Bayes Classifiers

Training time

Joint model $p(y, \underline{x}) = p(y)\,p(\underline{x}|y)$

$$p(y=k) = \pi_k \approx \frac{\#\text{ k labels}}{N}$$

$$p(\underline{x}|y=k) \;\ldots\; \text{eg } \mathcal{N}(\underline{x}; \underbrace{\mu^{(k)}, \Sigma^{(k)}}_{\text{Mean \& cov of } \underline{x}\text{'s in class } k})$$

Not Bayesian

Assuming we know all parameters.

Mean & cov of $\underline{x}$'s in class k

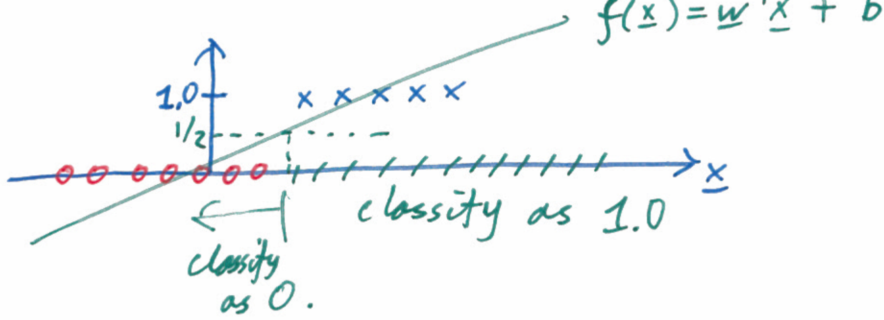
Naive Bayes $p(\underline{x}|y=k) = \prod_d \underbrace{p(x_d|y=k)}_{\text{Univariate Gaussian, Discrete, ...}}$

Test time

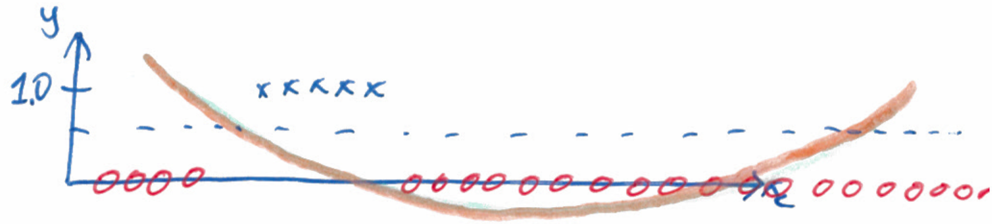
$p(y|\underline{x}) \propto p(y, \underline{x})$ (Bayes' Rule)

$$y_{guess} = \arg\max_k \underbrace{p(y=k, \underline{x})}_{\text{"goodness"}}$$

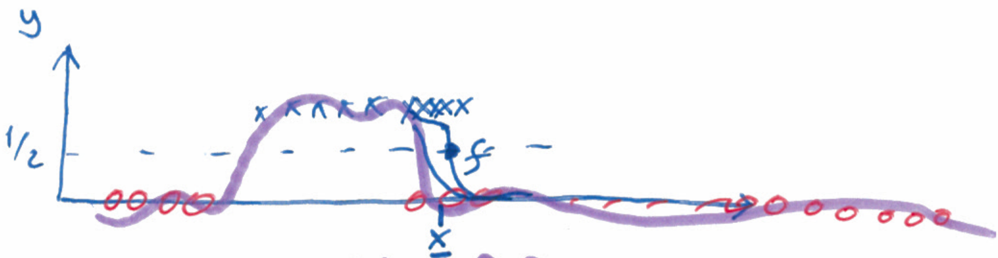
Regression to labels



If $f(x) > 1/2$, guess $y=1$



$$f(x) = w_1 + w_2 x + w_3 x^2$$



Fit with RBFs

If minimize square loss?

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

Cost

$$= \underbrace{p_1}_{p(y=1|\underline{x})}(1-f)^2 + \underbrace{(1-p_1)}_{p(y=0|\underline{x})}(0-f)^2$$

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

$$= f^2(\cancel{p_1 - p_1}\ 1) - 2p_1 f + p_1$$

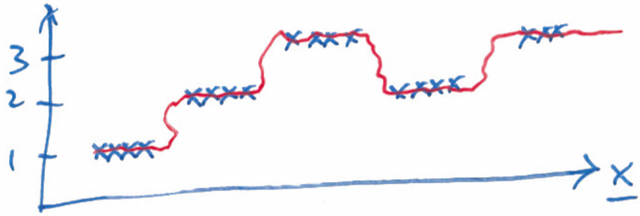
$$\frac{\partial \text{cost}}{\partial f} = 2f - 2p_1 = 0 \text{ at optimum}$$

$$\boxed{f = p_1}$$

Multiple classes

$y \in \{1, 2, 3, 4 \cdots 10\}$

"sport" "crime "romance"



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

Maybe replace $\underline{x}$ with $\phi(\underline{x})$

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

Vector output

Is in k^{th} position

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

$K \times 1$ vector

If we have K classes

If n^{th} example is in class k

Fit K functions, one for each bit y_k

This pre-processing step is also useful for input features

$$x_d \in \{\text{"red"}, \text{"green"}, \text{"blue"}\}$$
$$\{1,\ 2,\ 3\}$$

3 features

red → 1 0 0

green → 0 1 0

blue → 0 0 1

R doesn't create this column.

Puzzle: In R you can do one-hot encoding

red → 1 0

green → 0 1

blue → 0 0

Gradients for least squares cost

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

$\uparrow$ $N \times 1$ vector of scalar labels

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

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

"Gradient" vector of partial derivatives:

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

Scratch working

$$\nabla_{\underline{w}} \left[\underline{w}^T \underline{h} \right] = \begin{bmatrix} \frac{\partial \underline{w}^T \underline{h}}{\partial w_1} \\ \frac{\partial \underline{w}^T \underline{h}}{\partial w_2} \\ \vdots \\ \frac{\partial \underline{w}^T h}{\partial w_D} \end{bmatrix} = \underline{h}$$

$$\frac{\partial \underline{w}^T \underline{h}}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_j w_j h_j$$

$$= \frac{\partial}{\partial w_i} \left(w_1 h_1 + w_2 h_2 + \dots \underline{w_i h_i} + \dots w_D h_D \right)$$

$$= h_i$$

Matrix Cookbook.