Logistic regression (LR)

- linear decision rule with different w, b than LDA
- posterior probabilities (= soft classification) $p(Y=k \mid X)$
- idea: - use the generative model of LDA ($\hat{=}$ RHS of Bayes formula) and transform it into the corresponding posterior (LHS of Bayes)
  - train w such that the error rate of the maximum a-posteriori rule (MAP) is minimized $\hat{Y}_i = \arg\max_k p(Y=k \mid X_i)$ with $p(Y=k \mid X) = \frac{p(X \mid Y=k)\, p(Y=k)}{p(X)}$
- simplification: centralize the data before training:

  mean $\mu = \frac{1}{N} \sum_{i=1}^{N} X_i$, new features $\tilde{X}_i = X_i - \mu$

  [to avoid cluttered notation: do not write $\tilde{X}_i$, but X_i with implicit assumption that X_i are already centered)

  useful, ~~also~~ because $b = 0$ for 2-class classification

  $\hat{Y}_i = \text{sign}(w X_i^T)$ if X_i are centered
- simplification: balanced classes for $C=2$ $N_{-1} = N_{+1} = N/2$

$$\Rightarrow \frac{1}{N_{-1}} \sum_{i: Y_i = -1} X_i = \underline{\underline{\mu_{-1} = -\mu_1}} = -\frac{1}{N_{+1}} \sum_{i: Y_i = 1} X_i$$

$$\mu = \frac{1}{N} \sum_i X_i \overset{!}{=} 0 = \frac{1}{2N_{-1}} \sum_{i: Y_i = -1} X_i + \frac{1}{2N_1} \sum_{i: Y_i = 1} X_i = \frac{1}{2} \mu_{-1} + \frac{1}{2} \mu_1$$

- calculate the posterior for LDA when $N_{-1} = N_{+1}$ and $p(Y=-1) = p(Y=+1) = \frac{1}{2}$
  (empirical priors $\hat{p}(Y=k) = \frac{N_k}{N}$ will always fulfill this for balanced classes)

$$p(Y=1 \mid X) = \frac{p(X \mid Y=1)\, p(Y=1)}{p(X)} = \frac{p(X \mid Y=1)\, \cancel{p(Y=1)}}{p(X \mid Y=-1)\, \cancel{p(Y=-1)} + p(X \mid Y=1)\, \cancel{p(Y=1)}}$$

$$\boxed{p(Y=+1 \mid X) = \frac{1}{1 + \frac{p(X \mid Y=-1)}{p(X \mid Y=+1)}}}$$

always true for $C=2$ and $p(Y=-1) = p(Y=+1)$

insert $p(X \mid Y=k)$ as multi-variate Gaussian $\mathcal{N}(\mu_k, \Sigma_w)$

$$\frac{p(X \mid Y=-1)}{p(X \mid Y=+1)} = \frac{\cancel{\frac{1}{\sqrt{\det(2\pi\Sigma_w)}}} \exp\left(-\frac{1}{2}(X-\mu_{-1})\Sigma_w^{-1}(X-\mu_{-1})^T\right)}{\cancel{\frac{1}{\sqrt{\det(2\pi\Sigma_w)}}} \cdot \exp\left(-\frac{1}{2}(X-\mu_1)\Sigma_w^{-1}(X-\mu_1)^T\right)}$$

$$(X-\mu_k)\Sigma_w^{-1}(X-\mu_k)^T = X\Sigma_w^{-1}X^T - 2X\Sigma_w^{-1}\mu_k^T + \mu_k\Sigma_w^{-1}\mu_k^T$$

$$\frac{p(X \mid Y=-1)}{p(X \mid Y=+1)} = \exp\Big(\cancel{-\frac{1}{2}X\Sigma_w^{-1}X^T} + X\Sigma_w^{-1}\mu_{-1}^T - \frac{1}{2}\mu_{-1}\Sigma_w^{-1}\mu_{-1}^T + \cancel{\frac{1}{2}X\Sigma_w^{-1}X^T}$$
$$- X\Sigma_w^{-1}\mu_1^T + \frac{1}{2}\mu_1\Sigma_w^{-1}\mu_1^T\Big)$$

$$= \exp\Big(-X\Sigma_w^{-1}(\mu_1-\mu_{-1})^T + \frac{1}{2}\underbrace{\left(\mu_1\Sigma_w^{-1}\mu_1^T - \mu_{-1}\Sigma_w^{-1}\mu_{-1}^T\right)}_{=0 \text{ because } \mu_1 = -\mu_{-1}}\Big)$$

$$= \exp\left(-X\Sigma_w^{-1}(\mu_1-\mu_{-1})^T\right)$$

$$p(Y=+1 \mid x) = \frac{1}{1 + \frac{p(x \mid Y=-1)}{p(x \mid Y=+1)}} = \frac{1}{1 + \exp\left(-x \underbrace{\Sigma_w^{-1} (\mu_1 - \mu_{-1})^T}_{w^T}\right)}$$

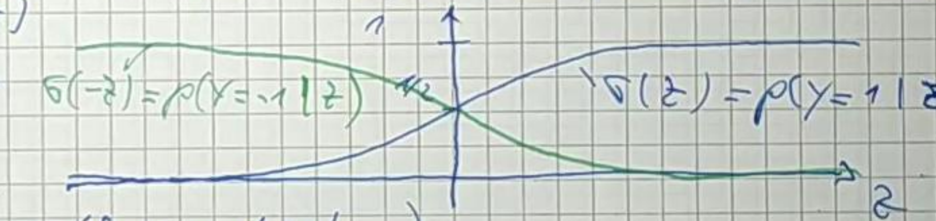
$$w = (\mu_1 - \mu_{-1}) \Sigma_w^{-1} \qquad = \boxed{\frac{1}{1 + \exp(-w x^T)} = p(Y=+1 \mid x)}$$

(remember: x_i are centered)

the function $\sigma(z) = \dfrac{1}{1 + \exp(-z)}$ is called "logistic sigmoid function"

sigmoid $\hat{=}$ graph looks like an "S"

logistic $\hat{=}$ posterior of logistic regression



(other sigmoid exist, e.g. error function, tanh)

properties:
$$p(Y=+1 \mid x) = \sigma(w x^T)$$
$$p(Y=-1 \mid x) = 1 - p(Y=+1 \mid x) = 1 - \sigma(w x^T) = \sigma(-w x^T)$$

e.g. $\cancel{z \geq 0}$ $z = w x^T > 0 \Rightarrow$ vote for class $+1$

$< 0 \Rightarrow$ vote for class -1

[proof: $1 - \dfrac{1}{1 + \exp(-z)} = \dfrac{1 + \exp(-z) - 1}{1 + \exp(-z)} = \dfrac{1}{\exp(z) + 1} = \sigma(-z)$]

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

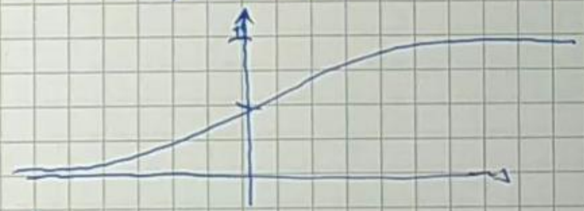
Logistic regression continued

- assume centered data $\frac{1}{N}\sum_{i=1}^{N} X_i = 0$, balanced classes $N_1 = N_0 = \frac{N}{2}$ $\quad Y_i \in \{0, 1\}$
- Bayes formula: $p(Y=k \mid X) = \frac{p(X \mid Y=k)\, p(Y=k)}{p(X)}$ with $p(X \mid Y=k) = \mathcal{N}(\mu_k, \Sigma_w)$

  common co-variance for all classes

  then the posterior becomes $p(Y=1 \mid X_i) = \sigma(w X_i^T) = \frac{1}{1+\exp(-w X_i^T)}$

  $p(Y=0 \mid X_i) = \sigma(-w X_i^T)$



- LDA: choose $w = (\mu_1 - \mu_0) \cdot \Sigma_w^{-1}$

  result of optimal fit of RHS of Bayes formula
- Logistic regression (LR) optimally fits the LHS of Bayes

  maximum likelihood principle: $\hat{w} = \arg\max_w p(Y_1, \dots, Y_N \mid X_1, \dots, X_N)$

  simplify via the i.i.d. assumption: joint probability factorizes

  $$p(Y_1, \dots, Y_N \mid X_1, \dots, X_N) = \prod_{i: Y_i = 0} p(Y=0 \mid X_i) \cdot \prod_{i: Y_i = 1} p(Y=1 \mid X_i)$$

  trick: use $Y_i \in \{0, 1\}$

  $$= \prod_{i=1}^{N} p(Y=0 \mid X_i)^{(1-Y_i)} \cdot p(Y=1 \mid X_i)^{Y_i}$$

  take negative logarithm

  $$-\log p(Y_1, \dots, Y_N \mid X_1, \dots, X_N) = -\sum_{i=1}^{N} \Big[(1-Y_i) \log \underbrace{p(Y=0 \mid X_i)}_{\sigma(-w X_i^T)} + Y_i \log \underbrace{p(Y=1 \mid X_i)}_{\sigma(w X_i^T)} \Big]$$

~~optimization objective:~~ $\cancel{\hat{w} = \arg\min_w - \sum_{i=1}^{N} (1-Y_i)\, \sigma(-w X_i^T) + Y_i}$

optimization objective of LR:

$$\boxed{\hat{w} = \arg\min_{w} - \sum_{i=1}^{N} \left[(1-Y_i) \log \sigma(-w X_i^T) + Y_i \log \sigma(w X_i^T) \right]}$$

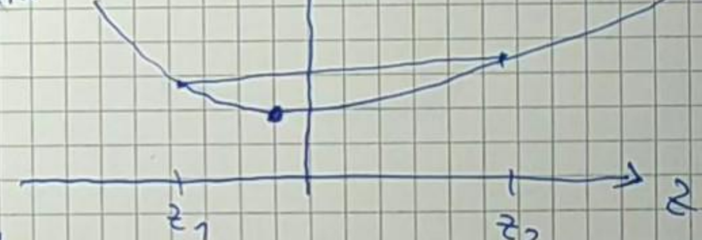
turns out that we now get a different $\hat{w}$ than LDA

- but there is no closed-form expression for $\hat{w}$ $\Rightarrow$ need iterative alg.

iterative alg. will always find the optimal solution, because the objective is convex [$f(z)$ is convex if $f(\tau z_1 + (1-\tau) z_2) \leq \tau f(z_1) + (1-\tau) f(z_2)$ with $\tau \in [0,1]$, intuition: the straight line between $f(z_1)$ and $f(z_2)$ is always above the graph of $f(z)$ between z_1 and z_2

strictly convex: $\tau \in (0,1)$ the "<" is valid

$\Rightarrow$ function f has only one minimum, which is automatically the global one]



- to optimize, take derivative w.r.t. w and set to zero

derivative of logistic function $\sigma'(z) = \frac{\partial}{\partial z} \frac{1}{1+e^{-z}} = \frac{\partial}{\partial z} (1+e^{-z})^{-1} = (-1)(1+e^{-z})^{-2}(-1)e^{-z}$

$$= \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}} = \frac{1}{1+e^{-z}} \cdot \frac{1}{1+e^{z}}$$

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

$$\frac{\partial}{\partial w} (1-Y_i) \log \sigma(-w X_i^T) = (1-Y_i) \frac{1}{\cancel{\sigma(-w X_i^T)}} \cdot \underbrace{\cancel{\sigma(-w X_i^T)}\, \sigma(w X_i^T) (-X_i)}_{= \sigma'(-w X_i^T)}$$

$$\frac{\partial}{\partial w} Y_i \log \sigma(w x_i^T) = Y_i \frac{1}{\cancel{\sigma(w x_i^T)}} \cancel{\sigma(w x_i^T)} \cdot \sigma(-w x_i^T) \cdot x_i$$

$$\frac{\partial}{\partial w} - \sum_{i=1}^{N} \dots = - \sum_{i=1}^{N} \Big[(1-Y_i)\, \sigma(w x_i^T)(-x_i) + Y_i \underbrace{\sigma(-w x_i^T)}_{(1-\sigma(w x_i^T))} x_i \Big]$$

$$= - \sum_{i=1}^{N} \sigma(w x_i^T)(-x_i) + \cancel{Y_i\, \sigma(w x_i^T)(+x_i)} + Y_i x_i - \cancel{Y_i\, \sigma(w x_i^T) x_i}$$

$$= \boxed{\sum_{i=1}^{N} (\sigma(w x_i^T) - Y_i)\, x_i \overset{!}{=} 0}$$ cannot analytically solve for w

if $\sigma(w x_i^T) \approx 1$ for instances with $Y_i = 1 \Rightarrow (\sigma(w x_i^T) - Y_i) \approx 0 \Rightarrow$ gradient vanishes

$\hat{=}\; w x_i^T \to \infty$

if $\sigma(w x_i^T) \approx 0$ for instances with $Y_i = 0 \Rightarrow (\sigma(w x_i^T) - Y_i) \approx 0 \Rightarrow$ gradient vanishes

$\hat{=}\; w x_i^T \to -\infty$

if an $Y_i = 1$ instance is wrongly classified by the current guess w $\hat{=}\; \sigma(w x_i^T) < 1$

$\Rightarrow$ gradient $(\sigma(w x_i^T) - Y_i)$ is negative $\Rightarrow$ change $w x_i^T$ in the opposite direction to make it bigger $\Rightarrow \sigma(w x_i^T)$ gets closer to 1

if an $Y_i = 0$ instance is wrongly classified $\Rightarrow (\sigma(w x_i^T) - Y_i) > 0 \Rightarrow$ pulls in the opposite direction to make $\sigma(w x_i^T)$ smaller

$\Rightarrow$ <u>solve by gradient descent</u> • initial guess $w_0 = 0$, learning rate τ

• for $t = 1, \dots, T$ # of iterations: $w_t = w_t - \tau \sum_{i=1}^{N} (\sigma(w_{t-1} x_i^T) - Y_i)\, x_i$

minus, because we want to <u>minimize</u> the objective $\Rightarrow$ gradient <u>descent</u>

variations of gradient descent: - mini-batch descent: only include a ~~finite~~ random subset of TS in the sum (choose B instances uniformly at random $\hat{=}$ mini-batch) faster per iteration (only $O(B)$ instead $O(N)$), but more iterations

- stochastic gradient descent: $B = 1$, i.e. approximate sum by a single random instance (SGD) $\Rightarrow$ very fast per iteration, but even more iterations
- learning rate schedule: decrease τ regularly, e.g. $\tau = \frac{t}{t+c} \cdot \tau_0$ ← initial τ (especially important for mini-batch SGD for neural nets) t ← iteration counter
- in practice: split loop into two

  for $e = 1, \dots, E$ e: epoch   E number of epochs

    shuffle the TS

    for $i = 1, \dots, N$:

      update w with the gradient of instance $\pi(i)$ ← permutation after shuffle

- faster algorithm: Newton alg. or quasi-Newton alg. needs much fewer iterations, ~~both~~ but each iteration is expensive
- fastest alg. depends on size of training set N
  - N small: Newton is fastest (few iterations, relatively cheap iterations)
  - N large: (mini-batch) SGD is fastest (many iterations, but compensated because ~~eda~~ each iteration is very cheap)