$$f(x) = w^T x + w_0$$

including excellent pen skills!

Linear Classifiers

_Marco Loog

Past, Present, …

_Yesterday, covered regression with linear model

_Today we get back to classifiers

- Notably, linear classifiers…
- Which ones did we see already?

_Meanwhile work towards framework that captures setup of many classifiers

More Specifically

_Covering

Gaussian-based linear classifiers [recap, 2-class case]
Logistic regression / classifier
Linear regression classifier
The perceptron

Encore : that general framework…

Reminder : Losses of Interest

Classification aims to minimize expected error rate

$$\sum_y \int [f(x) \neq y] p(x, y) dx$$

Regression aims to minimize expected squared loss

Other losses possible [any ideas?]

$$\int (f(x) - y)^2 p(x, y) dx dy$$

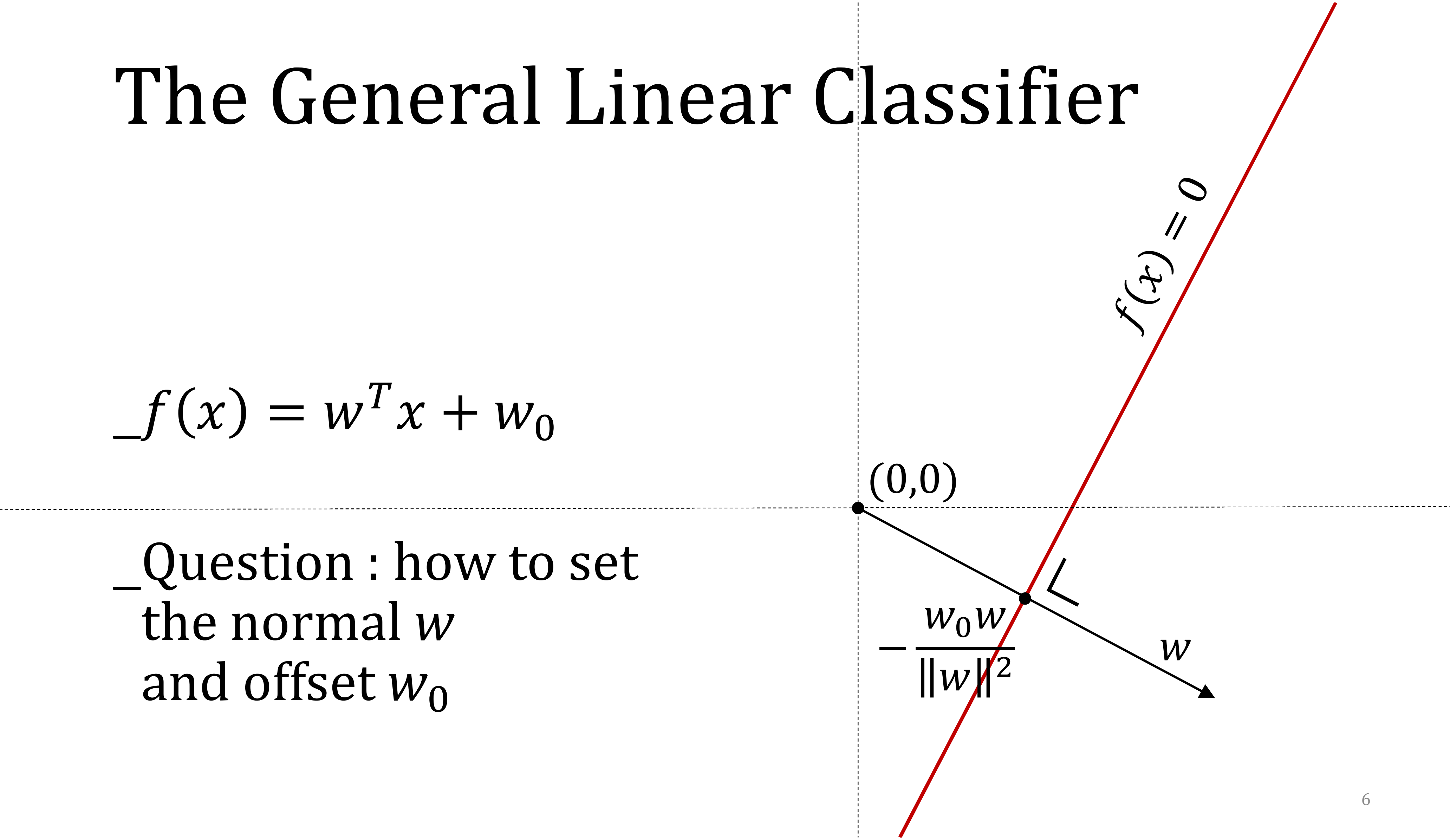
_We do not know p

_We need to assume a model for f

The General Linear Classifier

_$f(x) = w^T x + w_0$

_Question : how to set the normal w and offset w_0



LDA & NMC

Gaussian-based Classifiers

_Assumed model : Gaussian class conditionals

With equal covariance matrices

_Define $f(x) = \log p(y_1|x) - \log p(y_2|x)$

If > 0 assign to class 1

_Then $f(x) = w^T x - w_0$ with $w = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ and some unwieldy expression for w_0

Further Simplifying Assumptions…

_We have $f(x) = w^T x - w_0$ with $w = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ and some unwieldy expression for w_0

_Assuming covariance I and prior equal, we find

$$w = (\hat{\mu}_2 - \hat{\mu}_1), w_0 = \|\hat{\mu}_2\|^2 - \|\hat{\mu}_1\|^2,$$

and can take $f(x) = \|\hat{\mu}_2 - x\|^2 - \|\hat{\mu}_1 - x\|^2$

Logistic Regression

Let’s Assume Linear “Logit”

_Take $f(x) = \log p(y_1|x) - \log p(y_2|x)$ and assumed class-conditionals Gaussian

Result : a linear classifier if covariances are equal

_An alternative : immediately assume

$$\log p(y_1|x) - \log p(y_2|x) = w^T x + w_0 = f(x)$$

No class conditionals; just restricts posteriors

$$\log \frac{p(y_1|x)}{p(y_2|x)} = f(x)$$

_Derive $p(y_1|x)$...

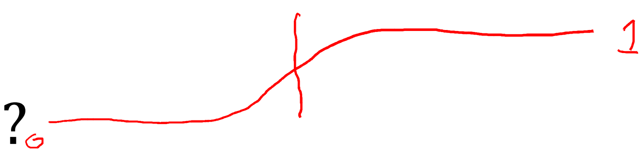
$\log \frac{p_1}{p_2} = f \Rightarrow \frac{p_1}{p_2} = \exp f \Rightarrow p_1 = (1-p_1)\exp f$

$p_1 = \frac{\exp f}{1 + \exp f} = \frac{1}{\exp -f + 1}$

Logistic Regression

_Classifier that takes $p(y_1|x) = \dfrac{1}{\exp(-f(x)) + 1}$

What shape does this have as a function of x?



How do we now find the actual parameters?

[Conditional] Likelihood!

_Maximize [its logarithm]

$\prod_1 p(y_i|x_i) \prod_2 p(y_i|x)$

$$\sum_{\text{all } x \text{ in class } y_1} \log_2\left(\frac{1}{\exp(-f(x))+1}\right) + \sum_{\text{all } x \text{ in class } y_2} \log_2\left(\frac{1}{\exp(f(x))+1}\right)$$

Rewrite into Minimization…

_Identify $y_1 = +1$ and $y_2 = -1$
_Then minimize

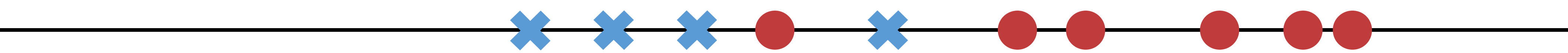
$$\sum_{i=1}^{N} \log_2(\exp(-y_i f(x)) + 1)$$

Fisher & Linear Regression

Linear Classifier by Least Squares?

_Also referred to as Fisher classifier, FLD,…

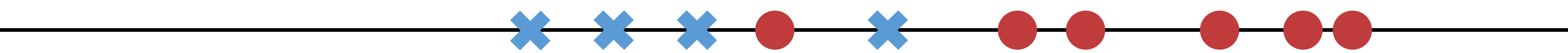
_How to?



Linear Classifier by Least Squares?

_How to?

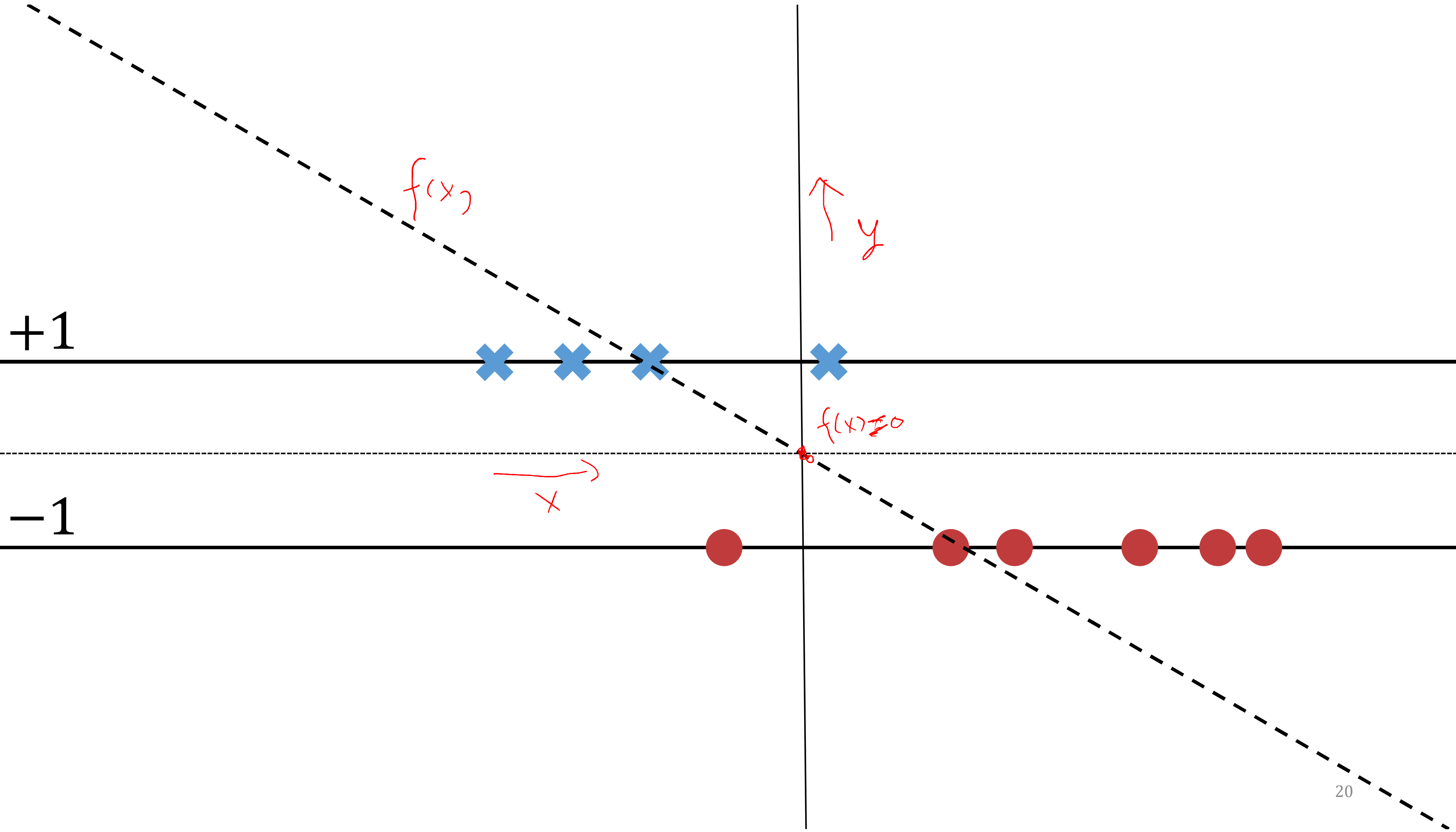
_Again identify $y_1 = +1$ and $y_2 = -1$?

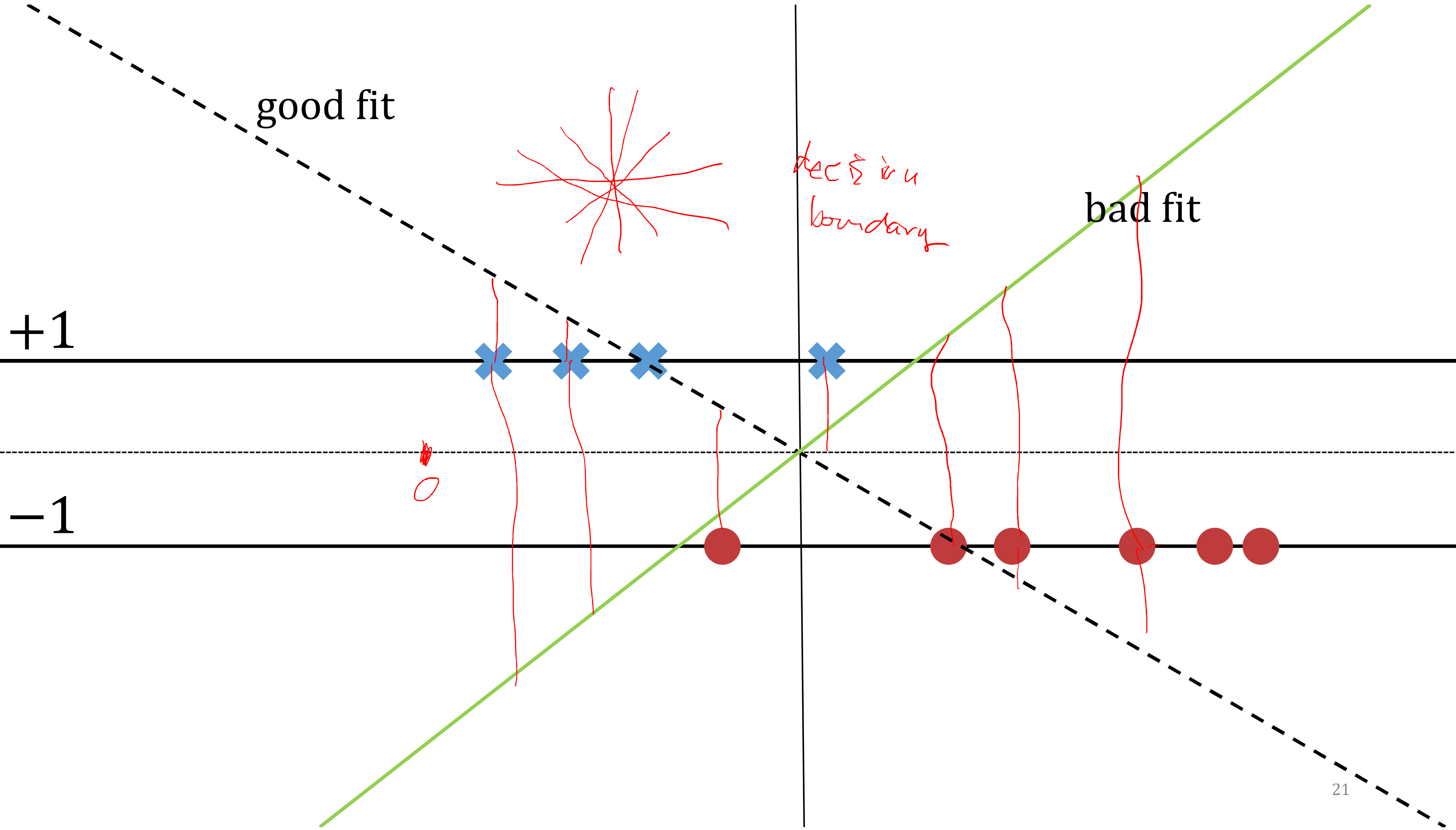


We Get...

$(x_i, y_i) \quad y_i \in \{-1, +1\}$

$$\sum_{i=1}^{N} ?(w^T x_i - y_i)^2$$


+1
−1
f(x)
y
x
f(x)=0


good fit
bad fit
+1
−1
decision boundary
0

General Setup of Fitting a Learner

General Setup of Fitting a Learner

_1) Choose a class of models

Linear functions, Gaussian classes, sigmoidal posteriors, …

_2) Choose a fitting function / loss

Log-likelihood, squared loss, MAP, …

_Sum over individual training elements

_Works for regression and classification

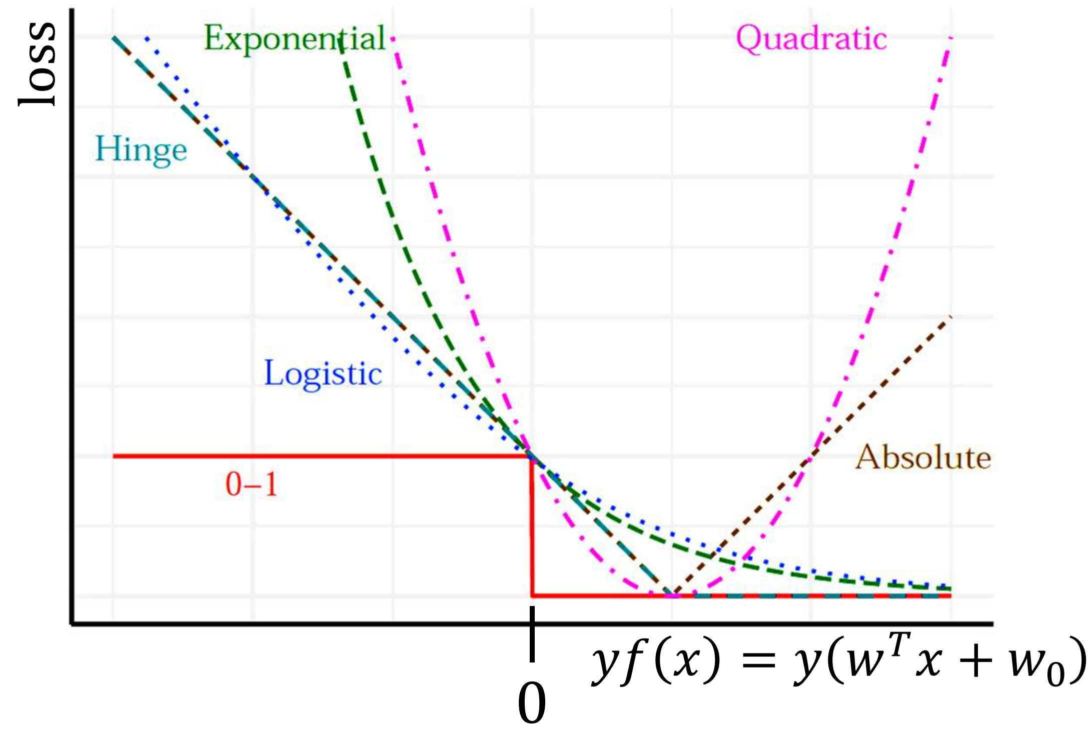
Formulations are Not Unique!

_NMC : spherical Gaussian model + LL
means as model + squared deviation

_Logistic regression : sigmoidal posterior + LL
linear model + logistic loss

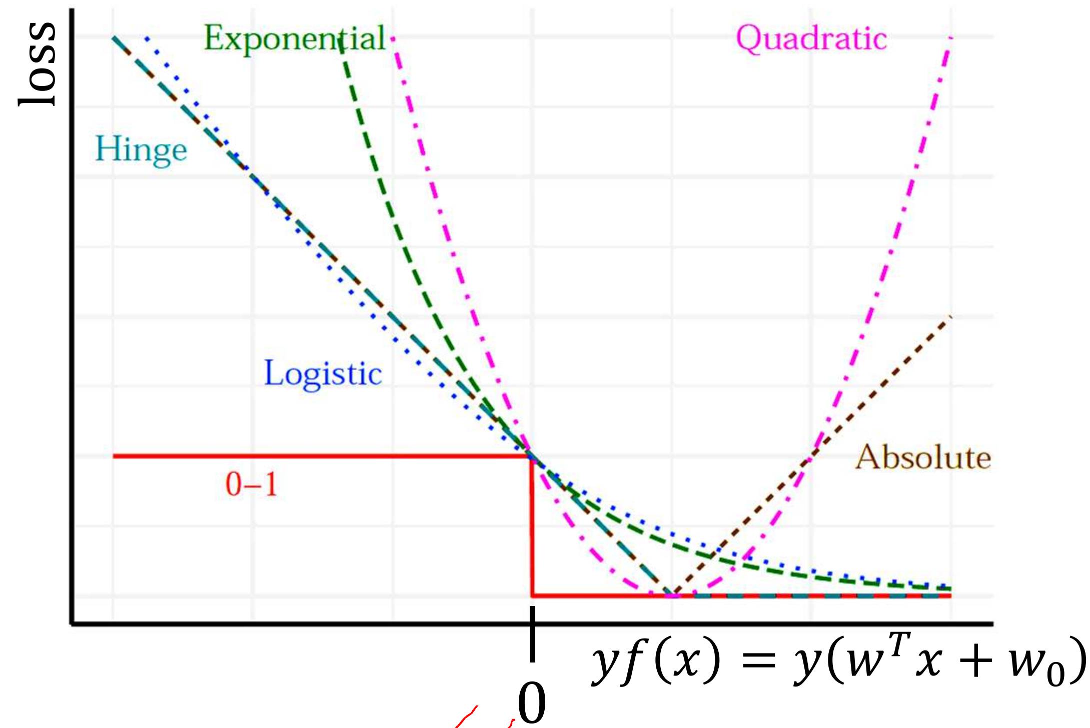
$$\sum_{i=1}^{N} \log_2\left(\boxed{\exp\left(-y\left(\boxed{w^T x + w_0}\right)\right) + 1}\right)$$

Somewhat Special Losses



_$[yf(x) < 0]$
_$(f(x) - y)^2 = (yf(x) - 1)^2$
_$\log_2(\exp(-yf(x)) + 1)$

Hinge and Perceptron



Define $|x|_+ = \frac{|x|+x}{2}$

_Final loss this lecture : "perceptron" loss $|-yf(x)|_+$

_Week 4 : hinge loss $|1 - yf(x)|_+$

The Perceptron

The Perceptron

_Minimizes $\sum|-y_i w^T x_i|_+$

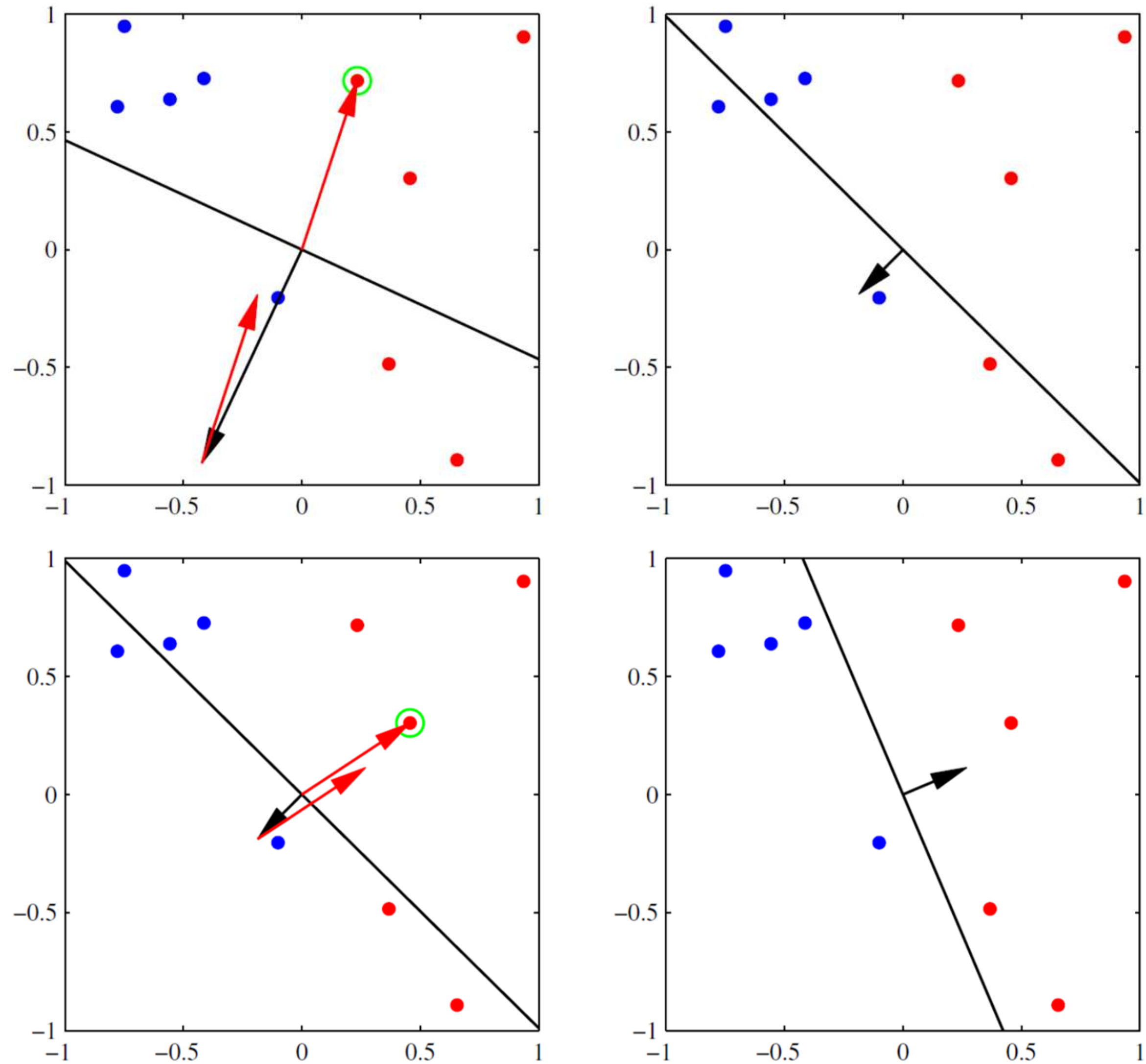
Yes, left out bias for simplicity…

_Way of optimizing = integral part of this learner

Cycle through all training points randomly
Check if random point is correctly classified
If not update $w \longleftarrow w + \eta y x$ [η = learning rate]
Repeat

Classical result : converges in finite steps if data separable

Two Example Iterations



Discussion & Conclusion

Various Linear Classifiers

_LDA, NMC, logistic regression, Fisher linear discriminant, perceptron, hinting at SVMs…

_More importantly?

Many classification and regression functions can be specified by defining 1) a hypothesis class H and 2) a loss or fit function ℓ to check which hypothesis fits best on which data

Strictly speaking, there are two more ingredients… Anybody?

_Note : most classifiers don’t minimize error rate!

Hypothesis-Loss Framework

_Good to realize that many learners have a similar structure [at some level]
  - Look out for [apparent?] exceptions to the rule…

_Can be handy to compare classifiers
  - Same hypothesis space, but different loss used to pick best
  - Same loss but different hypothesis spaces…

Some More Examples

_Linear regression : $H = \{w^T x + w_0 | w \in \mathbb{R}^d, w_0 \in \mathbb{R}\}$ and $\ell(h, x, y) = (h(x) - y)^2$

Or $H = \mathbb{R}^{d+1}$ and $\ell(h, x, y) = \left(h^T \binom{x}{1} - y\right)^2$

_Nearest mean : $H = \mathbb{R}^d \times \mathbb{R}^d$ and $\ell(h, x, y) = \left\|x - h_y\right\|^2$

_QDA in 1D : $H = \left\{\pi_y N(x | \mu_y, \sigma_y) | \mu_y \in \mathbb{R}, \sigma_y > 0\right\}$ and $\ell(h, x, y) = -\log h(x, y)$

Lots of Linear Stuff

$\phi(x)$

$\phi : \mathbb{R}^d \longrightarrow \mathbb{R}^D$

$w^T \phi(x) + w_0$

$C_d(x) = C_D(\phi(x))$

_How to construct nonlinear classifiers from linear ones?