# Lecture 8 - logistic regression & optimization

Thursday, September 28, 2023 2:28 PM

today: · logistic regression
· numerical optimization

## logistic regression:

setup: binary classification $Y = \{0,1\}$ $X \in \mathbb{R}^d$

generative model motivation:

suppose only assumption is there exists pdf's (densities) in $\mathbb{R}^d$ for each class conditionals:

$$p(x \mid Y=1) \quad \& \quad p(x \mid Y=0)$$

$$p(Y=1 \mid X=x) = \frac{p(Y=1, X=x)}{\{p(Y=1, X=x) + p(Y=0, X=x)\}\, p(X=x)}$$

$$= \frac{1}{1 + \frac{p(Y=0, X=x)}{p(Y=1, X=x)}} = \frac{1}{1+\exp(-f(x))}$$

where $f(x) \triangleq \log \frac{p(X=x \mid Y=1)}{p(X=x \mid Y=0)} + \log \frac{p(Y=1)}{p(Y=0)}$

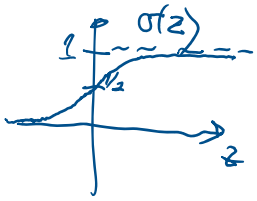
"log odd"

class-conditional ratio

prior odd ratio

in general $p(Y=1 \mid X=x) = \sigma(f(x))$ where $\sigma(z) \triangleq \frac{1}{1+\exp(-z)}$

"sigmoid function"




some properties of $\sigma(z)$: $\sigma(-z) = 1 - \sigma(z)$ $\quad [\sigma(z) + \sigma(-z) = 1]$

$$\frac{d}{dz}\sigma(z) = \sigma(z)\sigma(-z) = \sigma(z)(1-\sigma(z))$$

(*) to motivate linear logistic regression

consider class conditionals in the exponential family

$$p(x \mid \eta) \triangleq h(x) \exp(\eta^T T(x) - A(\eta))$$

- $\eta$: "canonical parameter"
- $T(x)$: "sufficient statistics"
- $A(\eta)$: scalar fct. → log partition fct. → normalize
- these specify the "flat" exponential family

Gaussian:

Gaussian:

$$\log p(x|\mu,\sigma^2) = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{(x-\mu)^2}{2\sigma^2}$$

$$= -\left(\frac{x^2}{2\sigma^2} - \frac{x\mu}{\sigma^2} + \frac{\mu^2}{2\sigma^2}\right)$$

$$p(x|\mu,\sigma^2) \propto \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$T(x) = \begin{bmatrix} -x^2/2 \\ x \end{bmatrix}$$

$$\eta(\mu,\sigma^2) = \begin{bmatrix} 1/\sigma^2 \\ \mu/\sigma^2 \end{bmatrix}$$

$$A(\eta) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{\mu^2}{2\sigma^2}$$

$$p(x|Y=1) = p(x|\eta_1)$$

$$p(x|Y=0) = p(x|\eta_0)$$

$$\text{log odds } f(x) = \log \underbrace{\frac{p(x|Y=1)}{p(x|Y=0)}}_{p(x|\eta_1)/p(x|\eta_0)} + \log \underbrace{\frac{p(Y=1)}{p(Y=0)}}_{\pi/(1-\pi)}$$

$$= (\eta_1-\eta_0)^T T(x) + A(\eta_0) - A(\eta_1) + \log\left(\frac{\pi}{1-\pi}\right)$$

$$\triangleq w^T \varphi(x)$$

where $w = \begin{pmatrix} \eta_1-\eta_0 \\ A(\eta_0)-A(\eta_1)+\log\frac{\pi}{1-\pi} \end{pmatrix}$ $\quad \varphi(x) = \begin{pmatrix} T(x) \\ 1 \end{pmatrix}$

get logistic regression model?

$$\boxed{p_w(Y=1|X=x) = \sigma(w^T\varphi(x))}$$

"feature map"

decision boundary

$$\mathbb{1}\{w^T\varphi(x) > 0\}$$

exercise to reader (hwk 2): try argument above with $p(x|y) = N(x|\mu_y, \Sigma_y)$

- if $\Sigma_0 = \Sigma_1$, then $\varphi(x) = \begin{pmatrix} x \\ 1 \end{pmatrix}$ → "linear boundary"
- otherwise, $\varphi(x) = \begin{pmatrix} -xx^T \\ x \\ 1 \end{pmatrix}$ (optional) → quadratic decision boundary

linear logistic regression model

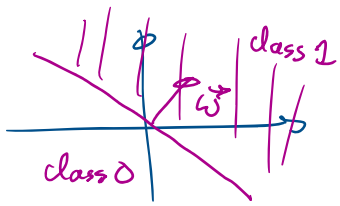
## linear logistic regression model

$p(y=1|x,w) = \sigma(w^T x)$ $\quad y=\{0,1\}$ $\quad$ decision rule: $\mathbb{1}\{w^T x > 0\}$

$p(y=0|x,w) = 1-\sigma(w^T x) = \sigma(-w^T x)$

[if $y=\{\pm 1\}$ "Rademacher R.V."]

encode $p(y|x) = \sigma(y w^T x)$




$Y|X=x$ is Bernoulli$(\sigma(w^T x))$

$p(y|x) = \sigma(w^T x)^y (1-\sigma(w^T x))^{1-y}$

given $(x_i, y_i)_{i=1}^n$ iid

### max-conditional log-likelihood to estimate $\hat{w}_{MCL}$

$$\ell(w) = \sum_{i=1}^n \log p(y_i|x_i,w) = \sum_{i=1}^n y_i \log \sigma(w^T x_i) + (1-y_i)\log(\sigma(-w^T x_i))$$

$\nabla_w \sigma(w^T x) = x\,[\sigma(w^T x)\sigma(-w^T x)]$ $\quad$ let $v_i \triangleq w^T x_i$

$$\nabla_w \ell(w) = \sum_{i=1}^n x_i \left[\sigma(v_i)\sigma(-v_i)\left[\frac{y_i}{\sigma(v_i)} - \frac{(1-y_i)}{\sigma(-v_i)}\right]\right]$$

$$= \sum_{i=1}^n x_i \left[y_i[\sigma(-v_i)+\sigma(v_i)] - \sigma(v_i)\right]$$

$$\boxed{\nabla \ell(w) = \sum_{i=1}^n x_i [y_i - \sigma(w^T x_i)]}$$

need to use numerical methods

solve for $\nabla\ell(w)=0 \Rightarrow$ need to solve transcendental equation

because $\sum_i \frac{1}{1+\exp(-w^T x_i)}(\dots) = 0$ ← solve for this

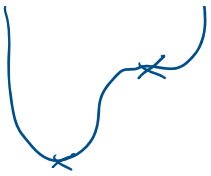
contrast to linear regression $\quad \nabla\ell(w) = \sum_{i=1}^n x_i [y_i - w^T x_i]$ ← linear in $w$

## numerical optimization

want to minimize $f(w)$ s.t. $w \in \mathbb{R}^d$ (unconstrained) [suppose $f$ is differentiable]

1) gradient descent (1st order method)

1) gradient descent (1st order method)



start $w_0$

iterate : $w_{t+1} = w_t - \gamma_t \nabla f(w_t)$

stopping criterion : if $\|\nabla f(w_t)\|^2 < \delta$ e.g. $\delta = 10^{-6}$

then stop

note : if $f$ is $\mu$-strongly convex $\Rightarrow f(w_t) - \min_w f(w) \leq \frac{1}{2\mu}\|\nabla f(w_t)\|^2$

($\Leftrightarrow f(w) - \frac{\mu}{2}\|w\|^2$ is convex in $w$)

if $\|\nabla f(w_t)\|^2 \leq \delta \Rightarrow$ subopt.$(w_t) \leq \frac{\delta}{2\mu}$

step-size rules

a) constant step-size : $\gamma_t = \frac{1}{L}$ ← Lipschitz continuity constant for $\nabla f$

i.e. $\|\nabla f(w) - \nabla f(w')\|_2 \leq L\|w - w'\|_2$

$\forall w, w' \in \mathbb{R}^d$

b) decreasing step-size rule [This is more common for stochastic optimization]

$\gamma_t = \underbrace{\quad}_{\text{constant}}$

usually want $\sum_t \gamma_t = +\infty$

$\sum_t \gamma_t^2 < \infty$

e.g. $f(w) \triangleq \mathbb{E}_\xi\, g(w, \xi)$

$w_{t+1} = w_t - \gamma_t \nabla_w g(w, \xi_t)$

e.g. $g(w, \xi) = \ell(y_i, h_w(x_i))$ for ERM

$\xi = (x_i, y_i)$

c) choose $\gamma_t$ by "line search" : $\min_{\gamma \in \mathbb{R}} f(w_t + \gamma d_t)$

costly in general ← direction of update (e.g. $-\nabla f(w_t)$)

↳ instead do approximate search

e.g. Armijo line search

[see Boyd's book]

15h38

Newton's method (2nd order method)

motive : minimizing a quadratic approximation

$\triangleq$ Hessian $[H(w_t)]_{ij} = \frac{\partial^2 f(w_t)}{\partial w_i \partial w_j}$

Taylor expansion $f(w) \approx f(w_t) + \nabla f(w_t)^T (w - w_t) + \frac{1}{2}(w - w_t)^T H(w_t)(w - w_t)$

Taylor expansion at $w_t$

$$f(w) = f(w_t) + \nabla f(w_t)^T (w - w_t) + \frac{1}{2}(w - w_t) H(w_t)(w - w_t) + O(\|w - w_t\|^3)$$

$$\triangleq Q_t(w)$$

$$= Q_t(w) + O(\|w - w_t\|^3)$$

$Q_t(w)$: quadratic model approximation

$O(\|w - w_t\|^3)$: Taylor's remainder

$w_{t+1}$ ↝ obtain by minimizing $Q_t(w)$

$$\nabla_w Q_t(w) \overset{want}{=} 0$$

$$\nabla f(w_t) + H(w_t)(w - w_t) \overset{want}{=} 0$$

$$\Rightarrow w - w_t = H^{-1}(w_t) \nabla f(w_t)$$

$$\boxed{w_{t+1} = w_t - H^{-1}(w_t) \nabla f(w_t)}$$

Newton's update

$H^{-1}$: inverse Hessian ↝ $O(d^3)$ time and $O(d^2)$ space

$d_t = H^{-1} \nabla f$

$H d_t = \nabla f$

Note for hwk 2: solve for $H_t d_t = \nabla f(w_t)$ for $d_t$

$$\min_d \|H_t d - \nabla f(w_t)\|^2$$

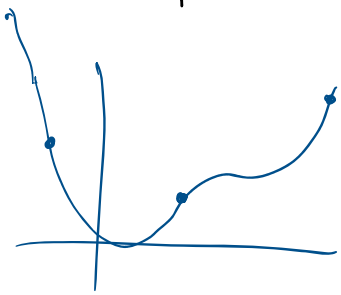
use `numpy.linalg.lstsq` to solve this

↝ much more numerically stable

Damped Newton: you add a step-size to stabilize Newton's method

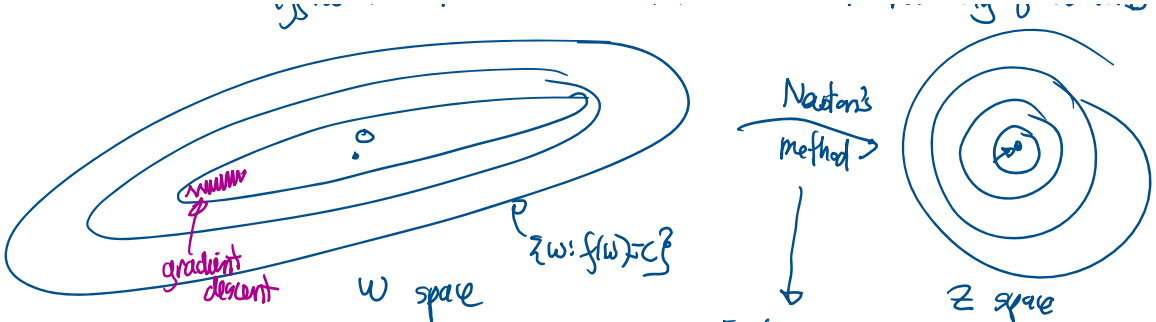
$$w_{t+1} = w_t - \gamma_t H^{-1}(w_t) \nabla f(w_t)$$

$\gamma_t$: step-size



why Newton's method?

- much faster convergence in # iterations vs gradient descent
- affine invariant ⇒ method is invariant to rescaling of variables




is using Hessian to make f "well conditioned"

$\frac{1}{2} w^T H w = c$, $H = P^T \Sigma P$ ($H$ is symmetric psd)

$\frac{1}{2} w^T P^T \Sigma P w = c \iff \frac{1}{2} z^T z = c$

diagonal; $z = \Sigma^{1/2} P w$

exercise to the reader: $z_{t+1} = z_t - \gamma \nabla \tilde{f}(z_t)$

$\iff w_{t+1} = w_t - \gamma H^{-1} \nabla f(w_t)$

## Big data logistic regression

- big d $\Rightarrow$ cannot do $O(d^2)$ or $O(d^3)$ operations $\Rightarrow$ first order methods
- if n is large, you cannot use batch methods $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w)$ $O(nd)$

gradient of loss on some data pt

(full) batch gradient

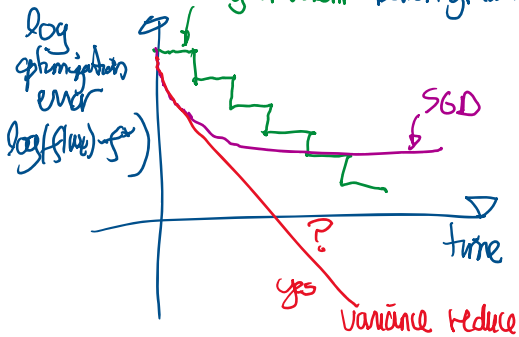
instead you can use "incremental gradient method"

e.g. stochastic gradient descent (SGD): $w_{t+1} = w_t - \gamma_t \nabla f_{i_t}(w_t)$

where $i_t$ is picked unif. at random

SGD → cheap updates, but slower convergence per iteration

batch grad. descent: batch gradient → expensive update, but faster convergence




→ SAG: stochastic average gradient

GD. $w_{t+1} = w_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w_t)$

SAG [2012] $w_{t+1} = w_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} v_i^t$ & memory

yes

variance reduced methods

SAG [2012] $w_{t+1} = w_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} v_i$ & memory

where $v_i = \nabla f_i(w_{old,i})$

at each $t$, update $v_{it} = \nabla f_i(w_t)$

SAGA [2014] $w_{t+1} = w_t - \gamma_t \left( \nabla f_{it}(w_t) + \frac{1}{n} \sum_j v_j - v_{it} \right)$

SAG

$\nabla f_{it}(w_t)$

variance reduction correction

(default method for large scale logistic regression in Scikit Learn)

SVRG

**Skipped part: Newton's method on logistic regression = IRLS:**

## Newton's method for logistic regression: IRLS

recall for $\ell(w)$: $\nabla \ell(w) = \sum_{i=1}^{n} x_i [y_i - \sigma(w^T x_i)]$

$H(\ell(w)) = -\sum_{i=1}^{n} x_i x_i^T \sigma(w^T x_i) \sigma(-w^T x_i)$

$v^T H v = -\sum_{i=1}^{n} \underbrace{(v^T x_i)(x_i^T v)}_{(x_i^T v)^2 \geq 0} \underbrace{\sigma(-) \sigma(-)}_{\geq 0}$ $\quad v^T H v \leq 0 \ \forall v \in \mathbb{R}^d$

i.e. $H \preceq 0$

i.e. concave fct.

Newton is maximizing instead of minimizing

Notation: recall

$X_{n \times d} = \begin{pmatrix} - x_1^T - \\ \vdots \end{pmatrix}$

let $\mu_i \triangleq \sigma(w^T x_i) \in ]0,1[$

$\mu_t \triangleq \begin{pmatrix} \vdots \\ \sigma(w^T x_i) \\ \vdots \end{pmatrix}$

$\nabla \ell(w) = \sum_i x_i [y_i - \mu_i] = X^T (y - \mu)$

Hessian $= -\sum_i x_i x_i^T \mu_i (1 - \mu_i) = -X^T D(w) X$

$D_{ii} \triangleq \mu_i (1 - \mu_i)$

diagonal matrix

Newton's update: $w_{t+1} = w_t - (-X^T D_t X)^{-1} X^T (y - \mu_t)$

$= (X^T D_t X)^{-1} \left[ (X^T D_t X) w_t + X^T (y - \mu_t) \right]$

$w_{t+1} = (X^T D_t X)^{-1} [X^T D_t z_t]$ where $z_t \triangleq X w_t + D_t^{-1} (y - \mu_t)$

this is a solution to a "weighted least square problem"

$$\min_w \| D^{1/2}(z_0 - Xw) \|_2^2$$

$D$: weights; $z_0$: new target

$$\sum_i \underbrace{(z_i - w^T x_i)^2}_{d_{ii}^{-1}}$$

compare with Gaussian noise model for least square

$$\sum_i \frac{(y_i - w^T x_i)^2}{2\sigma_i^2}$$

Newton's method for logistic regression

= iterated reweighted least squares (IRLS)