Linear models for classification

Ref : Sec 4.2 of Bishop's book

Probabilistic Generative model:

Model the class-conditional densities & prior

Then, compute the posterior

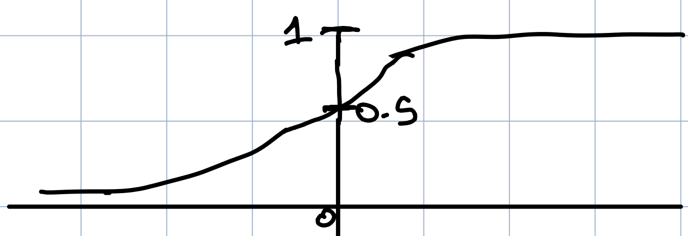
Two-class classification problem:-

$$q_1(x) = \frac{f_0(x)\, p_0}{f_0(x)\, p_0 + f_1(x)\, p_1}$$

$$= \frac{1}{1 + \dfrac{f_1(x)\, p_1}{f_0(x)\, p_0}}$$

$$q_1(x) = \frac{1}{1 + \exp(-a)}, \quad a = \log \frac{f_0(x)\, p_0}{f_1(x)\, p_1}$$

$$= \sigma(a)$$



Note: ① $\dfrac{d\sigma(a)}{da} = \sigma(a)\,(1 - \sigma(a))$ ← check this

② $\sigma(-a) = 1 - \sigma(a)$

③ Logit function: $a = \log\left(\dfrac{\sigma}{1-\sigma}\right)$

Classification task is easier if "a" has a simple functional form, such as "linear".

<u>Example</u>:- Class conditional densities are Gaussian with the same covariance matrix

$$f_i(x) = \frac{1}{(2\pi)^{d/2}} \frac{1}{\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma^{-1} (x-\mu_i)\right),$$

$i = 0, 1$

Calculating the sigmoid parameter "a" in this case:

$$a = \log \frac{f_0(x)}{f_1(x)} + \log \frac{P_0}{P_1}$$

$$\log \frac{f_0(x)}{f_1(x)} = -\frac{1}{2}(x-\mu_0)^T \Sigma^{-1} (x-\mu_0) + \frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)$$

$$= x^T \Sigma^{-1}(\mu_0 - \mu_1) - \frac{1}{2}\mu_0^T \Sigma^{-1}\mu_0 + \frac{1}{2}\mu_1^T \Sigma^{-1}\mu_1$$

$$= w^T x + w_0, \text{ where}$$

$$w = \Sigma^{-1}(\mu_0 - \mu_1), \text{ and}$$

$$w_0 = -\frac{1}{2}\mu_0^T \Sigma^{-1}\mu_0 + \frac{1}{2}\mu_1^T \Sigma^{-1}\mu_1$$

So, $\quad q_i(x) = \sigma(w^T x + w_0)$

Max-likelihood estimation:

Dataset: $\{(x_i, y_i), i = 1 \dots n\}$

y_i: class labels $\in \{0,1\}$

To estimate: $p_0, \mu_0, \mu_1, \Sigma$

For a data point from class-0,

$$f(x_i, \text{class-}0) = p_0 f_0(x_i)$$

Similarly, $f(x_i, \text{class-}1) = (1-p_0) f_1(x_i)$

Likelihood function:

$$\mathcal{L}(p_0, \mu_0, \mu_1, \Sigma) = \prod_{i=1}^{n} \big(p_0 \underbrace{f_0(x_i)}_{N(\mu_0, \Sigma)}\big)^{y_i} \big((1-p_0) \underbrace{f_1(x_i)}_{N(\mu_1, \Sigma)}\big)^{1-y_i}$$

<u>Estimate p_0:</u>

log-likelihood: $\ell(p_0, \mu_0, \mu_1, \Sigma) = \log \mathcal{L}(p_0, \mu_0, \mu_1, \Sigma)$

In $\ell(p_0, \mu_0, \mu_1, \Sigma)$, the terms that depend on p_0 are as follows:

$$\nabla_{p_0} \left(\sum_{i=1}^{n} y_i \log p_0 + (1-y_i) \log (1-p_0) \right) = 0$$

$$P_0 = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{n_0}{n_0+n_1}, \text{ where}$$

$n_0 =$ # of points in class-0

$n_1 =$ # of points in class-1

Estimate μ_0:

$$\nabla_{\mu_0}\left(\ell(P_0, \mu_0, \mu_1, \Sigma)\right) = 0$$

$$\nabla_{\mu_0}\left(-\frac{1}{2}\sum_{i=1}^{n} y_i (x_i - \mu_0)^T \Sigma^{-1} (x_i - \mu_0)\right) = 0$$

$$\mu_0 = \frac{1}{n_0}\sum_{i=1}^{n} y_i \, x_i$$

$\boxed{y_i = 1 \text{ if } x_i \in \text{class-0.}}$

Similarly, $\mu_1 = \frac{1}{n_1}\sum_{i=1}^{n}(1-y_i)x_i$

Estimating the covariance matrix Σ:

The terms involving Σ in the log-likelihood function are

$$\left(-\frac{1}{2}\sum_{i=1}^{n} y_i \log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n} y_i (x_i-\mu_0)^T \Sigma^{-1}(x_i-\mu_0)\right.$$

$$\left.-\frac{1}{2}\sum_{i=1}^{n}(1-y_i)\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}(1-y_i)(x_i-\mu_1)^T\Sigma^{-1}(x_i-\mu_1)\right)$$

$$= -\frac{n}{2}\log|\Sigma| - \frac{n}{2}\operatorname{Tr}\left(\Sigma^{-1}S\right), \text{ where}$$

$$S = \frac{n_0}{n} S_0 + \frac{n_1}{n} S_1, \quad \text{——— (1)}$$

$$S_0 = \frac{1}{n_0} \sum_{i \in \text{class-}0} (x_i - \mu_0)(x_i - \mu_0)^T$$

$$S_1 = \frac{1}{n_1} \sum_{i \in \text{class-}1} (x_i - \mu_1)(x_i - \mu_1)^T$$

The ML-estimate of Σ is "S"

Note: In the univariate case, it is easy to infer the S in (1) is the ML estimate.

$$+\frac{n}{2S} = -\frac{1}{2} \sum y_i (x_i - \mu_0)^2 \times \left(-\frac{1}{S^2}\right)$$

$$- \frac{1}{2} \sum (1-y_i)(x_i - \mu_1)^2 \times \left(-\frac{1}{S^2}\right)$$

$$n = \frac{1}{S}\left[\sum y_i (x_i - \mu_0)^2 + \sum (1-y_i)(x_i - \mu_1)^2\right]$$

$$S = \left[\frac{n_0}{n} \times \frac{1}{n_0} \sum_{i \in \text{class-}0} (x_i - \mu_0)^2 + \frac{n_1}{n} \times \frac{1}{n_1} \sum_{i \in \text{class-}1} (x_i - \mu_1)^2\right]$$

$$= \frac{n_0}{n} S_0 + \frac{n_1}{n} S_1, \quad \text{where } S_0 = \frac{1}{n_0} \sum_{i \in \text{class-}0} (x_i - \mu_0)^2$$

$$S_1 = \frac{1}{n_1} \sum_{i \in \text{class-}1} (x_i - \mu_1)^2$$

Probabilistic discriminative models

We have seen that

(i) For a 2-class classification problem, the posterior probability of each class can be written as a logistic sigmoid acting on a linear function of the feature vector.

ML approach can be adopted to estimate density parameters.

(ii) Alternatively, we the functional form of the generalized linear model & determine the parameters (weights) directly.

$$q_0(x) = \sigma(w^T x), \quad q_1(x) = 1 - q_0(x)$$

ML-approach:- $x \in \mathbb{R}^d$

Take Gaussian case,

parameters to estimate are

(i) $\mu_0, \mu_1 \Rightarrow 2d$ parameters

(ii) Σ (symmetric) $\Rightarrow \frac{d(d+1)}{2}$ parameters

(iii) prior prob p_0

Overall ML approach requires $O(d^2)$ parameters to estimate

If we use the generalized linear model & estimate parameters directly, then the complexity of the task reduces to estimating $O(d)$ # of parameters.

Logistic regression:

Dataset $\{(x_i, y_i), i = 1 \ldots n\}$ $y_i \in \{0,1\}$

Likelihood $$\mathcal{L}(\omega) = \prod_{i=1}^{n} \left(\sigma(\omega^T x_i)\right)^{y_i} \left(1 - \sigma(\omega^T x_i)\right)^{1-y_i}$$

Cross-entropy loss $$l(\omega) = -\log \mathcal{L}(\omega)$$

$$l(\omega) = -\sum_{i=1}^{n} y_i \log(\sigma(\omega^T x_i)) + (1-y_i)\log(1-\sigma(\omega^T x_i))$$

$$\nabla l(\omega) = -\sum_{i=1}^{n} \frac{y_i}{\sigma\cancel{(\omega^T x_i)}} \sigma\cancel{(\omega^T x_i)}(1-\sigma(\omega^T x_i))\, x_i$$
$$- \sum_{i=1}^{n} \frac{(1-y_i)}{(1\cancel{-\sigma(\omega^T x_i)})} \sigma(\omega^T x_i)\,(1\cancel{-\sigma(\omega^T x_i)})x_i$$

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

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

The gradient descent update

$$w_{k+1} = w_k - \alpha_k \left[\sigma(w_k^T x(k)) - y(k) \right] x(k)$$

where $(x(k), y(k))$ could be chosen such that we cycle through the dataset.

<u>Claim:</u> l is strictly convex

The Hessian of $l(\cdot)$ is

let $z_i = \sigma(w^T x_i)$

$$H(w) = \nabla^2 l(w)$$
$$= \sum_{i=1}^{n} z_i (1-z_i) x_i x_i^T$$

depends on w

$$H(w) = A^T R(w) A$$

where R is a diagonal matrix with entries $z_i(1-z_i)$, &

A is the usual feature matrix, i.e., with rows x_i^T

The claim of $l(\cdot)$ being strictly convex can be inferred if H is p.d., i.e.,

$$u^T H u > 0 \quad \forall u \neq 0$$

$$\Leftrightarrow \quad u^T A^T R A u > 0$$

$$LHS = u^T \left[\sum_{i=1}^{n} z_i(1-z_i)\, x_i x_i^T \right] u$$

$z_i \in (0,1)$ so, $z_i(1-z_i) > 0$

$$= \sum_{i=1}^{n} z_i(1-z_i)\, u^T x_i x_i^T u$$

$$= \sum_{i=1}^{n} z_i(1-z_i)\, (x_i^T u)^T x_i^T u$$

$$= \sum_{i=1}^{n} z_i(1-z_i)\, (x_i^T u)^2$$

why? > 0 —— $(**)$

$z_i(1-z_i) > 0$, & assuming A is full column rank, $x_i^T u \neq 0 \ \forall i$ [why?]

& hence $(**)$ holds $\Rightarrow$ l is strictly convex.

Background: Newton's method

$$\min_{\omega} \ell(\omega), \quad \ell \text{ is strictly convex}$$
$$\Rightarrow \quad H(\omega) = \nabla^2 \ell(\omega) \text{ is p.d.}$$

Incremental update: $\omega_{k+1} = \omega_k - \Delta\omega_k$

Quadratic approximation to ℓ:

$$(*) \rightarrow \hat{\ell}(\omega) = \ell(\omega_k) + \nabla \ell(\omega_k)^T(\omega - \omega_k) + \frac{1}{2}(\omega - \omega_k)^T H(\omega_k)(\omega - \omega_k)$$

Claim: $\hat{\ell}$ is minimized at $\left(\omega_k - H(\omega_k)^{-1}\nabla \ell(\omega_k)\right)$

To see this, rewrite $\hat{\ell}$ as follows:

$$\hat{\ell}(\omega) = \omega^T C \omega + d^T \omega + h, \text{ where}$$

$$C = \frac{1}{2} H(\omega_k), \quad d = \nabla \ell(\omega_k) - H(\omega_k)\omega_k,$$
$$h = \ell(\omega_k) - \nabla \ell(\omega_k)^T \omega_k + \frac{1}{2}\omega_k^T H(\omega_k)\omega_k$$

From the rewrite, it is easy to see that the minimizer of $\hat{\ell}$ is

$$\omega_{min} = -\frac{1}{2} C^{-1} d = \omega_k - H(\omega_k)^{-1}\nabla \ell(\omega_k)$$

So, the Newton's method update rule is

$$w_{k+1} = w_k - H(w_k)^{-1} \nabla l(w_k)$$

l is strictly convex

Ref: Convex optimization by Boyd & Vandenberghe

H.w.: Work out Newton's method for linear regression

Back to logistic regression

$$l(w) = -\log \mathcal{L}(w)$$

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

$$\nabla l(w) = \sum_{i=1}^{n} (\sigma(w^T x_i) - y_i)\, x_i = A^T(z-y)$$

$$H(w) = \nabla^2 l(w) = A^T R(w) A$$

A is the feature matrix with rows x_i^T &

R is a diagonal matrix with entries

$$z_i(1-z_i), \quad z_i = \sigma(w^T x_i) \in (0,1)$$

$z = (z_1 \cdots z_n)$, $y = (y_1 \cdots y_n)$

$$w_{k+1} = w_k - H(w_k)^{-1} \nabla \ell(w_k)$$

$$= w_k - \left(A^T R(w_k) A\right)^{-1} \left(A^T (z-y)\right)$$

$$= \left(A^T R(w_k) A\right)^{-1} \left(A^T R(w_k) A w_k - A^T (z-y)\right)$$

$$w_{k+1} = \left(A^T R(w_k) A\right)^{-1} A^T R(w_k)\, b_k, \quad \text{--- } (\ast\ast\ast)$$

where $b_k = A w_k - R(w_k)^{-1} (z-y)$

Note: The update in $(\ast\ast\ast)$ resembles the solution to a weighted least squares (WLS)

$$\left(\text{WLS-objective} = \frac{1}{2} \sum_{i=1}^{n} \beta_i \left(y_i - w^T x_i\right)^2 \right)$$

H.w. Calculate the solution to the problem above.

The update in $(\ast\ast\ast)$ is the "Iterative reweighted least squares"