Lecture 7 - scribles

Friday, September 23, 2016
13:23

today:- logistic regression
Fisher LDA and multivariate Gaussians
(generative)

Logistic regression

$\sigma(z) \triangleq \frac{1}{1+\exp(-z)}$

$Y \in \{0,1\}$, $X \in \mathbb{R}^d$

$p(x|y)$ is in exp. family for each y

then $p(y=1|x) = \sigma(w^T \varphi(x))$

parameter; related to sufficient statistics $T(x)$ of exp family

logistic regression model

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

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

[if had encoded $y \in \{-1, 1\}$, $p(y|x) = \sigma(y\, w^T x)$]

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

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

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

maximum conditional likelihood:

$$L(w) = \sum_{i=1}^n \log p(y_i|x_i; w) = \sum_{i=1}^n \left[y_i \log \sigma(w^T x_i) + (1-y_i) \log \sigma(-w^T x_i) \right]$$

$\sigma'(z) = \sigma(z)\sigma(-z)$

$\nabla \sigma(w^T x_i) = x_i \left[\sigma(w^T x_i)\sigma(-w^T x_i) \right]$ $\quad v_i \triangleq w^T x_i$

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

$$\left[y_i \left[\sigma(-v_i) + \sigma(v_i) \right] - \sigma(v_i) \right]$$

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

need numerical optimization

$\nabla L(w) = 0 \Rightarrow$ need to solve transcendental eq.

solve for this

because $\frac{1}{1+\exp(\cdots)}(\cdots) = 0$

(contrast with least squares when get linear in w

$$\sum_{i=1} x_i [y_i - w^T x_i] = 0$$

brief recap on optimization:

want to minimize $\min_{w \in \mathbb{R}^d} f(w)$

1) gradient descent: start at w_0

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

(γ_t: step-size)

step-size rules:

- constant step-size $\gamma_t = \frac{1}{L}$ ← Lipschitz constant of ∇f
- decreasing step-size rule: $\gamma_t = \frac{c}{t}$ (c: constant)

  usually want $\sum_t \gamma_t = \infty$, $\sum_t \gamma_t^2 < \infty$
- choose γ_t by "line-search"; $\min_{\gamma \in \mathbb{R}} f(w_t + \gamma d_t)$ (d_t: direction you are exploring, $[-\nabla f(w_t)]$)

  ↓ costly in general

  → can use instead approximate search

  eg. Armijo line search

2) Newton's method (2nd order method)

motivation: minimize quadratic approximation:

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

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

$[H(w)]_{ij} \triangleq \frac{\partial f(w)}{\partial w_i \partial w_j}$

want $w_{t+1} = \min_w Q_t(w)$

$\nabla Q_t(w) = \nabla f(w_t) + H(w_t)(w - w_t)$

$= 0$

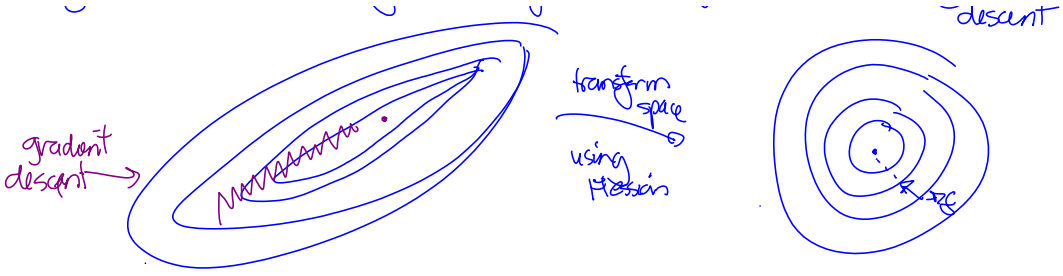
$\Rightarrow w - w_t = -H^{-1}(w_t) \nabla f(w_t)$

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

(Newton's update; H^{-1}: inverse Hessian)

damped Newton: you add a step-size γ: $w_{t+1} = w_t - \gamma H^{-1}(w_t) \nabla f(w_t)$

why Newton: • much faster convergence in # of iterations vs gradient



- affine invariant

but Newton need Hessian → $O(d^2)$ space
$O(d^3)$ time

Newton for logistic regression: IRLS:

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

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

$v^T H v = -\sum_{i=1}^n (v^T x_i)^2 \sigma(-)\sigma(-)$ ($(v^T x_i)^2 \geq 0$, $\sigma(-)\sigma(-) > 0$)

$v^T H v \leq 0 \ \forall v$

i.e. $H \leq 0$

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

Let's look at minimizing the negative log-likelihood

recall design $X = \begin{pmatrix} - x_i^T - \\ \vdots \end{pmatrix}$ $n \times d$

then Hessian $= \sum_{i=1}^n x_i x_i^T [\mu_i(1-\mu_i)]$

gradient $= \sum_{i=1}^n x_i [\mu_i - y_i] = X^T(\mu - y)$

$= X^T (DX)$ where $D_{ii} = \mu_i(1-\mu_i)$] depends on w

$[\cdots x_i \cdots] \begin{bmatrix} D_{ii} x_i^T \\ \vdots \end{bmatrix}$

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

$= (X^T D X)^{-1} [X^T D X w_t + X^T(y - \mu)]$

$= (X^T D X)^{-1} [X^T D z_t]$ $\quad z_t \triangleq X w_t + D^{-1}(y - \mu)$

$(z_t)_i = x_i^T w_t + \frac{y_i - \mu_i}{\mu_i(1-\mu_i)}$

this is solution to "weighted least square problem"

new target

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

$\sum_{i=1}^n \frac{(z_i - x_i^T w)^2}{d_{ii}^{-1}}$

compare with Gaussian noise model for least square $\sum_{i=1}^n \frac{(y_i - x_i^T w)^2}{2\sigma^2}$ (noise)

Newton's method for logistic regression = iterative reweighted least squares

Newton's method for logistic regression = iterative reweighted least squares

big data logistic regression

cannot do $O(d^2)$, $O(d^3)$ operations

⇒ first order methods

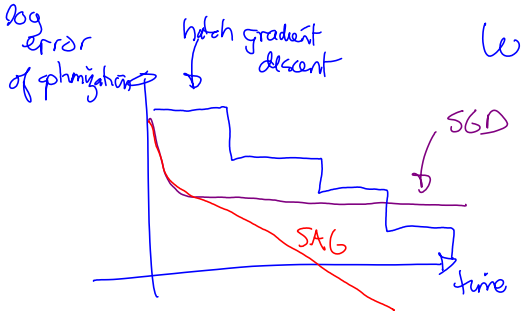
if n is huge, you cannot use batch method $\nabla f = \sum_{i=1}^{n} \nabla f_i(w)$ ($O(n)$)

instead you can use SGD: stochastic gradient descent suppose $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$

SGD: $w_{t+1} = w_t - \gamma_t \nabla f_i(w_t)$ (much cheaper update but slow convergence)

batch gradient $w_{t+1} = w_t - \gamma_t \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_t)$ (expensive update, fast convergence)

SAG: stochastic average gradient:



$w_{t+1} = w_t - \gamma \frac{1}{n}\sum_{i=1}^{n} v_i$ where $v_i = \nabla f_i(w_{old})$

and at each t, update only one $v_i^t = \nabla f_i(w_t)$

Kernels

recall for least-square, gradient $\sum_{i=1}^{n} x_i (y_i - w^T x_i)$ (scalars)

log-regression $\sum_{i=1}^{n} x_i (y_i - \sigma(w^T x_i))$

you always have $w_t = \sum_{i=1}^{n} \alpha_i x_i$ (α_i scalars)

e.g. least-square:

$w = (X^T X + \lambda I_d)^{-1} X^T y$

$= X^T \alpha$ where $\alpha = (XX^T + \lambda I_n)^{-1} y$

Gram matrix K

$K_{ij} = x_i^T x_j$

for prediction $w^T x = \sum_{i=1}^{n} \alpha_i (x_i^T x)$

suppose map $x \xrightarrow{\varphi} \varphi(x)$

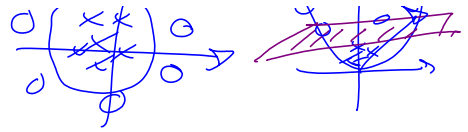
$\tilde{w}^T \varphi(x) = \sum_{i=1}^{n} \alpha_i \varphi(x_i)^T \varphi(x)$

$k(x_i, x)$ kernel

"kernel trick"

→ you can work implicitly in high dimensional space $\varphi(x)$ only using $k(\cdot,\cdot)$ kernel

φ: $(x,y) \to (x^2, \sqrt{2}xy, y^2)$



$z=(x,y)$

$\langle \varphi(z), \varphi(z') \rangle$

$= \left\langle \begin{pmatrix} x^2 \\ \sqrt{2}xy \\ y^2 \end{pmatrix}, \begin{pmatrix} x'^2 \\ \sqrt{2}x'y' \\ y'^2 \end{pmatrix} \right\rangle = (xx')^2 + 2(xx')(yy') + y^2y'^2$

$= (xx' + yy')^2$

$= (\langle z, z' \rangle)^2$