Learning Linear Classifiers
Maximum Likelihood estimation

Quality metric for logistic regression: Maximum likelihood estimation

Learning problem.

Training data: N observations (x_i, y_i)

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
0	2	-1
3	3	-1
4	1	+1
1	4	+1
2	3	-1
0	1	-1
0	1	+1
2		

Optimize quality metric on training data → $\hat{w}$

Finding best coefficients

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1
3	3	-1
2	4	-1
0	3	-1
0	1	-1

$P(y=+1|x_i, w) = 0.0$

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
4	1	+1
1	1	+1
2	1	+1

$P(y=+1|x_i, w) = 1.0$

Pick $\hat{w}$ that makes

Quality metric = Likelihood function

Negative data points: $P(y=+1|x_i,w)=0.0$

Positive data points: $P(y=+1|x_i,w)=1.0$

No $\hat{w}$ achieves perfect predictions (usually)

Likelihood $\ell(w)$: Measures quality of fit for model with coefficients w

Find "best" classifier

Maximize likelihood over all possible w_0, w_1, w_2

$\ell(w_0=1, w_1=0.5, w_2=-1.5)=10^{-4}$

Best model:

Highest likelihood $\ell(w)$

$\hat{w}=(w_0=1, w_1=0.5, w_2=1.5)$

gradient ascent to find $\hat{w}$

Data likelihood

Quality metric: probability of data

$x[1]$ = #awesome	$x[2]$ = #awful	y = sentiment
x_1 2	1	y_1 +1

If model good, should predict $\hat{y}_1=+1$

Pick w to maximize:

$$P(y=+1|x_1,w)=P(y=+1|x[1]=2, x[2]=1, w)$$

x[1] = #awesome x[2] = #awful y = sentiment

x_2 = 0 2 y_2 = -1

if model good, should predict $\hat{y}_2 = -1$

Pick w to maximize

$P(y=-1|x_2,w)$

Maximizing (likelihood / probability of data)

Data point	x[1]	x[2]	y
x_1, y_1	2	1	+1
x_2, y_2	0	2	-1
x_3, y_3	3	3	-1
x_4, y_4	4	1	+1
x_5, y_5	1	1	+1
x_6, y_6	2	4	-1
x_7, y_7	0	3	-1
x_8, y_8	0	1	-1
x_9, y_9	2	1	+1

Choose w to maximize

$P(y=+1|x_1,w) = P(y=+1|x[1]=2, x[2]=1, w)$

$P(y=-1|x_2,w)$

$P(y=-1|x_3,w)$

$P(y=+1|x_4,w)$

Must combine into single measure of quality?

Multiply probability

$P(y=+1|x_1,w)P(y=-1|x_2,w)P(y=-1|x_3,w)\ldots$

Learn logistic regression model with maximum likelihood estimation (MLE)

Data point	x[1]	x[2]	y	Choose w to maximize
x_1, y_1	2	1	$y_1 = +1$	$P(y=+1 \mid x[1]=2, x[2]=1, w)$
x_2, y_2	0	2	-1	$P(y=-1 \mid x[1]=0, x[2]=2, w)$
x_3, y_3	3	3	$y_3 = -1$	$P(y=-1 \mid x[1]=3, x[2]=3, w)$
x_4, y_4	4	1	$+1$	$P(y=+1 \mid x[1]=4, x[2]=1, w)$

$$\ell(w) = P(y=+1 \mid x[1]=2, x[2]=1, w)\, P(y=-1 \mid x[1]=0, x[2]=2, w) \cdot P(y=-1 \mid x[1]=3, x[3]=3, w) P(y=+1 \mid x[1]=4, x[2]=1, w)$$

$$P(y=-1 \mid x[1]=3, x[3]=3, w) \rightarrow P(y_3 \mid x_3, w)$$

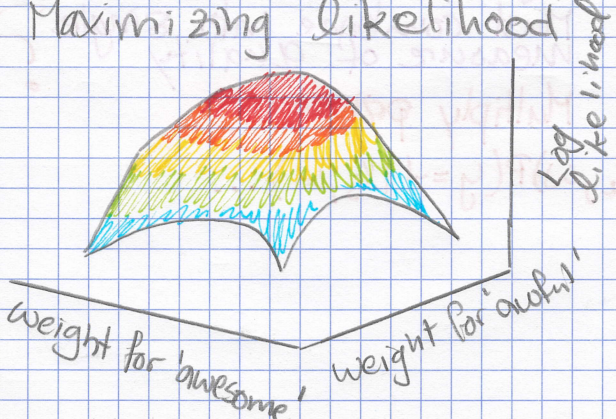
$$\ell(w) = \prod_{i=1}^{N} P(y_i \mid x_i, w)$$

← pick w to make this fn. as large as possible

num. of data points

Finding best linear classifier with gradient ascent

Maximizing likelihood



Maximize function over all possible w_0, w_1, w_2

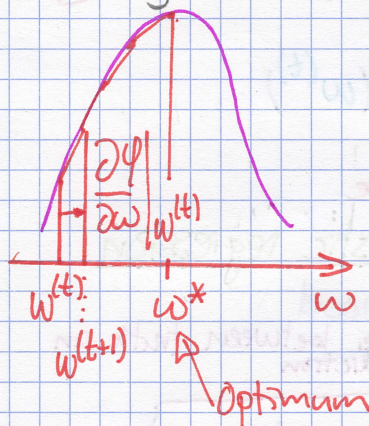
$$\max_{w_0, w_1, w_2} \prod_{i=1}^{N} P(y_i \mid x_i, w)$$

$\ell(w_0, w_1, w_2)$ is a function of 3 variables

No closed-form solution → use gradient ascent

Review of gradient ascent

Finding the max via hill climbing



Algorithm

While not converged

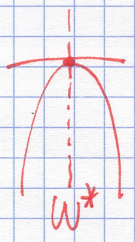
$$w^{(t+1)} \leftarrow w^{(t)} + \eta \frac{\partial \varphi}{\partial w}\Big|_{w^{(t)}}$$

η: step size

Convergence criteria

For convex functions, optimum occurs when

$$\frac{\partial \varphi}{\partial w} = 0$$



In practice, stop when

$$\left|\frac{\partial \varphi}{\partial w}\Big|_{w^{(t)}}\right| < \varepsilon$$

ε: tolerance

Moving to multiple dimensions:

Gradients

$$\nabla \varphi(w) = \begin{bmatrix} \frac{\partial \varphi}{\partial w_0} \\ \frac{\partial \varphi}{\partial w_1} \\ \vdots \\ \frac{\partial \varphi}{\partial w_D} \end{bmatrix}$$

D+1 dim. vector

Gradient ascent

Algorithm $w^{(0)} = 0$, random, or something else

while not converged:

$$w^{(t+1)} \leftarrow w^{(t)} + \eta \nabla \ell(w^{(t)})$$

η: stepsize

Learning algorithm for logistic regression

Derivative of (log-)likelihood

$$\frac{\partial \ell(w)}{\partial w_j} = \sum_{i=1}^{N} h_j(x_i)\left(\mathbb{1}[y_i = +1] - P(y=+1 \mid x_i, w)\right)$$

Sum over data points; Feature value; Difference between truth an prediction

$P(y=+1 \mid x_i, w)$: predict x_i is positive

Indicator function

$$\mathbb{1}[y_i = +1] = \begin{cases} 1 \text{ if } y_i = +1 \\ 0 \text{ if } y_i = -1 \end{cases}$$

Example of computing derivative for logistic regression

Computing derivative

$$\frac{\partial \ell(w^{(t)})}{\partial w_j} = \sum_{i=1}^{N} h_j(x_i)\left(\mathbb{1}[y_i = +1] - P(y=+1 \mid x_i, w^{(t)})\right)$$

$w_i^{(t)}$

$w_0^{(t)}$	0
$w_1^{(t)}$	-1
$w_2^{(t)}$	-2

$\frac{\partial \ell}{\partial w_1}$

$h_1(x) =$ # awesome

x[1]	x[2]	y	$P(y=+1 \mid x_i, w)$	Contribution to derivative for w_1
2	1	+1	0.5	$2(1-0.5)=1$
0	2	-1	0.02	$0(0-0.02)=0$
3	3	-1	0.05	$3(0-0.05)=-0.15$
4	1	+1	0.88	$4(1-0.88)=0.48$

Total derivative:

$$\frac{\partial \ell(w^{(t)})}{\partial w_1} = 1+0-0.15+0.48 = 1.33$$

$$w_1^{(t+1)} = w_1^{(t)} + \eta \frac{\partial \ell(w^{(t)})}{\partial w_1} \quad | \quad \eta = 0.1$$

$$= 1 + 0.1 * 1.33 = 1.133$$

Interpreting derivative for logistic regression

Derivative of (log-)likelihood. Interpretation

$$\frac{\partial \ell(w)}{\partial w_j} = \sum_{i=1}^{N} h_j(x_i)\left(\mathbb{1}[y_i=+1] - P(y=+1 \mid x_i, w)\right)$$

Sum over data points; feature value; difference between truth and prediction; Δ_i

If $h_j(x_i)=1$:

$y_i=+1$: $P(y=+1 \mid x_i, w) \approx 1$: $\Delta_i = (1-1) \approx 0$ → don't change anything

$P(y=+1 \mid x_i, w) \approx 0$: $\Delta_i \approx 1 \Rightarrow$ increase w_j $\Rightarrow$ score(x_i) becomes larger $\Rightarrow P(y=+1 \mid x_i, w)$ increases

$y_i=-1$: $\Delta_i = -1 \Rightarrow w_j$ to decrease $\Rightarrow$ score(x_i) decreases $\Rightarrow P(y=+1 \mid x_i, w)$ decrease

$\Delta_i \approx 0 \Rightarrow$ don't change anything

Summary of gradient ascent for logistic regression

init $w^{(1)} = 0$ (or randomly, or smartly), $t = 1$

while $\|\nabla \ell(w^{(t)})\| > \varepsilon$ ← tolerance

for $j = 0, \ldots, D$: ← coefficient

$$\text{partial}[j] = \sum_{i=1}^{N} h_j(x_i)\left(\mathbb{1}[y_i = +1] - P(y = +1 \mid x_i, w^{(t)})\right)$$

$P(y = +1 \mid x_i, w^{(t)}) = \frac{1}{1 + e^{-w^{(t)T} h(x_i)}}$

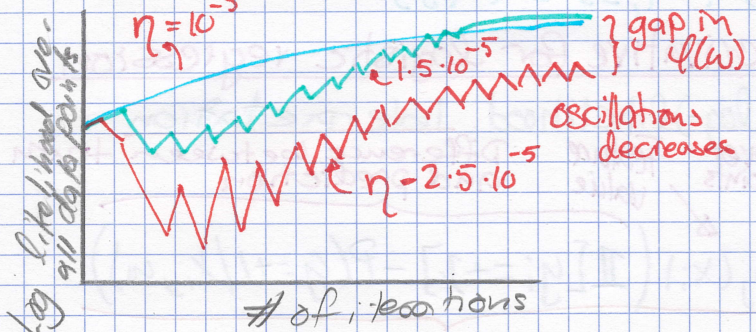
$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta \, \text{partial}[j]$$

$t \leftarrow t + 1$

η ← stepsize; $\frac{\partial \ell(w^{(t)})}{\partial w_j}$

Choosing step size η

- Careful with step sizes that are too large



- Very large step sizes can even cause divergence or wild oscillations

Rule of thumb for choosing step size

Simple rule of thumb for picking stepsize η

- Unfortunately, picking step size requires a lot of trial and error
- Try a several values, exponentially spaced
  - Goal: plot learning curves to
    - find one η that is too small (smooth but moving too slowly)
    - find one η that is too large (oscillating or divergence)

- Try values in between to find "best" η
  - ↳ exponentially spaced, pick one that leads best training data likelihood
- Advanced tip: can also try step size that decreases w/ iterations, e.g.,

$$\eta_t = \frac{\eta_0}{t}$$