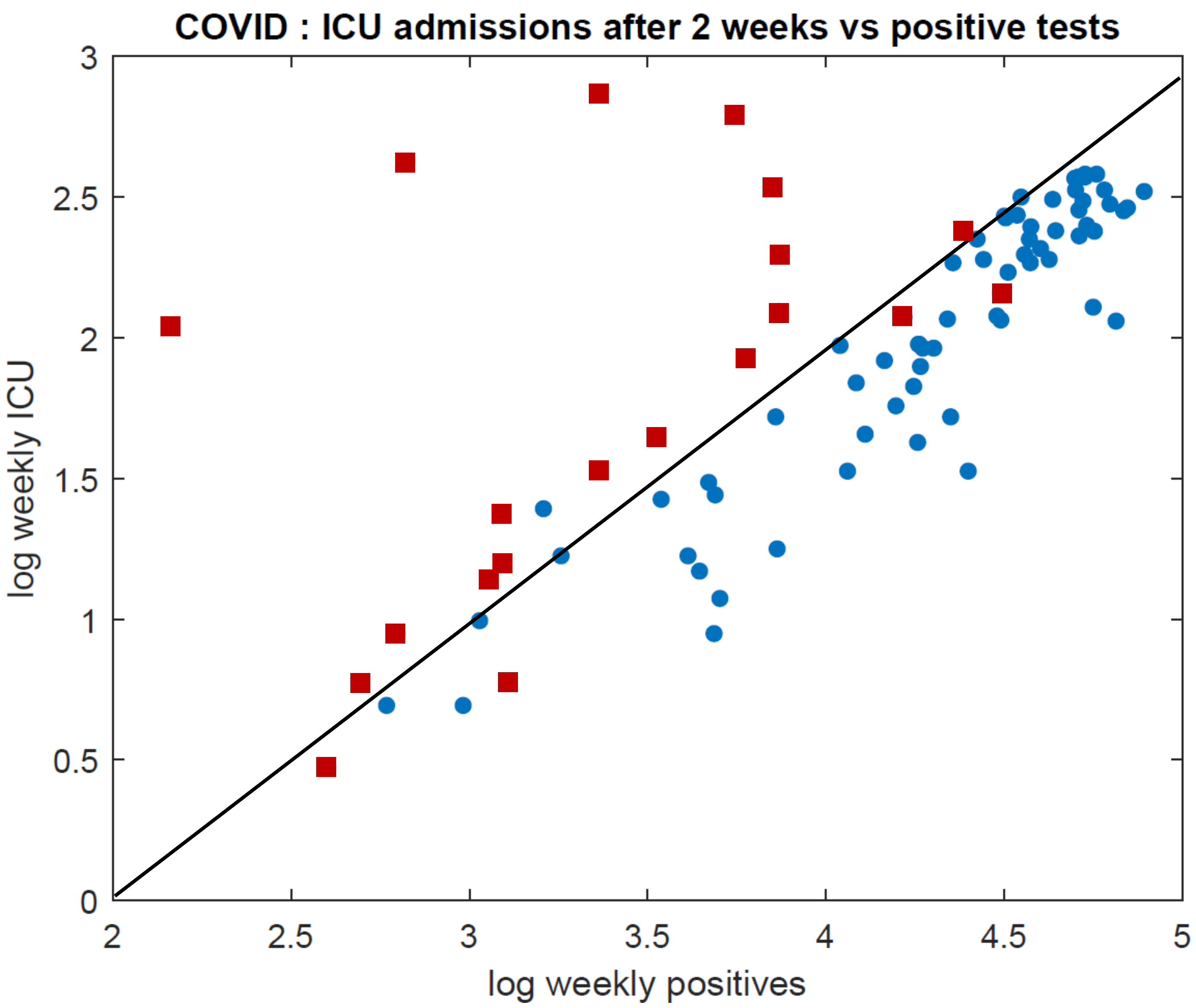
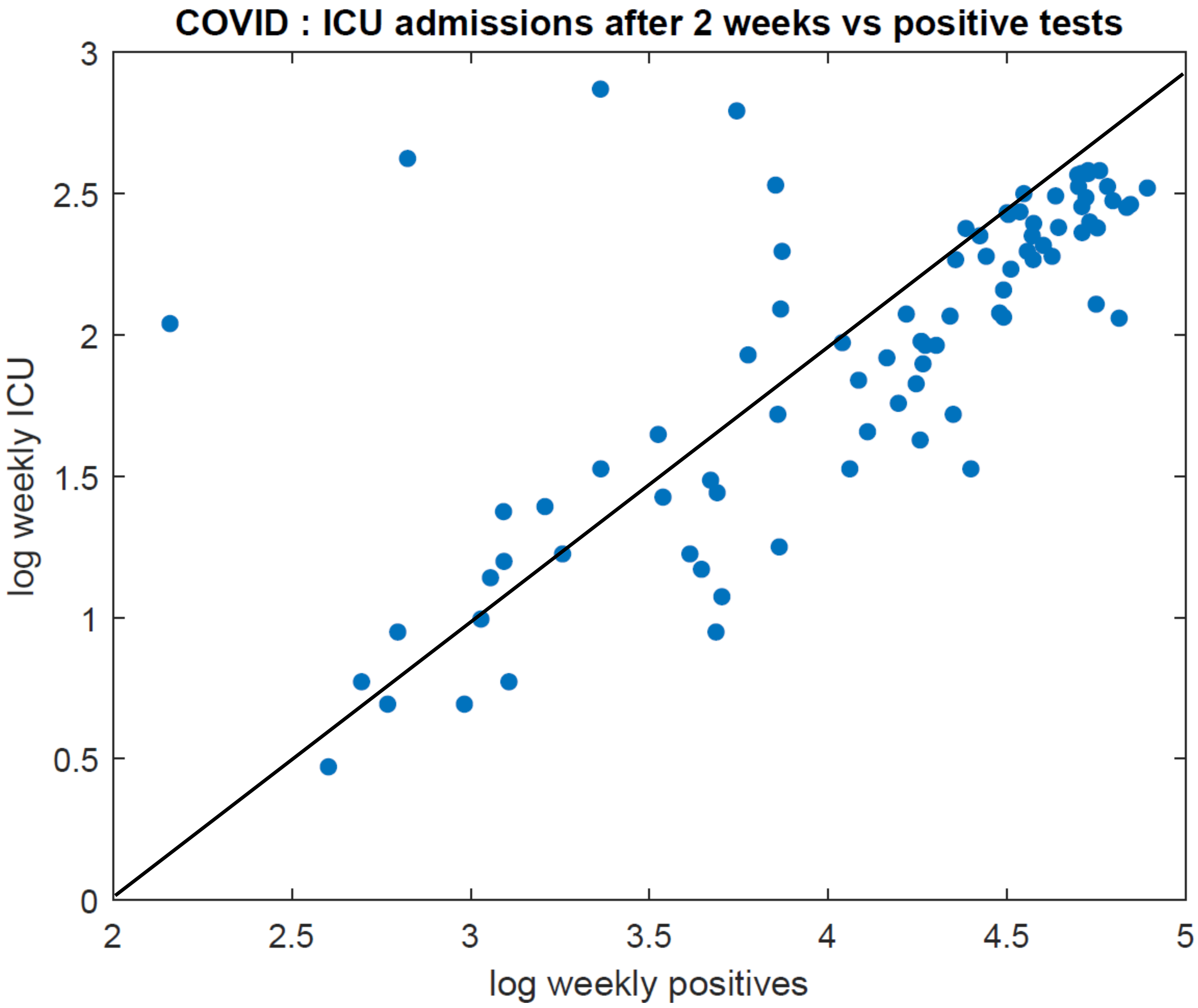
› Linear regression or linear classification?



- Linear regression
  or
  linear classification
  ?



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 general 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?]

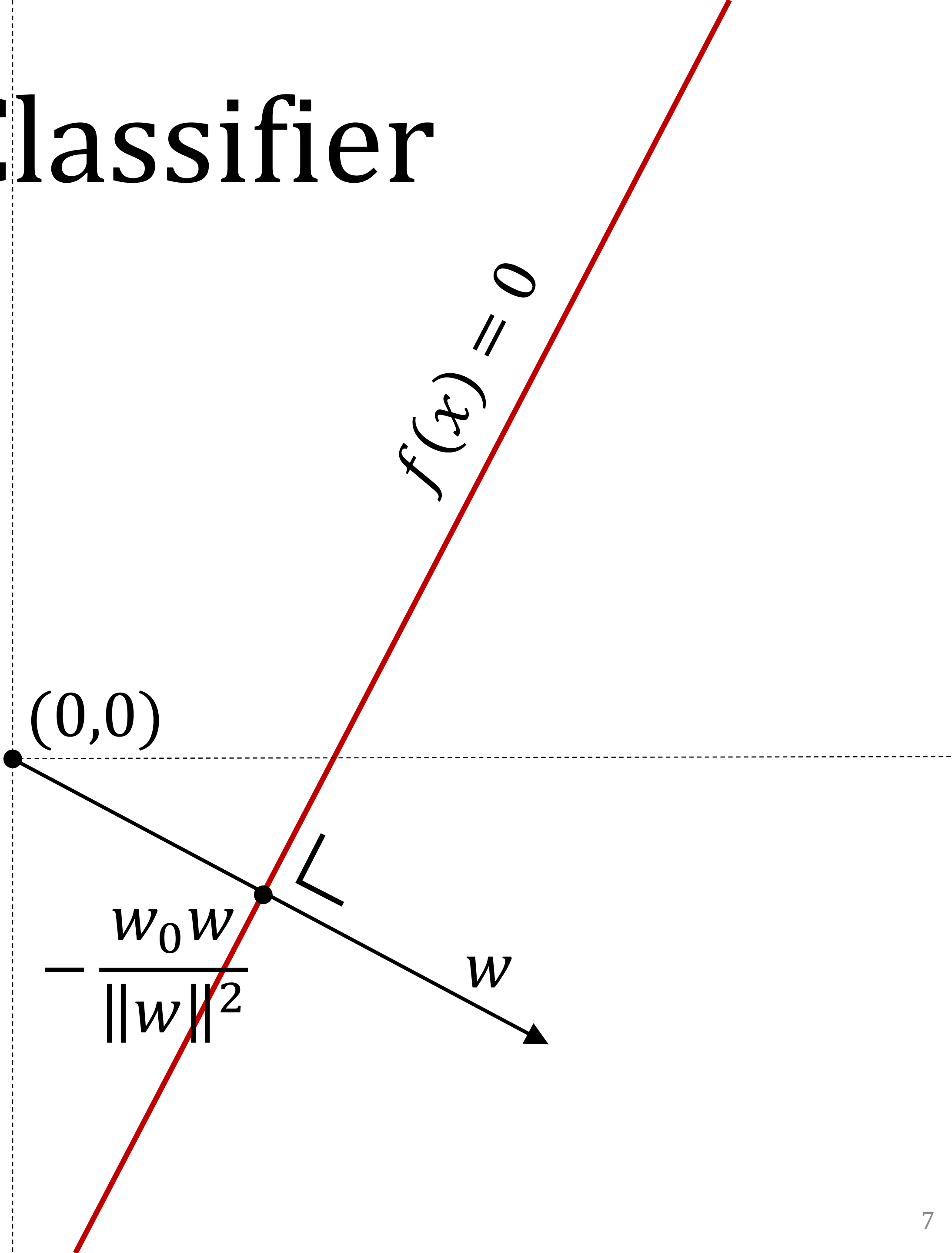
$$\int (f(x) - y)^2 p(x, y) dx dy$$

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

Variation on Theme : Logistic Regression

Let's Assume Linear "Logit"

- Take $f(x) = \log p(y_1|x) - \log p(y_2|x)$ and assume class-conditionals to be 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 $p(y_1|x)$...

Logistic Regression

› Classifier that takes $p(y_1|x) = \dfrac{1}{\exp(-w^T x - w_0) + 1}$

What shape does this have as a function of x?
How do we now find the actual parameters?

[Conditional] Likelihood!

› Maximize [its logarithm]

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

Rewrite into Minimization…

› Identify $y_1 = +1$ and $y_2 = -1$

› Then minimize

[What exactly do we minimize over?]

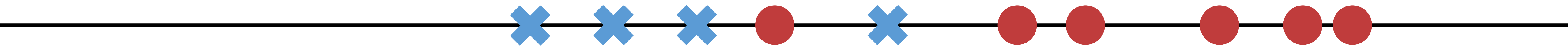
$$\sum_{i=1}^{N} \log_2(\exp(-y_i f(x)) + 1)$$

Fisher & Linear Regression

Linear Classifier by Least Squares?

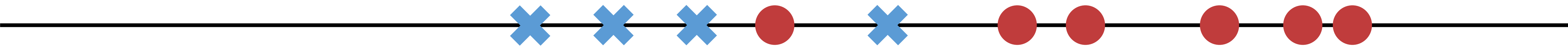
› Also referred to as Fisher classifier, FLD,…

› How to?



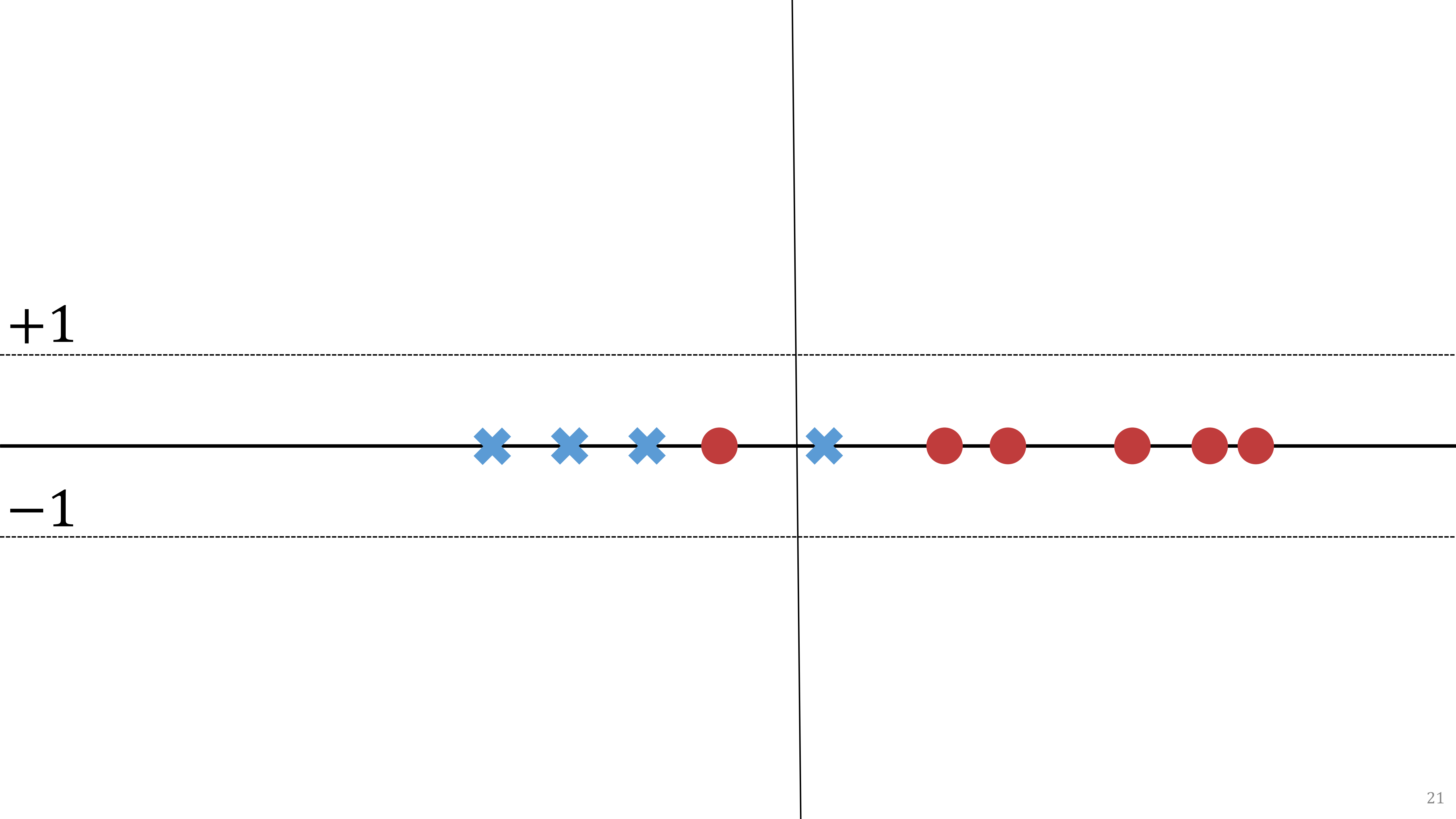
Linear Classifier by Least Squares?

- How to?
- Again identify $y_1 = $ ✖ $= +1$ and $y_2 = $ ● $= -1$?

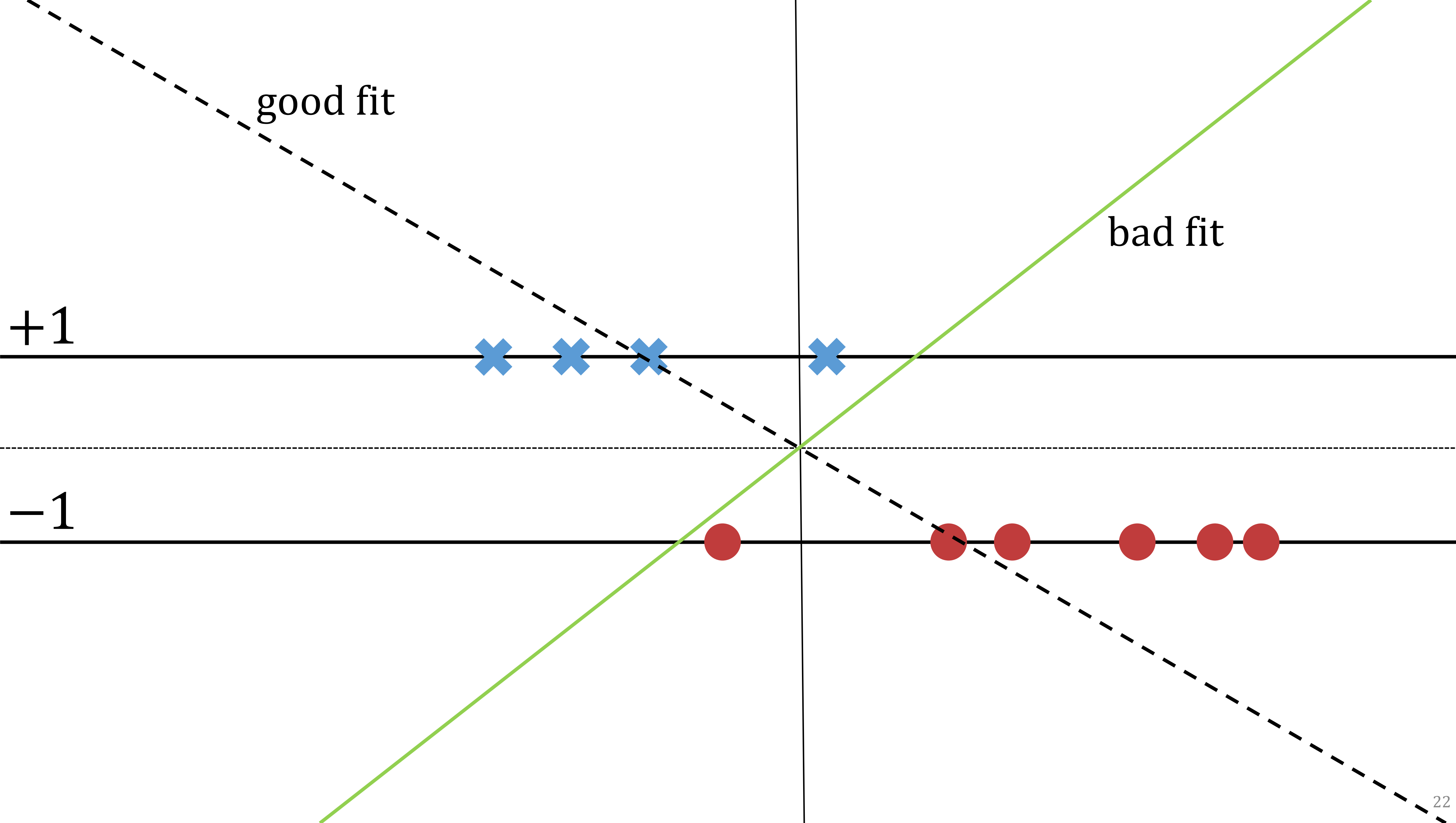


We Get…

$$\sum_{i=1}^{N} ?$$



+1
−1


good fit
bad fit
+1
−1

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

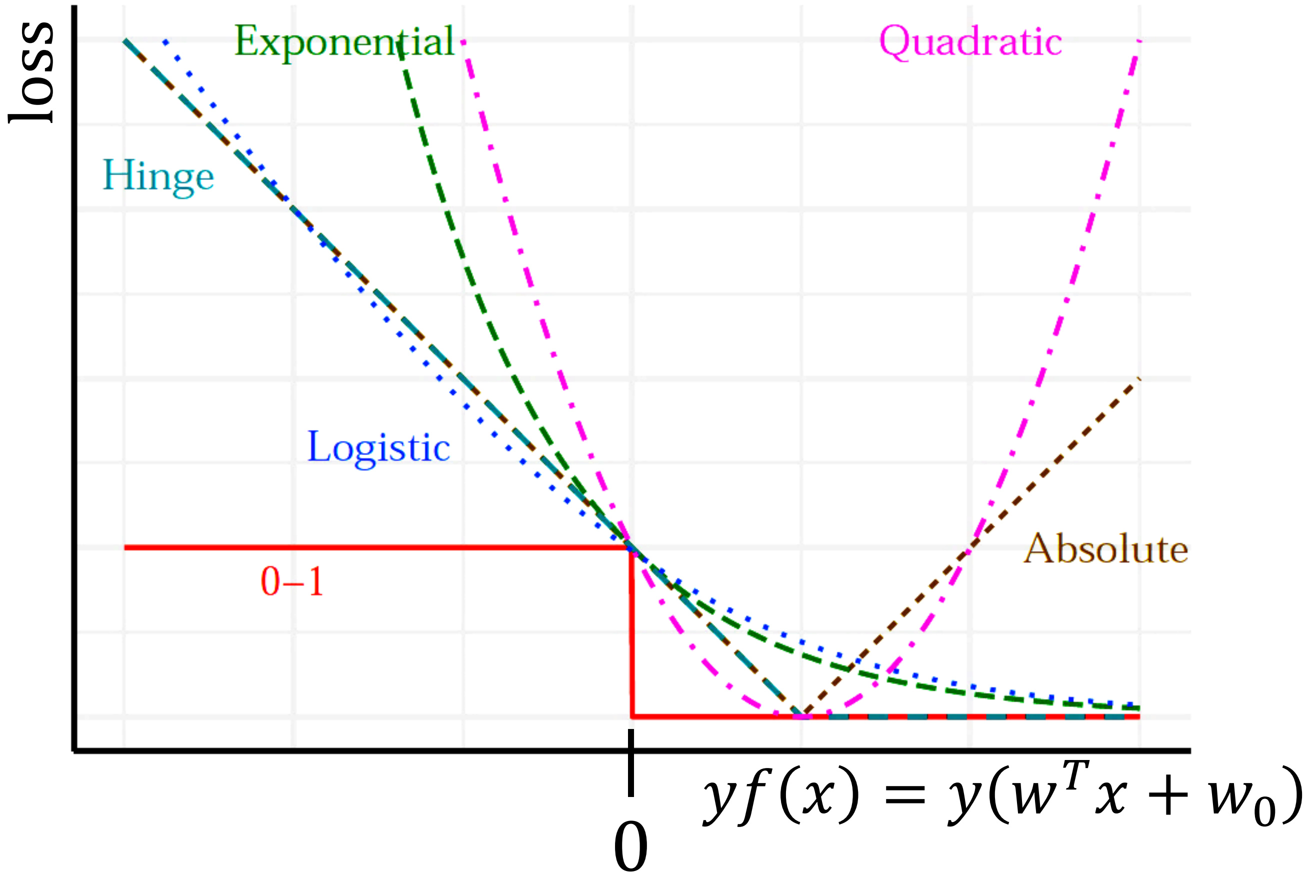
Formulations are Not Unique!

› NMC : spherical Gaussian model + LL
means as model + squared deviation

› Logistic regression : sigmoidal posterior + LL
linear model + logistic loss

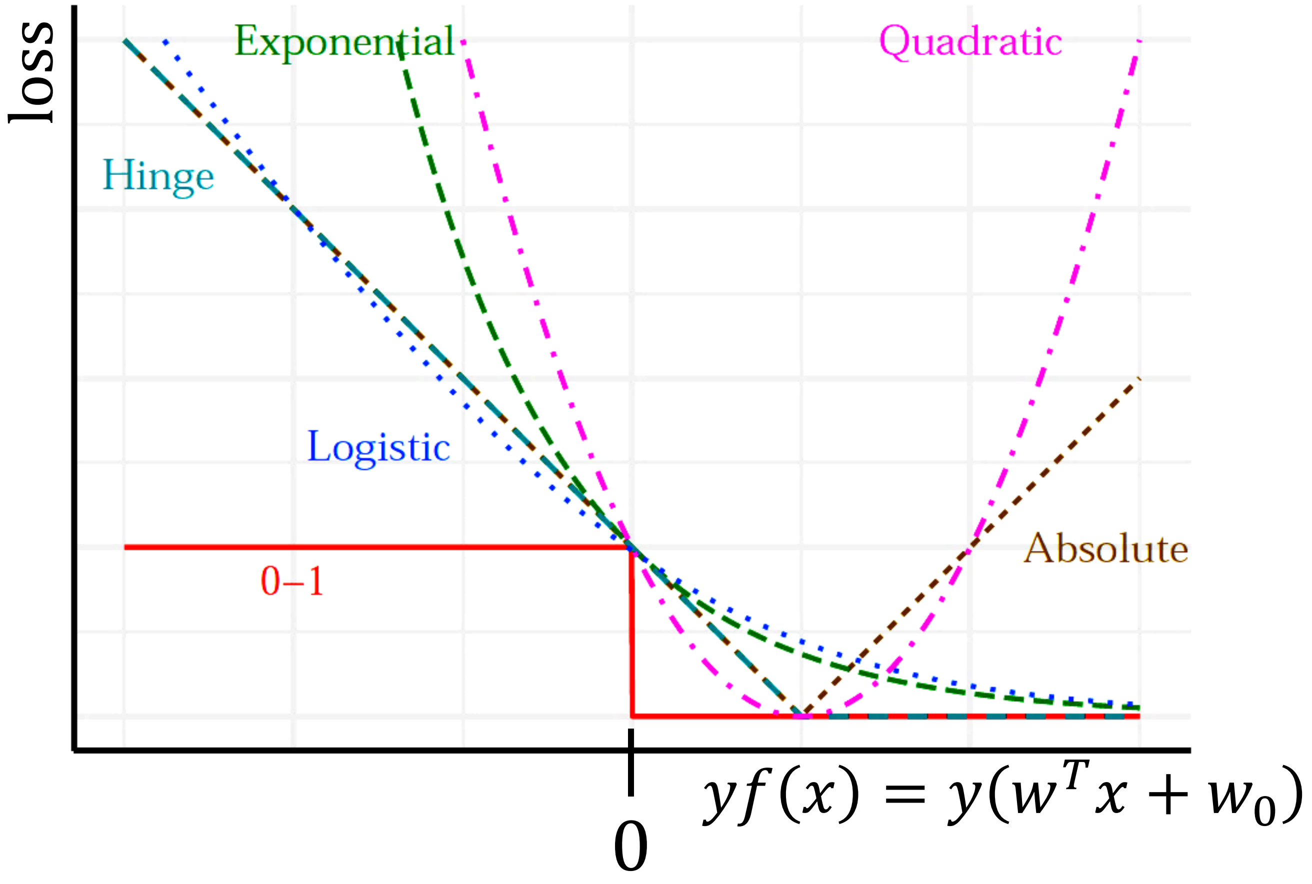
$$\sum_{i=1}^{N} \log_2\left(\boxed{\exp(-y(\boxed{w^T x + w_0})) + 1}\right)$$

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…

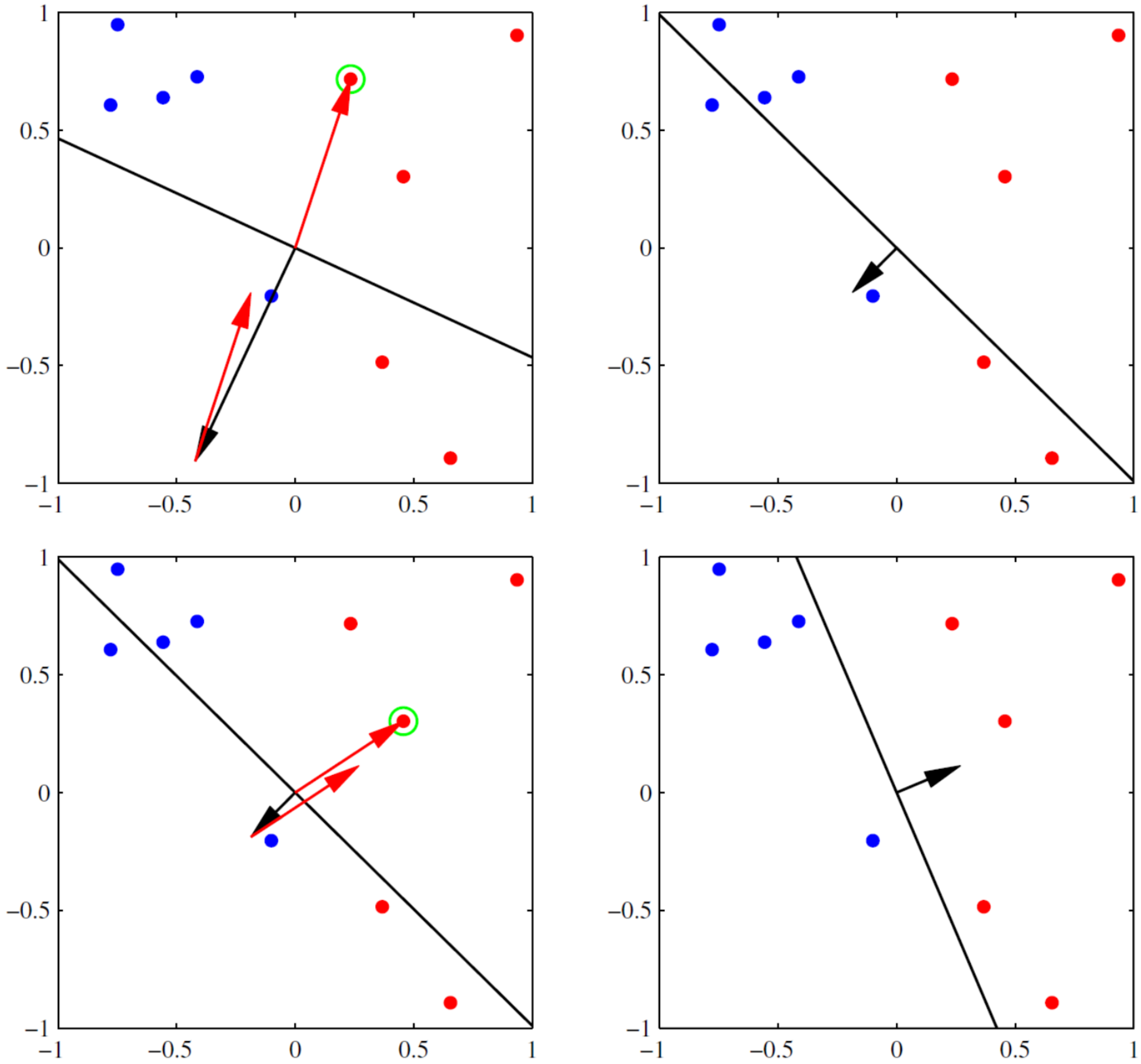
- Way of optimizing = integral part of this learner

  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

[with η=1]



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

› 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 :

  $H = \{w^T x + w_0 \mid w \in \mathbb{R}^d, w_0 \in \mathbb{R}\}; \ell(h, x, y) = (h(x) - y)^2$

  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 :

  $H = \left\{\pi_y N\left(x \middle| \mu_y, \sigma_y\right) \mid \mu_y \in \mathbb{R}, \sigma_y > 0\right\}; \ell(h, x, y) = -\log h(x, y)$

Lots of Linear Stuff

› How to construct nonlinear classifiers from linear ones?