MLPR: where we've been, where we're going

Representing functions

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

fitting to data, regularizing and evaluating models

If you can't do and explain this stuff, it needs review

Some statistics + probability

(Co)variance, standard errors, expectations, Gaussians

(≠ deviation!)

Have you done w0f question

Classification (y is discrete)

- Bayes classifiers, especially with Gaussians (more today)
- Linear regression (this lecture)
- Later: beyond linear / Gaussian models

Arno's lectures (start tomorrow)

"What more can we do with linear / Gaussian models?"

→ Give uncertainty that increases when extrapolating

→ "infinite dimensional $\underline{\phi}(\underline{x})$", Gaussian processes

"Bayesian

Iain's return (weeks 6 – 10)

Nonlinear models, non-Gaussian outputs,

fitting these models, approximate Bayesian inferenc

(+ whatever else we have time for)

Bayes Classifiers

Training time:

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

$P(y=k) = \pi_k \approx \frac{\#\ k \text{ labels}}{N}$ [Or set by domain knowledge]

$p(\underline{x} \mid y=k) \ldots$ eg $N(\underline{x};\ \underline{\mu}^{(k)}, \Sigma^{(k)})$

This estimation is not Bayesian { Set to mean & cov of $\underline{x}$'s in class k

Naive Bayes $p(\underline{x} \mid y=k) = \prod_d p(x_d \mid y=k)$

Univariate Gaussian, discrete, ...

Test time:

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

$$y_{guess} = \arg\max_k\ p(y=k, \underline{x})$$

a score for each class

$N(\underline{x}; \underline{\mu}^{(0)}, \Sigma^{(0)})$

quadric decision boundary

$N(\underline{x}; \underline{\mu}^{(1)}, \Sigma^{(1)})$

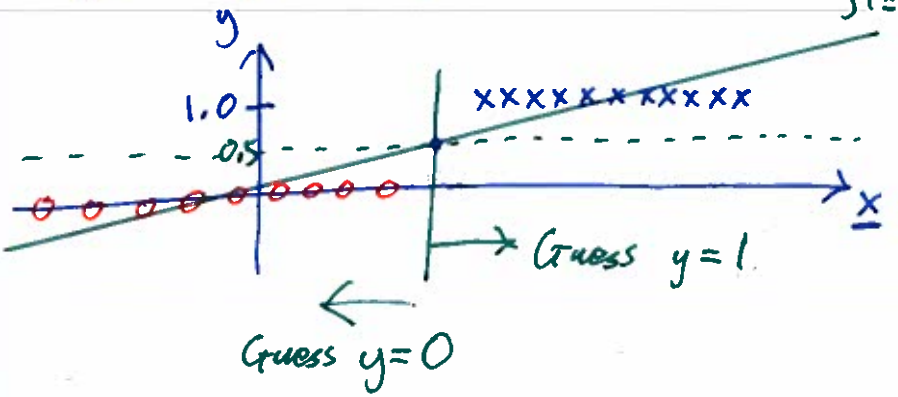
Maybe not what you want

Decision boundary $p(y|\underline{x}) = 0.5$

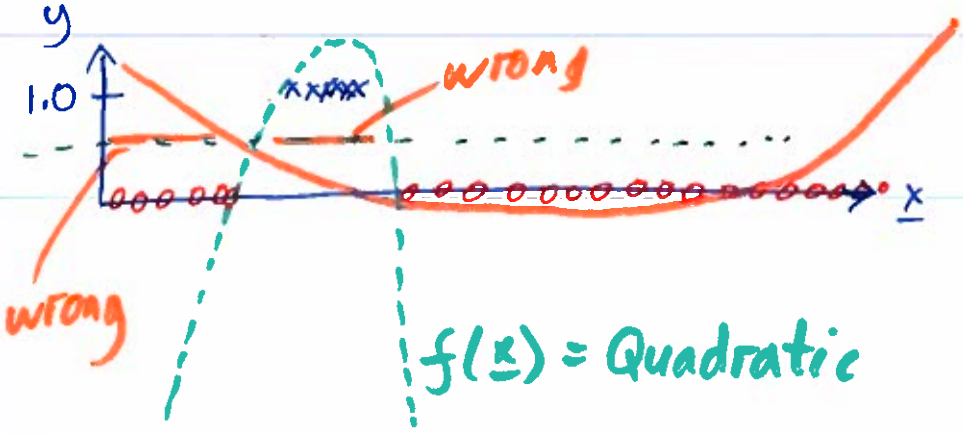
$$\pi_0 N(\underline{x}; \mu^{(0)}, \Sigma^{(0)}) = \pi_1 N(\underline{x}; \mu^{(1)}, \Sigma^{(1)})$$

Regression for classification

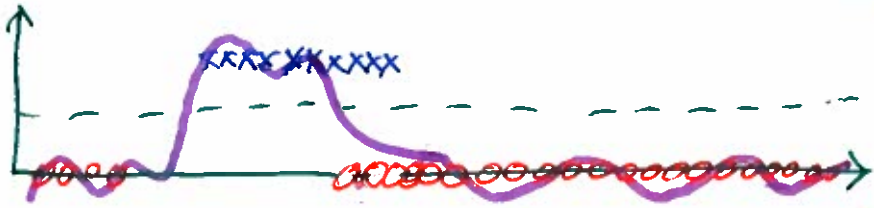
$f(\underline{x}) = \underline{w}^T \underline{x} + b$



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



$f(\underline{x})$ = Quadratic

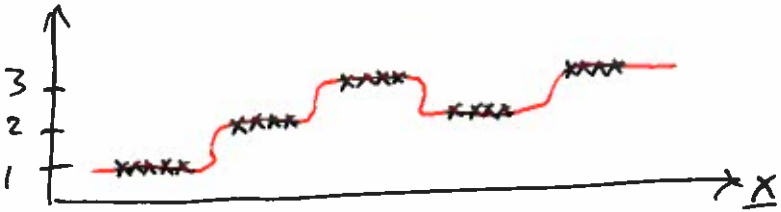


↳ fit with RBFs

Multiple classes

$y = \{1, 2, 3, \ldots 103\}$

"sport" "crime" "technology"



$f(\underline{x}) = \underline{w}^T \underline{x}$ [Replace $\underline{x}$ with $\underline{\phi}(\underline{x})$ if you like]

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

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

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

What happens if we minimize square loss?

What's the best function theoretically?

At a particular location $\underline{x}$, let $f(\underline{x}) = f$

$$\text{cost} = \mathbb{E}_{p(y|\underline{x})}\left[(y-f)^2\right]$$

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

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

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

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

$$\boxed{f = p_1}$$

One-hot encoding: One-of-k encoding
One-of-M encoding
...

Vector output

$y^{(n)} = [000...010...00]^T$ ($k \times 1$)
↑ k^{th} position
for k classes
If nth example is in class k

Easiest thing:
Fit k separate functions, one for each bit y_k
Predict class where $f_k(\underline{x})$ biggest.

Pre-processing also useful for features $\underline{x}$

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

3 features
red → 1 0 0
green → 0 1 0
blue → 0 0 1

Don't need. Libraries in R don't create this column.

What if features are ordered?

E.g. Movie ratings