Bayes Classifiers

At training time create model:

- Labels: $P(y=k) = \pi_k$
- Features: $P(\underline{x} \mid y=k)$ e.g. $\mathcal{N}(\underline{x}; \mu^{(k)}, \Sigma^{(k)})$ or discrete dist.

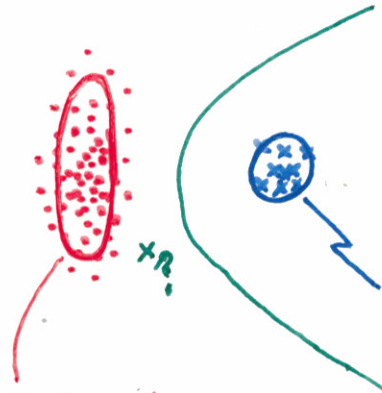
→ Models $\underline{x}$ and y:

$$P(y, \underline{x}) = p(y)\, p(\underline{x} \mid y)$$

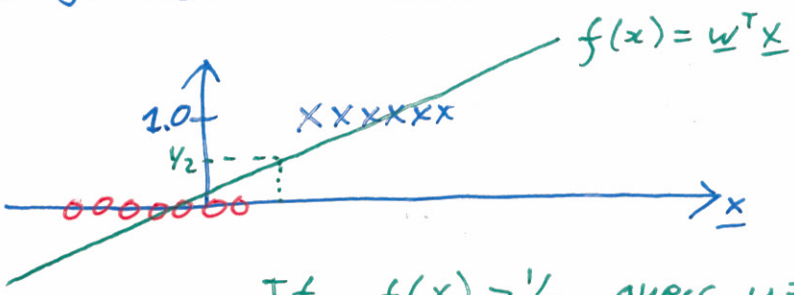
At test time use Bayes' rule:

$$p(y \mid \underline{x}) \propto p(y, \underline{x})$$

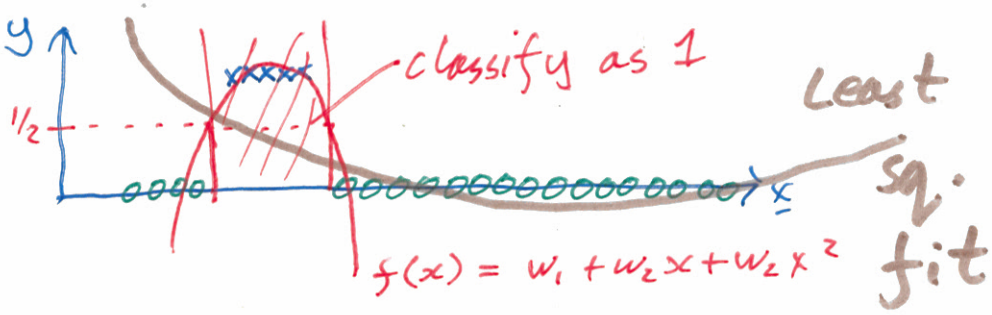
$$p(y=k \mid \underline{x}) = \frac{p(y=k, \underline{x})}{\sum_{k'} p(y=k', \underline{x})}$$

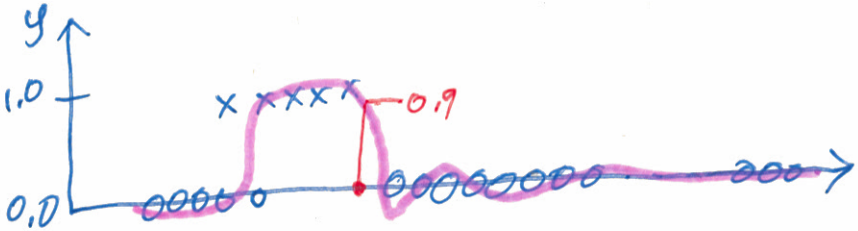

Labelled train points
• and × class 1
class 0
Contour of $N(\underline{x}; \mu^{(1)}, \Sigma^{(1)})$
Contour of
$N(\underline{x}; \mu^{(0)}, \Sigma^{(0)})$
Along this line,
$N(\underline{x}, \mu^{(1)}, \Sigma^{(1)})\pi_1 = N(\underline{x}, \mu^{(0)}, \Sigma^{(0)})\pi_0$
To right, class 1, more probable
On line, $p(y=1|\underline{x}) = 1/2$

Regressing on the labels



If $f(\underline{x}) > 1/2$ guess $y = 1$





Radial basis f^n fit.

If minimize square loss

Minimize $\underset{p(y|\underline{x})}{\mathbb{E}}[(y-f)^2]$ at some location $\underline{x}$

$= p_1(1-f)^2 + (1-p_1)(0-f)^2$

p_1: $p(y=1|\underline{x})$; $(1-p_1)$: Two classes, $p_0 + p_1 = 1$

$\text{cost} = f^2 - 2p_1 f + p_1$

$\frac{\partial \text{cost}}{\partial f} = 2f - 2p_1 = 0$ at optimum

$f = p_1$

Multiple classes

$y = \{1, 2, 3, 4, \ldots, 10\}$

"sport" "crime" "romance"

If $f(\underline{x}) = \underline{w}^T \underline{x}$

$f(\underline{x}^{(1)}) \approx 1 \Rightarrow$ "sport"

$f(\underline{x}^{(2)}) \approx 3 \Rightarrow$ "romance"

$f\left(\frac{\underline{x}^{(1)} + \underline{x}^{(2)}}{2}\right) \approx 2 \Rightarrow$ "crime"

One-hot encoding, One-of-k encoding

Create vector output

$$\underline{y}^{(n)} = [0\ 0\ 0\ ..\ 0\ 1\ 0\ 0\ 0]^T$$

↑ k^{th} position

If example n is in class k.

Puzzle:

R comes with a f^n to do one-hot encoding for linear regression

If you have K classes

it returns a $k-1$ dim. vector

If you have class K it return $k-1$ zeros.

Gradients for least squares

X: $N \times D$ design matrix of all inputs

Residuals $\underline{r} = \underline{y} - X\underline{w}$

Cost $\underline{r}^T\underline{r} = (\underline{y} - X\underline{w})^T(\underline{y} - X\underline{w})$

$= \underline{y}^T\underline{y} - 2\underline{w}^T X^T \underline{y} + \underline{w}^T X^T X \underline{w}$

"Gradient"

vector of partial derivatives

$$\nabla_{\underline{w}} [\underline{r}^T\underline{r}] = -2X^T\underline{y} + 2X^TX\underline{w}$$

Gradient descent:

$$\underline{w} \leftarrow \underline{w} - \eta \nabla_{\underline{w}} [\underline{r}^T\underline{r}]$$

η: step-size, small.