Classification

Training data $\{(\underset{\sim}{x}_i, y_i)\}_{i=1}^{N}$

- $\underset{\sim}{x}_i$: predictors, features
- y_i: response, labels

y_i: categorical / qualitative

x_{ij}: continuous, ordinal, categorical

Goal: Learn from training data a good classification function / classifier $\hat{y} = \hat{f}(\underset{\sim}{x})$ or $\hat{y} = \hat{C}(\underset{\sim}{x})$

Ex: $y_i \in C = \{\underbrace{\text{"good email"}}_{\text{"0"}}, \underbrace{\text{"spam"}}_{\text{"1"}}\}$

$\underset{\sim}{x}_i = [x_{i1}, \dots, x_{ip}]^T$

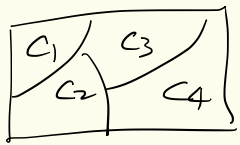
- x_{i1}: total word count
- x_{ip}: number of irregular spelling

Step 1: Use a naive Bayes to learn $\hat{y} = \hat{C}_{Tr}(\underset{\sim}{x})$

Step 2: For incoming data $\underset{\sim}{x}^0$, predict if spam or not using $\hat{C}_{Tr}(\underset{\sim}{x}^0)$.

Error metric for multiclass classifiers:

- error rate : $\frac{1}{M} \sum_{i=N+1}^{N+M} \mathbb{1}(y_i \neq \hat{y}_i)$



Bayesian Decision Theory (Murphy 5.7)

Loss Function $L(y, c)$, e.g., $L = \mathbb{1}[y \neq c]$ 0-1 loss

↑ true label ↑ label/class (to be) assessed, "action"

Expected loss $\rho(c|\underset{\sim}{x}) = \mathbb{E}\left[L(y,c) \,\middle|\, \underset{\sim}{x} \right]$

$$= \sum_{y \in \{c_1, \ldots, c_k\}} L(y,c)\, p(y|\underset{\sim}{x})$$

Bayes Decision Rule:

$$\hat{C}(\underset{\sim}{x}) = \arg\min_c \rho(c|\underset{\sim}{x}) \qquad L(y,c) = \mathbb{1}[y \neq c]$$

$$= \arg\min_c \sum_y L(y,c)\, p(y|\underset{\sim}{x})$$

$$= \arg\min_c \left(1 - p(c|\underset{\sim}{x})\right)$$

$$= \arg\max_{y \in \{c_1, \ldots, c_k\}} p(y|\underset{\sim}{x})$$

Maximum a posteriori (MAP) estimator

Will be used for "logistic regression" estimation.

Error Metrics for binary classifiers:

- False positive rate (FPR, Type-1 error)
  "alarm"
- False negative rate (FNR, Type-2 error)
  "miss"

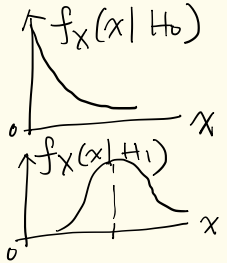
Decision \ Ground truth	H_0	H_1
0	TN	FN
1	FP	TP

[TPR = sensitivity, recall. TNR = specificity, selectivity]

Ex: (Simple thresholding rule for COVID detection)

H_0 (not infected): $f_X(x|H_0) = \lambda_0 e^{-\lambda_0 x}$, $x \geq 0$

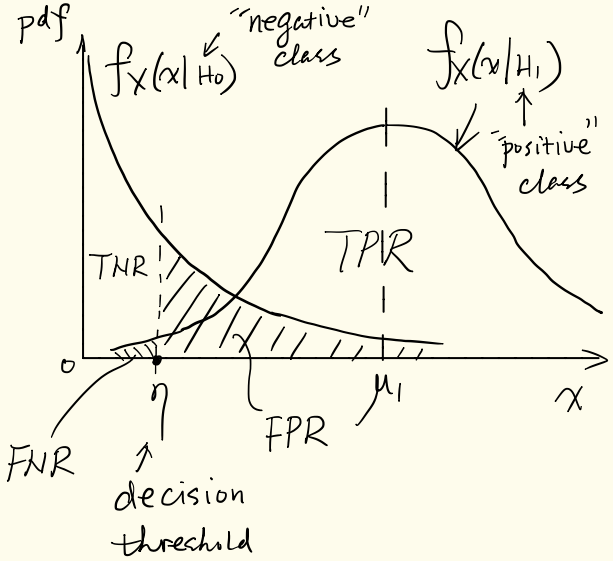
H_1 (infected): $f_X(x|H_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu)^2}{2\sigma_1^2}}$, $x \in \mathbb{R}$



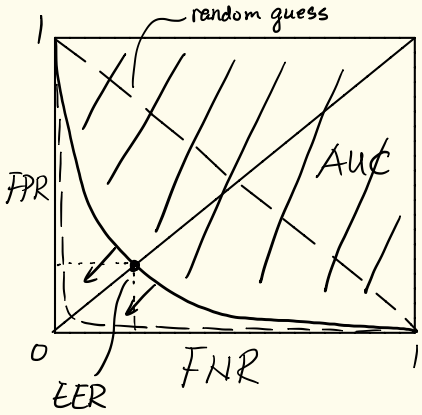
Decision rule / classifier $\hat{C}(x) = \begin{cases} 1, & x \geq \eta, \\ 0, & x < \eta. \end{cases}$

$$FPR(\eta) = \int_{\eta}^{\infty} f_X(x|H_0)\,dx$$
$$= 1 - F_0(\eta)$$
$$= 1 - (1 - e^{-\lambda_0 \eta}) = e^{-\lambda_0 \eta}$$

$$FNR(\eta) = \int_{-\infty}^{\eta} f_X(x|H_1)\,dx$$
$$= F_1(\eta) = \Phi\left(\frac{\eta - \mu_1}{\sigma_1}\right)$$

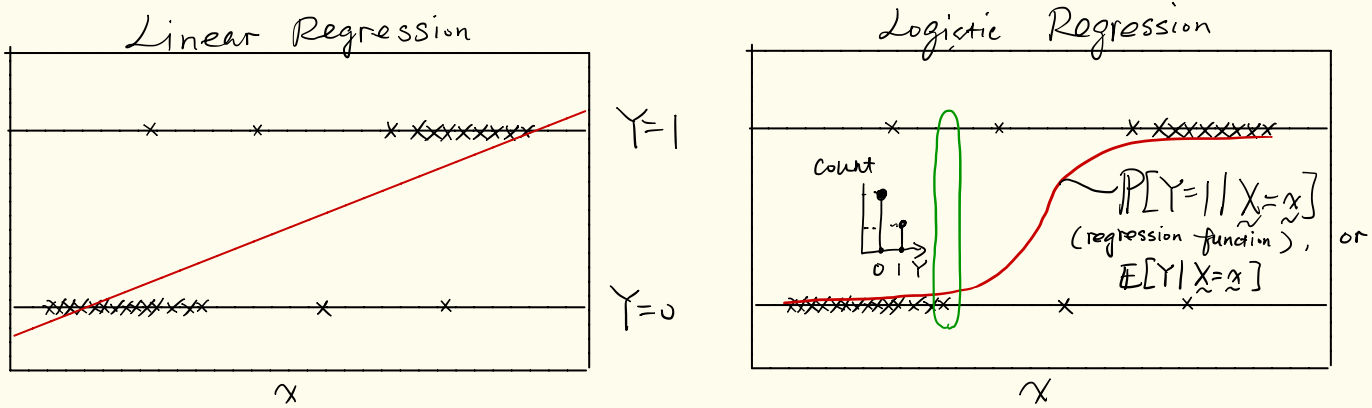


- ROC Curve $\overset{\text{def}}{=} \{(FNR(\eta), FPR(\eta))\}_{\eta}$
- EER: Equal Error Rate $\overset{\text{def}}{=} FNR(\eta) = FPR(\eta)$
- AUC: Area Under Curve $\overset{\text{def}}{=} \int_0^1 TPR(FNR)\,d(FNR)$



Logistic Regression (LR) ESL 4.4, Murphy 8.1-3

Why logistic for binary data?



Need a nonlinear mapping g:

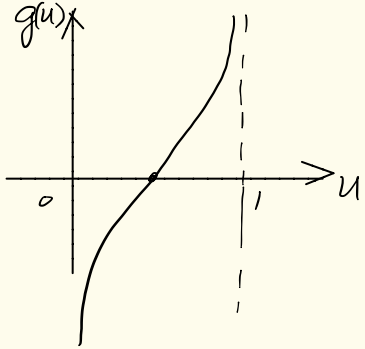
$$g: [0,1] \to \mathbb{R}$$

$$\mathbb{P}[Y=1 \mid X=x_i] \mapsto x_i^T \beta$$

$$\overset{\text{def}}{=} p(x_i)$$

Not pdf/pmf, Not likelihood of x_i

- "Logit": $g(u) = \log\left(\frac{u}{1-u}\right)$
- "Probit": $g(u) = \Phi^{-1}$ (historically)
  inverse of CDF for Gaussian



Model:

- $\log\left(\frac{\mathbb{P}[Y=1 \mid \underset{\sim}{x}]}{\mathbb{P}[Y=0 \mid \underset{\sim}{x}]}\right) = \log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$ (odds; log odds)
- $g(\underbrace{p(x)}_{u}) = \beta_0 + \beta_1 x$, where $g(u) = \log\left(\frac{u}{1-u}\right)$, or
- $p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$

Parameter Estimation for Logistic Regression

To estimate (β_0, β_1). Training dataset $\{(x_i, y_i)\}_{i=1}^n$ (x_i: predictor, y_i: label.)

How? Use Bayes classifier $\max p(y|x)$, which becomes MLE for (β_0, β_1), treating labels $\{y_i\}$ as the "data" for MLE purposes: $\max_{\beta} \prod_{i=1}^n \mathbb{P}[Y = y_i | x_i]$

$$L(\beta) = \prod_{i=1}^n \mathbb{P}[Y = y_i | X = x_i]$$

$$= \prod_{i=1}^n p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i} \qquad \beta = (\beta_0, \beta_1)$$

$$\ln L(\beta) = \ell(\beta) = \sum_{i=1}^n y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i)) \leftarrow \text{"cross entropy"}$$

$$= \sum \log (1 - p(x_i)) + y_i \log \underbrace{\frac{p(x_i)}{1 - p(x_i)}}_{\beta_0 + \beta_1 x}$$

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

$$= \sum -\ln(1+e^{\beta_0+\beta_1 x_i}) + \sum y_i(\beta_0+\beta_1 x_i)$$

$$\frac{\partial \ell(\beta)}{\partial \beta_1} = -\sum \frac{1}{1+e^{\beta_0+\beta_1 x_i}} \cdot e^{\beta_0+\beta_1 x_i} \cdot x_i + \sum y_i x_i$$

$$= -\sum p(x_i) x_i + \sum y_i x_i = \sum [y_i - \underbrace{p(x_i;\beta_0,\beta_1)}_{\text{nonlinear, needs IRLS}}] x_i$$

$$= \Big|_{\beta=\hat{\beta}} 0$$

B+w, for linear regression, using MLE (= LS when Gaussian)

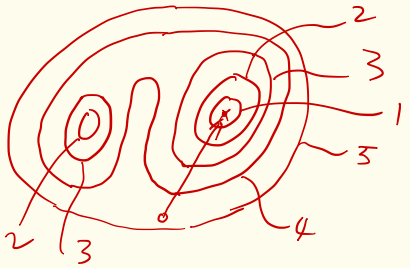
$$\min_{\beta_1} \sum_{i=1}^{n} (y_i - \beta_1 x_i)^2$$

$$\frac{\partial J}{\partial \beta_1} = \sum 2(y_i - \beta_1 x_i)\cdot(-x_i) = \Big|_{\beta_1=\hat{\beta}_1} 0 \quad \Rightarrow \sum [y_i - \hat{\beta}_1 x_i] x_i = 0$$

Optimization 101:

Goal $\min_{\underset{\sim}{x}} f(\underset{\sim}{x})$

f not necessarily convex



Approaches: 1. gradient / steepest descent (1^{st}-order)

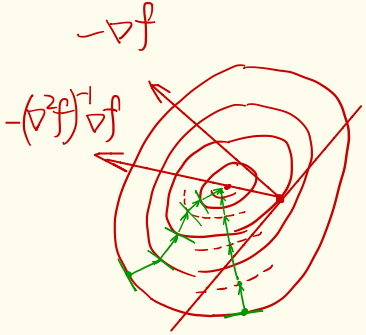
2. Newton's method (2^{nd}-order)

3. Quasi-Newton (1^{st}-order) "BFGS"

Algorithm: start w/ $x^{(0)}$

for $k = 0, 1, 2, \ldots$

compute $g^{(k)} = \nabla f(x^{(k)})$ ← gradient (vector)



if steepest == 1

$$d^{(k)} = -g^{(k)}$$

elseif Newton == 1

$$H^{(k)} = \nabla^2 f(x^{(k)}) \leftarrow \text{Hessian (matrix)}$$

$$d^{(k)} = -H^{(k)-1} g^{(k)}$$

Update: $x^{(k+1)} = x^{(k)} + \alpha_k \cdot d^{(k)}$ (α_k: step size)

Stop when $\|\nabla f(x^{(k)})\| < \varepsilon$

$$g = \nabla_\beta \ell(\beta) = \sum_{i=1}^{n} \left(y_i - p(x_i; \beta) \right) x_i = X^T (y - p)$$

$$H = \nabla^2_\beta \ell(\beta) = - \sum_{i=1}^{n} p(x_i) \left[1 - p(x_i) \right] x_i x_i^T = - X^T W X$$

where $W = \text{diag}\left(p(x_i)[1 - p(x_i)] , i = 1, \ldots, n \right)$

Iteratively Reweighted LS (IRLS)

① $\beta^{new} \leftarrow \beta^{old} - H^{-1} g$

$$= \beta^{old} + (X^T W X)^{-1} X^T (y - p)$$

$$= (X^T W X)^{-1} X^T W \left[X \beta^{old} + W^{-1} (y - p) \right]$$

$$= (X^T W X)^{-1} X^T W z \quad \overset{\text{def}}{=} z \quad \text{"adjusted response"}$$

$\Leftrightarrow \beta^{new} = \underset{\beta}{\text{argmin}} (z - X\beta)^T W (z - X\beta)$ Weighted LS (WLS).

Prove yourself: $\hat{\beta}_{WLS} = (X^T W X)^{-1} X^T W z$

② Update W, or reweight

Multinomial / Multiclass Logistic Regression

$$\log \frac{\mathbb{P}[Y=1 \mid \underset{\sim}{x}]}{\mathbb{P}[Y=k \mid \underset{\sim}{x}]} = \underset{\sim}{\beta}_1^T \underset{\sim}{x}$$

$$\vdots$$

$$\log \frac{\mathbb{P}[Y=k-1 \mid \underset{\sim}{x}]}{\mathbb{P}[Y=k \mid \underset{\sim}{x}]} = \underset{\sim}{\beta}_{k-1}^T \underset{\sim}{x}$$

$\Big\}$ $(k-1)$ log odds, $Y \in \{1, \dots, k\}$

$$\mathbb{P}[Y=k \mid X=\underset{\sim}{x}] = \frac{\exp(\underset{\sim}{\beta}_k^T \underset{\sim}{x})}{1 + \sum_{l=1}^{k-1} \exp(\underset{\sim}{\beta}_l^T \underset{\sim}{x})}, \quad k=1, \dots, k-1.$$

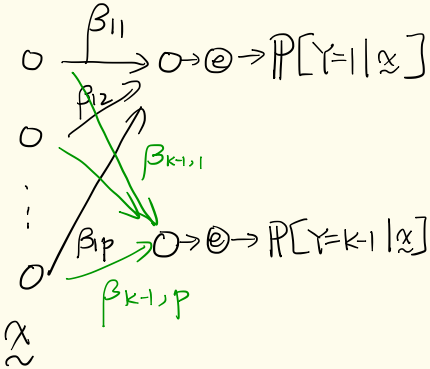
Soft (arg) max function: $\sigma_i(\underset{\sim}{z}) = \frac{e^{\beta z_i}}{\sum_{j=1}^{k} e^{\beta z_j}}$

smooth version of "arg max".

Ex: $K=2$. $\sigma_1(\underset{\sim}{z}) = \frac{e^{\beta z_1}}{e^{\beta z_1} + e^{\beta z_2}} = \frac{1}{1 + e^{\beta(z_2 - z_1)}}$

When β very large

$z_2 > z_1$ leads to $\begin{cases} \sigma_1 = 0 \\ \sigma_2 = 1 \end{cases}$



Generalized Linear Model (GLM)

For linear model, $g(\mu_i) = \beta_0 + \beta_1 x_i$, $g(\cdot)$ is identity

$$\mu_i = \mathbb{E}[Y \mid x_i]$$

For logistic reg, $g(\mu_i) = \beta_0 + \beta_1 x_i$, $g(u) = \log \frac{u}{1-u}$

For Poisson dist, $g(\mu_i) = \beta_0 + \beta_1 x_i$, $g(u) = \log(u)$

$Y_i \in \{0, 1, 2, \dots\}$

↑ link func.

Exponential Family Distributions:

$$f(y;\theta) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right], \text{ where } \phi \text{ is fixed}$$

↑ natural param

$a(\phi)$ ↑ scale param

Bernoulli: $\mathbb{P}[Y=y] = p^y(1-p)^{1-y} = \exp[y\ln p + (1-y)\ln(1-p)]$

$= \exp\left[y \underbrace{\ln\frac{p}{1-p}}_{\theta} + \ln(1-p)\right]$

Poisson: $\mathbb{P}[Y=y] = \frac{\mu^y e^{-\mu}}{y!} = \exp[y \underbrace{\ln\mu}_{\theta} - \mu - \ln(y!)]$

Gaussian: $f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2}$ (σ^2 fixed, known)

$= \exp\left[-\ln(\sqrt{2\pi}\sigma) - \frac{(y-\mu)^2}{2\sigma^2}\right]$

$= \exp\left[\frac{y\overbrace{\mu}^{\theta} - \cdots}{\sigma^2} + c(y,\phi)\right]$

σ^2: scale param

$g(\cdot)$ such selected are called "canonical link function".

Exp family dist can use the same method, i.e., IRLS to find the best params:

$$\log \mathcal{L}(\beta) = \log\left(\prod_{i=1}^{n} f(y_i;\beta)\right)$$

$$= \sum_{i=1}^{n}\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right]$$

$$\frac{\partial}{\partial \beta_i}\log\mathcal{L}(\beta) = 0 \quad \text{for all } i.$$

Details about IRLS for GLM, see McCulloh 5.4.