1}{1+\exp(-\eta)}$ is the **sigmoid function** (aka logistic function)
following the two steps:

1. replace \( y \sim \mathcal{N}(\mu(x), \sigma^2(x)) \) with \( y \sim \text{Ber}(y|\mu(x)) \) (we want \( y \in \{0, 1\} \))

2. replace \( \mu(x) = w^T x \) with \( \mu(x) = \text{sigm}(w^T x) \) (we want \( 0 \leq \mu(x) \leq 1 \))

we start from a linear regression

\[
p(y|x, \theta) = \mathcal{N}(w^T x, \sigma^2) \quad \text{where} \quad y \in \mathbb{R}
\]

to obtain a logistic regression

\[
p(y|x, w) = \text{Ber}(y|\text{sigm}(w^T x)) \quad \text{where} \quad y \in \{0, 1\}
\]
1 Intro
   1. Logistic Regression
   2. Decision Boundary

2 Maximum Likelihood Estimation
   1. Negative Log-Likelihood

3 Optimization Algorithms
   1. Gradient Descent
   2. Newton’s Method
   3. Iteratively Reweighted Least Squares (IRLS)

4 Regularized Logistic Regression
   1. Concept
Logistic Regression
Linear Decision Boundary

\[ p(y|x, w) = \text{Ber}(y|\text{sigm}(w^Tx)) \quad \text{where } y \in \{0, 1\} \]

- \[ p(y = 1|x, w) = \text{sigm}(w^Tx) = \frac{\exp(w^Tx)}{1+\exp(w^Tx)} = \frac{1}{1+\exp(-w^Tx)} \]
- \[ p(y = 0|x, w) = 1 - p(y = 1|x, w) = 1 - \text{sigm}(w^Tx) = \text{sigm}(-w^Tx) \]
- \[ p(y = 1|x, w) = p(y = 0|x, w) = 0.5 \text{ entails} \]
  \[ \text{sigm}(w^Tx) = 0.5 \implies w^Tx = 0 \]
- hence we have a **linear decision boundary** \( w^Tx = 0 \)
Logistic Regression

Linear Decision Boundary

- **Linear decision boundary** $\mathbf{w}^T \mathbf{x} = 0$ (hyperplane passing through the origin)
- indeed, as in the linear regression case $\mathbf{w}^T \mathbf{x} = [w_0, \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}]^T$ where $\mathbf{x} = [1, \tilde{x}]^T$ and $\tilde{x}_i$ are the actual data samples
- as a matter of fact, our linear decision boundary has the form $\mathbf{w}^T \tilde{\mathbf{x}} + w_0 = 0$

- **Hyperplane** $\mathbf{a}^T \mathbf{x} + b = 0$ equivalent to $\mathbf{n}^T \mathbf{x} - d = 0$ where $\mathbf{n}$ is the normal unit vector (i.e. $\|\mathbf{n}\| = 1$) and $d \in \mathbb{R}$ is the distance origin-hyperplane
- one can define $\mathbf{x}_0 \triangleq \mathbf{n}d$ and rewrite the plane equation as $\mathbf{n}^T (\mathbf{x} - \mathbf{x}_0) = 0$
we can replace $x$ by a non-linear function $\phi(x)$ and obtain a

$$p(y|x, w) = \text{Ber}(y | \text{sigm}(w^T \phi(x)))$$

if $x \in \mathbb{R}$ we can use $\phi(x) = [1, x, x^2, \ldots, x^d]$ which is the vector of polynomial basis functions

in principle, if $x \in \mathbb{R}^D$ we could use a multivariate polynomial expansion

$$w^T \phi(x) = \sum w_{i_1 i_2 \ldots i_D} \prod_{j=1}^D x_j^{i_j}$$

up to a certain degree $d$

$$p(y = 1|x, w) = \text{sigm}(w^T \phi(x))$$

$$p(y = 0|x, w) = \text{sigm}(-w^T \phi(x))$$

$p(y = 1|x, w) = p(y = 0|x, w) = 0.5$ entails

$$\text{sigm}(w^T \phi(x)) = 0.5 \implies w^T \phi(x) = 0$$

hence we have a non-linear decision boundary $w^T \phi(x) = 0$
solid black dots are data \((x_i, y_i)\)
open red circles are **predicted probabilities**: \(p(y = 1|x, w) = \text{sigm}(w_0 + w_1x)\)
in this case data is **not** linearly separable
the linear decision boundary is \(w_0 + w_1x = 0\) which entails \(x = -w_0/w_1\)
in general, when data is not linearly separable, we can try to use the basis function expansion as a further step
Logistic Regression

A 2D Example

- **left**: a linear decision boundary on the "feature plane" \((x_1, x_2)\)
- **right**: a 3D plot of \(p(y = 1|x, w) = \text{sigm}(w_0 + w_1x_2 + w_2x_2)\)
Logistic Regression

Examples

- left: non-linearly separable data with a linear decision boundary
- right: the same dataset fit with a quadratic model (and quadratic decision boundary)
another example of non-linearly separable data which is fit by using a polynomial model
1 Intro
   - Logistic Regression
   - Decision Boundary

2 Maximum Likelihood Estimation
   - Negative Log-Likelihood

3 Optimization Algorithms
   - Gradient Descent
   - Newton’s Method
   - Iteratively Reweighted Least Squares (IRLS)

4 Regularized Logistic Regression
   - Concept
the likelihood for the logistic regression is given by

\[ p(D|\theta) = \prod_i p(y_i|x_i, \theta) = \prod_i \text{Ber}(y_i|\mu_i) = \prod_i \mu_i^{y_i=1}(1 - \mu_i)^{y_i=0} \]

where \( \mu_i \triangleq \text{sigm}(w^T x_i) \)

the Negative Log-Likelihood (NLL) is given by

\[ NLL = -\log p(D|\theta) = \sum_i \left[ \mathbb{I}(y_i=1) \log \mu_i + \mathbb{I}(y_i=0) \log(1 - \mu_i) \right] = \sum_i \left[ y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i) \right] \]
we have

\[ NLL = \sum_i \left[ y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i) \right] \]

where \( \mu_i \triangleq \text{sigm}(w^T x_i) \)

in order to find the MLE we have to minimize the NLL and impose \( \frac{\partial NLL}{\partial w_i} = 0 \)

given \( \sigma(a) \triangleq \text{sigm}(a) = \frac{1}{1 + e^{-a}} \) it is possible to show (homework ex 8.3) that

\[ \frac{d\sigma(a)}{da} = \sigma(a)(1 - \sigma(a)) \]

using the previous equation and the chain rule for calculus we can compute the gradient \( \mathbf{g} \)

\[ \mathbf{g} \triangleq \frac{d}{d\mathbf{w}} NLL(\mathbf{w}) = \sum_i \frac{\partial NLL}{\partial \mu_i} \frac{d\mu_i}{da_i} \frac{da_i}{d\mathbf{w}} = \sum_i (\mu_i - y_i)x_i \]

where \( \mu_i = \sigma(a_i) \) and \( a_i \triangleq \mathbf{w}^T \mathbf{x}_i \)
the gradient can be rewritten as

\[ g = \sum_i (\mu_i - y_i)x_i = X^T(\mu - y) \]

where \( X \) is the design matrix, \( \mu \triangleq [\mu_1, ..., \mu_N]^T \), \( y \triangleq [y_1, ..., y_N]^T \) and \( \mu_i \triangleq \text{sigm}(w^T x_i) \)

the Hessian is

\[ H \triangleq \frac{d}{dw} g(w)^T = \sum_i \left( \frac{d\mu_i}{da_i} \frac{da_i}{dw} \right) x_i^T = \sum_i \mu_i(1 - \mu_i)x_i x_i^T = X^T S X \]

where \( S \triangleq \text{diag}(\mu_i(1 - \mu_i)) \)

it is easy to see that \( H > 0 \) \( (v^T H v = (v^T X^T) S (X v) = z^T S z > 0 ) \)

given that \( H > 0 \) we have that the NLL is convex and has a unique global minimum

unlike linear regression, there is no closed form for the MLE (since the gradient contains non-linear functions)

we need to use an optimization algorithm to compute the MLE
Outline

1. Intro
   - Logistic Regression
   - Decision Boundary

2. Maximum Likelihood Estimation
   - Negative Log-Likelihood

3. Optimization Algorithms
   - Gradient Descent
   - Newton’s Method
   - Iteratively Reweighted Least Squares (IRLS)

4. Regularized Logistic Regression
   - Concept
Gradient Descent

The Gradient

given a continuously differentiable function $f(\theta) \in \mathbb{R}$ we can use first order Taylor’s expansion an approximate

$$f(\theta) \approx f(\theta^*) + g(\theta^*)^T(\theta - \theta^*)$$

where the gradient $g$ is defined as

$$g(\theta) \triangleq \frac{\partial f}{\partial \theta} = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \vdots \\ \frac{\partial f}{\partial \theta_m} \end{bmatrix}$$

hence, in a neighbourhood of $\theta^*$ one has

$$\Delta f \approx g^T \Delta \theta$$

it is easy to see that with $\|\Delta \theta\| = \eta$

1. $\Delta f$ is max when $\Delta \theta = +\eta \frac{g}{\|g\|}$
2. $\Delta f$ is min when $\Delta \theta = -\eta \frac{g}{\|g\|}$ (steepest descent)

where $\hat{g} \triangleq \frac{g}{\|g\|}$ is the unit vector in the gradient direction
the simplest algorithm for unconstrained optimization is gradient descent (aka steepest descent)

$$\theta_{k+1} = \theta_k - \eta g_k$$

where $\eta \in \mathbb{R}^+$ is the step size (or learning rate) and $g_k \triangleq g(\theta_k)$

starting from an initial guess $\theta_0$, at each step $k$ we move towards the negative gradient direction $-g_k$
Gradient Descent

problem: how to choose the step size $\eta$?

- left: using a fixed step size $\eta = 0.1$
- right: using a fixed step size $\eta = 0.6$

- if we use constant step size and we make it too small, convergence will be very slow, but if we make it too large, the method can fail to convergence at all
Gradient Descent

Line Search

- **convergence to the global optimum**: the method is guaranteed to converge to the global optimum $\theta^*$ no matter where we start
- **global convergence**: the method is guaranteed to converge to a local optimum no matter where we start
- let's develop a more stable method for picking $\eta$ so as to have global convergence
- consider a general update
  \[ \theta_{k+1} = \theta_k + \eta d_k \]
  where $\eta > 0$ and $d_k$ are respectively our step size and selected descent direction
- by Taylor’s theorem, we have
  \[ f(\theta_k + \eta d_k) \approx f(\theta_k) + \eta g_k^T d_k \]
- if $\eta$ is chosen small enough and $d_k = -g_k$, then $f(\theta_k + \eta d_k) < f(\theta_k)$ (since $\Delta f \approx -\eta g_k^T g < 0$)
- but we don’t want to choose the step size $\eta$ too small, or we will move very slowly and may not reach the minimum
- **line minimization** of line search: pick $\eta$ so as to minimize
  \[ \phi(\eta) \triangleq f(\theta_k + \eta d_k) \]
in order to minimize

$$\phi(\eta) \triangleq f(\theta_k + \eta d_k)$$

we must impose

$$\frac{d\phi}{d\eta} = \left. \frac{\partial f}{\partial \theta} \right|_{\theta_k + \eta d_k}^T d_k = g(\theta_k + \eta d_k)^T d_k = 0$$

since in the gradient descent method we have $d_k = g_k$, the following condition must be satisfied

$$g(\theta_k + \eta d_k)^T g_k = 0$$
from the following condition

\[ \mathbf{g}(\theta_k + \eta \mathbf{d}_k)^T \mathbf{g}_k = 0 \]

we have that consecutive descent directions are orthogonal and we have a zig-zag behaviour.
Outline

1 Intro
   - Logistic Regression
   - Decision Boundary

2 Maximum Likelihood Estimation
   - Negative Log-Likelihood

3 Optimization Algorithms
   - Gradient Descent
   - Newton’s Method
   - Iteratively Reweighted Least Squares (IRLS)

4 Regularized Logistic Regression
   - Concept
given a twice-continuously differentiable function $f(\theta) \in \mathbb{R}$ we can use a second order Taylor’s expansion to approximate

$$f(\theta) \approx f(\theta^*) + g(\theta^*)^T(\theta - \theta^*) + \frac{1}{2}(\theta - \theta^*)^T H(\theta^*)(\theta - \theta^*)$$

the **Hessian matrix** $H = \frac{\partial^2 f(\theta)}{\partial \theta^2}$ of a function $f(\theta) \in \mathbb{R}$ is defined as follows (element-wise)

$$H_{ij} = \frac{\partial^2 f(\theta)}{\partial \theta_i \partial \theta_j}$$
Newton’s Method

- hence if we consider an optimization algorithm, at step $k$ we have

$$f(\theta) \approx f_{quad}(\theta) \triangleq f(\theta_k) + g_k^T(\theta - \theta_k) + \frac{1}{2}(\theta - \theta_k)^T H_k(\theta - \theta_k)$$

- in order to find $\theta_{k+1}$ we can then minimize $f_{quad}(\theta)$

$$f_{quad}(\theta) = \theta^T A \theta + b^T \theta + c$$

where

$$A = \frac{1}{2} H_k, \quad b = g_k - H_k \theta_k, \quad c = f_k - g_k^T \theta_k + \frac{1}{2} \theta_k^T H_k \theta_k$$

- we can then impose

$$\frac{\partial f_{quad}}{\partial \theta} = 0 \quad \Rightarrow \quad 2A \theta + b = 0 \quad \Rightarrow \quad H_k \theta + g_k - H_k \theta_k = 0$$

- the minimum of $f_{quad}$ is then

$$\theta = \theta_k - H_k^{-1} g_k$$

- in the Newton’s method one selects $d_k = -H_k^{-1} g_k$
Newton’s Method

- in the Newton’s method one selects $d_k = -H_k^{-1}g_k$
- the step $d_k = -H_k^{-1}g_k$ is what should be added to $\theta_k$ to minimize the second order approximation of $f$ around $\theta_k$
- in its simplest form, Newton’s method requires that $H_k > 0$ (the function is strictly convex)
- if not, the objective function is not convex, then $H_k$ may not be positive definite, so $d_k = -H_k^{-1}g_k$ may not be a descent direction
Newton’s Method

Algorithm 8.1: Newton’s method for minimizing a strictly convex function

1. Initialize $\theta_0$;
2. for $k = 1, 2, \ldots$ until convergence do
   3. Evaluate $g_k = \nabla f(\theta_k)$;
   4. Evaluate $H_k = \nabla^2 f(\theta_k)$;
   5. Solve $H_k d_k = -g_k$ for $d_k$;
   6. Use line search to find stepsize $\eta_k$ along $d_k$;
   7. $\theta_{k+1} = \theta_k + \eta_k d_k$;
Outline

1. Intro
   - Logistic Regression
   - Decision Boundary

2. Maximum Likelihood Estimation
   - Negative Log-Likelihood

3. Optimization Algorithms
   - Gradient Descent
   - Newton’s Method

4. Regularized Logistic Regression
   - Concept
Iteratively Reweighted Least Squares

let us now apply Newton’s algorithm to find the MLE for binary logistic regression

the Newton update at iteration $k + 1$ for this model is as follows (using $\eta_k = 1$, since the Hessian is exact)

$$w_{k+1} = w_k - H_k^{-1} g_k$$

since

$$g_k = X^T (\mu_k - y), \quad H_k = X^T S_k X$$

we have

$$w_{k+1} = w_k + (X^T S_k X)^{-1} X^T (y - \mu_k) =$$

$$= (X^T S_k X)^{-1} [(X^T S_k X)w_k + X^T (y - \mu_k)] = (X^T S_k X)^{-1} X^T (S_k X w_k + y - \mu_k)$$

then we have

$$w_{k+1} = (X^T S_k X)^{-1} X^T S_k z_k$$

where $z_k \triangleq X w_k + S_k^{-1} (y - \mu_k)$
the following equation

\[
w_{k+1} = (X^T S_k X)^{-1} X^T S_k z_k
\]

with \( z_k \triangleq Xw_k + S_k^{-1}(y - \mu_k) \) is an example of weighted least squares problem, which is a minimizer of

\[
J = \sum_{i=1}^{N} s_{ki} (z_{ki} - w^T x_i)^2 = \|z_k - Xw_k\|_{S_k^{-1}}
\]

where \( S_k = \text{diag}(s_{ki}) \), \( z_k = [z_{k1}, \ldots, z_{kN}]^T \)

since \( S_k \) is a diagonal matrix we can write the element-wise update

\[
z_{ki} = w_k^T x_i + \frac{y_i - \mu_{ki}}{\mu_{ki}(1 - \mu_{ki})}
\]

where \( \mu_k = [\mu_{k1}, \ldots, \mu_{kN}]^T \)

this algorithm is called iteratively reweighted least squares (IRLS)
Algorithm 8.2: Iteratively reweighted least squares (IRLS)

1. \( w = 0_D; \)
2. \( w_0 = \log(\bar{y}/(1 - \bar{y})); \)
3. repeat
   4. \( \eta_i = w_0 + w^T x_i; \)
   5. \( \mu_i = \text{sigm}(\eta_i); \)
   6. \( s_i = \mu_i(1 - \mu_i); \)
   7. \( z_i = \eta_i + \frac{y_i - \mu_i}{s_i}; \)
   8. \( S = \text{diag}(s_{1:N}); \)
   9. \( w = (X^T S X)^{-1} X^T S z; \)
4. until converged;
Outline

1. Intro
   - Logistic Regression
   - Decision Boundary

2. Maximum Likelihood Estimation
   - Negative Log-Likelihood

3. Optimization Algorithms
   - Gradient Descent
   - Newton’s Method
   - Iteratively Reweighted Least Squares (IRLS)

4. Regularized Logistic Regression
   - Concept
consider the **linearly separable** 2D data in the above figure

there are different decision boundaries that can perfectly separate the training data (4 examples are shown in different colors)

the likelihood surface is shown: it is unbounded as we move up and to the right in parameter space, along a ridge where $w_2/w_1 = 2.35$ (the indicated diagonal line)
we can maximize the likelihood by driving $\|w\|$ to infinity (subject to being on this line), since large regression weights make the sigmoid function very steep, turning it into an infinitely steep sigmoid function $\mathbb{I}(w^T x > w_0)$

consequently the **MLE is not well defined** when the data is linearly separable
to prevent this, we can move to \textbf{MAP estimation} and hence add a \textbf{regularization component} in the classification setting (as we did in the ridge regression)

to regularize the problem we can simply add spherical prior at the origin $p(w) = \mathcal{N}(0, \lambda I)$ and then maximize the posterior $p(w|D) \propto p(D|w)p(w)$
as a consequence a simple $l_2$ regularization can be easily obtained by using the following new objective, gradient and Hessian

$$f'(w) = \text{NLL}(w) + \lambda w^T w$$
$$g'(w) = g(w) + 2\lambda w$$
$$H'(w) = H(w) + 2\lambda I$$

these modified equations can be used into any of the presented optimizers
• Kevin Murphy’s book