Lecture 8 - logistic regression & optimization

Monday, September 30, 2024 12:33 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 (densities) in $\mathbb{R}^d$ for each class conditionals

$p(x|Y=1)$ & $p(x|Y=0)$

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

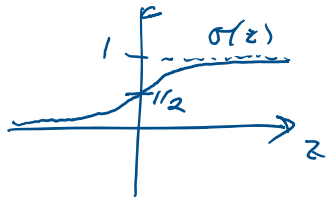
$$= \frac{1}{1 + \frac{p(Y=0, X=x)}{p(Y=1, X=x)}} = \frac{1}{1+\exp(-f(x))}$$

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

"log odd ratio" — class conditional ratio — prior odd ratio

in general

$p(Y=1|X=x) = \sigma(f(x))$ where $\sigma(z) \triangleq \frac{1}{1+\exp(-z)}$ "sigmoid function"



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

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

(*) to motivate linear logistic regression

consider class conditional in the exponential family

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

scalar fct. → log partition fct. → normalize

"canonical parameter" — "sufficient statistics"

these specify the "flat" exponential family

Gaussian:

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

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

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

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

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

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

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

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

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

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

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

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

get logistic regression model

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

"feature map"

decision boundary

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

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

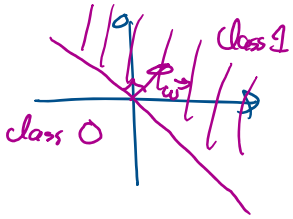
- if $\Sigma_0 = \Sigma_1$, then $\varphi(x) = \begin{pmatrix} x \\ 1 \end{pmatrix}$ → "linear boundary in x"
- otherwise, $\varphi(x) = \begin{pmatrix} -xx^T \\ x \\ 1 \end{pmatrix}$ ~ quadratic decision boundary

Linear logistic regression model

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

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

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



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

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

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

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

$h(w) = w^T x$

$\frac{d}{dw}\sigma(h(w)) = \frac{d}{dz}\sigma(z)\Big|_{z=h(w)} \frac{d}{dw} h(w)$

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

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

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

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

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

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

need to use numerical methods

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

because $\sum_i \frac{1}{1+\exp(w^T x_i)}(\cdots) = 0$

↑ solve for this

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

↑ linear in w

13h28

numerical optimization

want to minimize $f(w)$ (unconstrained) [suppose f is differentiable ?

Unconstrained optimization

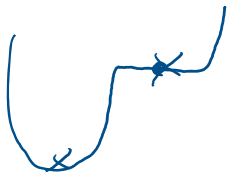
want to minimize $f(w)$ s.t. $w \in \mathbb{R}^d$ (unconstrained) [suppose f is differentiable]

1) gradient descent (1st order method)

start w_0

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

stopping criterion: if $\|\nabla f(w_t)\|^2 < \delta$ then stop



e.g. $\delta = 10^{-6}$

note: If f is μ-strongly convex ($\Leftrightarrow f(w) - \frac{\mu}{2}\|w\|^2$ is convex in w)

$$\Rightarrow f(w_t) - \min_w f(w) \leq \frac{1}{2\mu}\|\nabla f(w_t)\|^2$$

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

Step-size rules

a) constant step-size: $\gamma_t = \frac{1}{L}$ L = Lipschitz continuity constant for ∇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]

$\gamma_t = \frac{c}{t}$ c = constant

usually want $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$

e.g. $f(w) \triangleq \mathbb{E}_z g(w, z)$

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

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

$z = (x_i, y_i)$

c) choose γ_t by "line search": $\min_{\gamma \in \mathbb{R}} f(w_t + \gamma d_t)$ (d_t = direction to update, e.g. $-\nabla f(w_t)$)

costly in general

↳ instead do approximate search

e.g. Armijo line search [see Boyd's book]

Newton's method (2nd order method)

motivation: minimizing a quadratic approximation

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

Newton's method (2nd order method)

method is minimizing a quadratic approximation

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

Taylor expansion at w_t:

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

Taylor's remainder

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

$\nabla_w Q_t(w) = 0$ (want)

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

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

$$\boxed{w_{t+1} = w_t - H^{-1}(w_t) \nabla f(w_t)}$$ Newton's update

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

inverse Hessian $\rightsquigarrow O(d^3)$ time and $O(d^2)$ space

$H d_t = \nabla f$

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

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

use `numpy.linalg.lstsq` to solve this

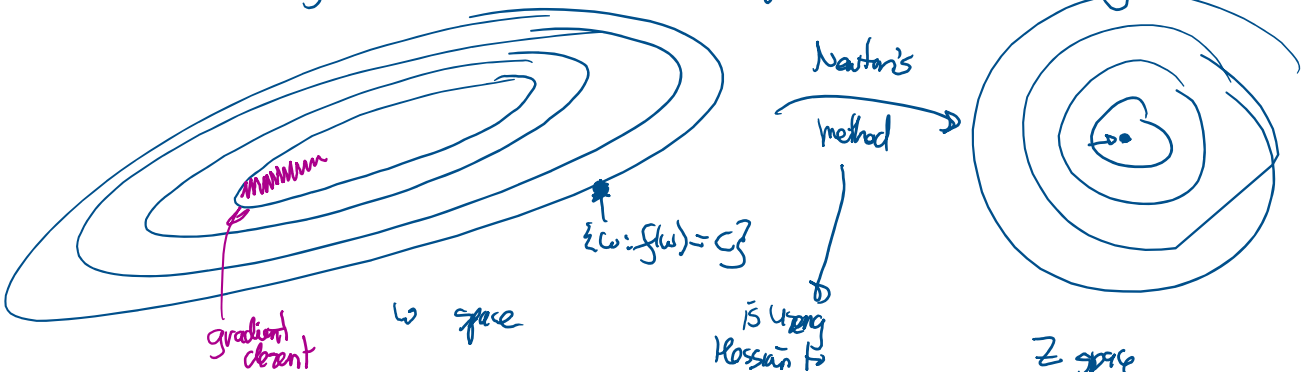
$\rightsquigarrow$ much more numerically stable

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

$w_{t+1} = w_t - \alpha_t H^{-1}(w_t) \nabla f(w_t)$ (α_t: step-size)

Why Newton's method?

- much faster convergence in # iterations vs. gradient descent
- affine invariant $\Rightarrow$ method's convergence is invariant to rescaling of variables



gradient descent

w space

γ is strong Hessian to make f "well conditioned"

z space

$\frac{1}{2} w^T H w = c$ diagonal

$H = P^T \Sigma P$ (H is symmetric psd)

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

$z \triangleq \Sigma^{1/2} P w$

exercise to 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 $\Rightarrow$ use first order methods
- if n is large, you cannot use batch methods

$$\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w)$$

gradient of loss on some data point i

(full) batch gradient

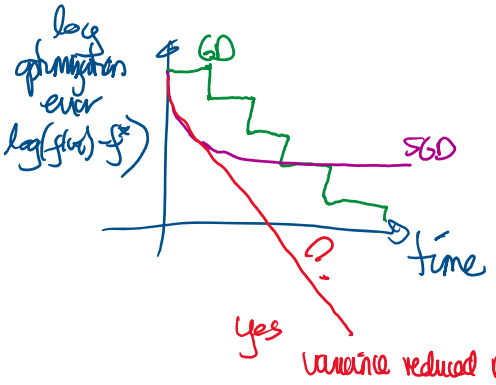
$O(nd)$

instead you can use "incremental gradient method"

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

where i_t is picked uniformly at random

SGD → cheap updates, but slower convergence per iteration



SAG: stochastic average gradient

GD: $w_{t+1} = w_t - \gamma_t \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(w_t)$

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

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

at each t, update $v_{i_t} = \nabla f_{i_t}(w_t)$

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

$\frac{1}{n} \sum_j v_j$ ← SAG/n

variance reduction core

(default method for large logistic regression in Scikit-learn)

SVRG

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

Newton's method for logistic regression: IRLS

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

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

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

i.e. $H \preceq 0$

i.e. concave fct.

$\Updownarrow$

Newton is maximizing instead of minimizing

notation: recall

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

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

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

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

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

$D(w)$: diagonal matrix, $D_{ii} \triangleq \mu_i(1-\mu_i)$

Newton's update:

$$w_{t+1} = w_t - (-X^T D_t X)^{-1} X^T (y - \mu_t)$$

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

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

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

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

$D^{1/2}$: weights; z: 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)