DATA221 Intro Machine Learning
10-Soft margin discrimination

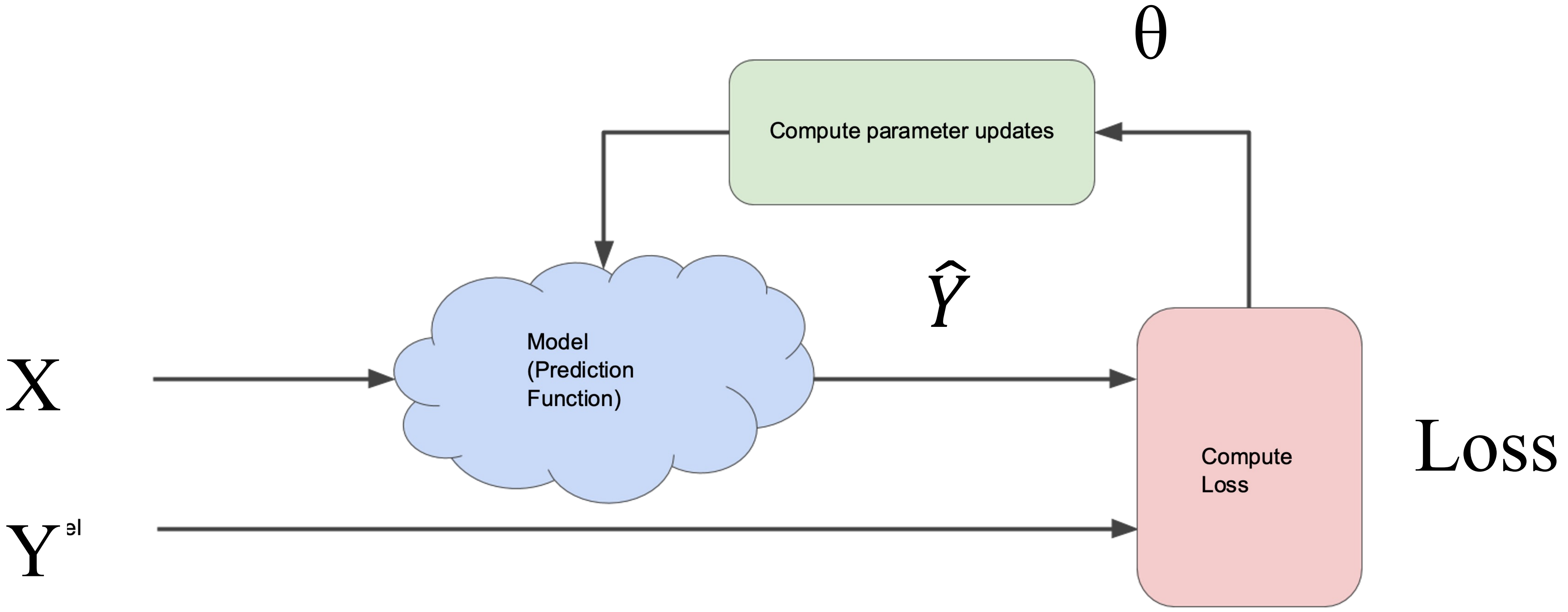
William Trimble

Spring 2023

THE UNIVERSITY OF CHICAGO

The optimization paradigm

$$\hat{\theta} = \mathrm{argmin}_{\theta}\ L(\theta,\ X,\ Y)$$



Linear discrimination

- So there were a handful of approaches for classification that divided the space of features up with straight lines:
  - linear regression
  - logistic regression
  - Gaussian models + shared covariance matrix

- And there were a handful of methods that would allow nonlinear boundaries:
  - linear discrimination with kernel vectors that are nonlinear in the features
  - Gaussian models + different covariance matrices

Soft Margin Classification

- Logistic regression came from statistics, and drops out of an argmax formulation.
- There is another approach that conceptually comes out of engineering but uses the same machinery (argmax of a loss function that evaluates the mismatch between the model and the data) but starts from a different place (like discrete math CS vs calculus stats)
- The textbooks say that it was developed in the 90s and that after some years it was found to be equivalent to classification/regression with a certain sort of loss functions.
- Most of the terms in the soft margin loss function vanish; only a few remain nonzero; convenient for some problems.

Linear on the frontend…

$$z = \begin{bmatrix} x_0 \ x_1 \ x_2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

Nonlinear in the rear

$$\phi(z) = \mathrm{logistic}(z) = \frac{1}{1 + exp(-z)}$$

Linear on the frontend…

$$z = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

Nonlinear in the rear

$$\phi(z) = \mathrm{sign}(z)$$

Linear on the frontend…

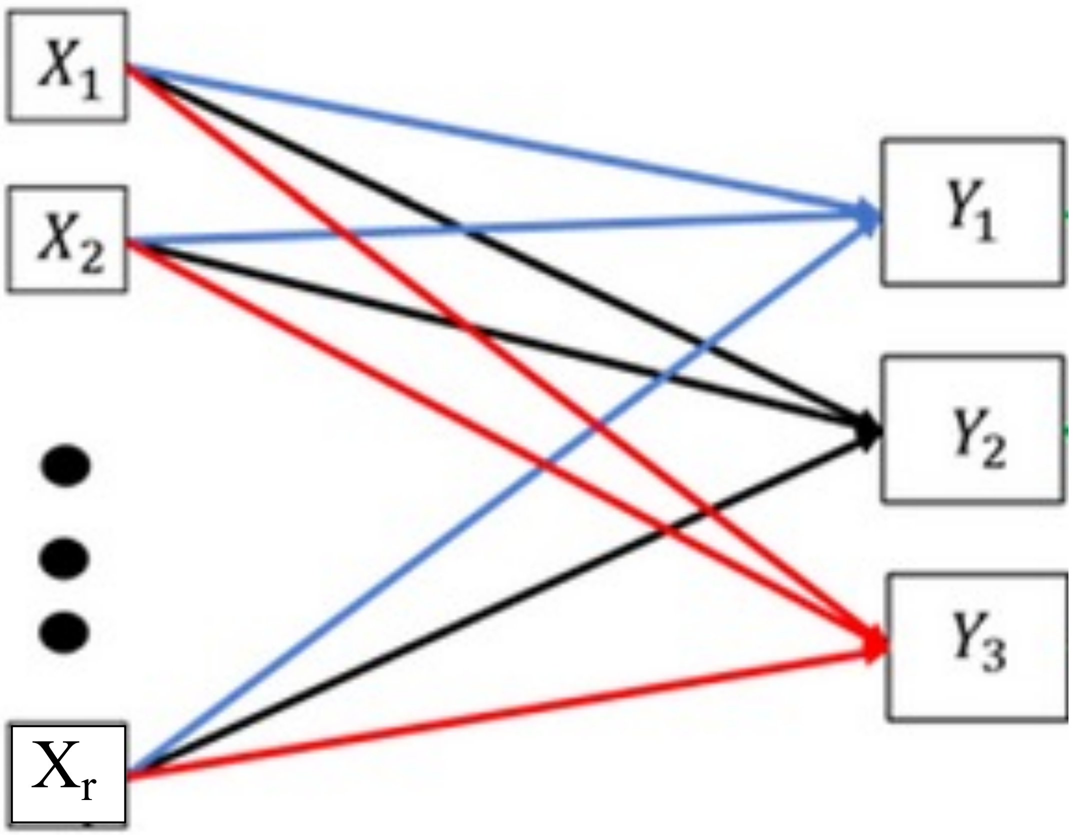
$$z = \begin{bmatrix} x_0 \; x_1 \; x_2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

Nonlinear in the rear

$$\phi(z) = \Theta(z > 0)$$

Beak length, mass

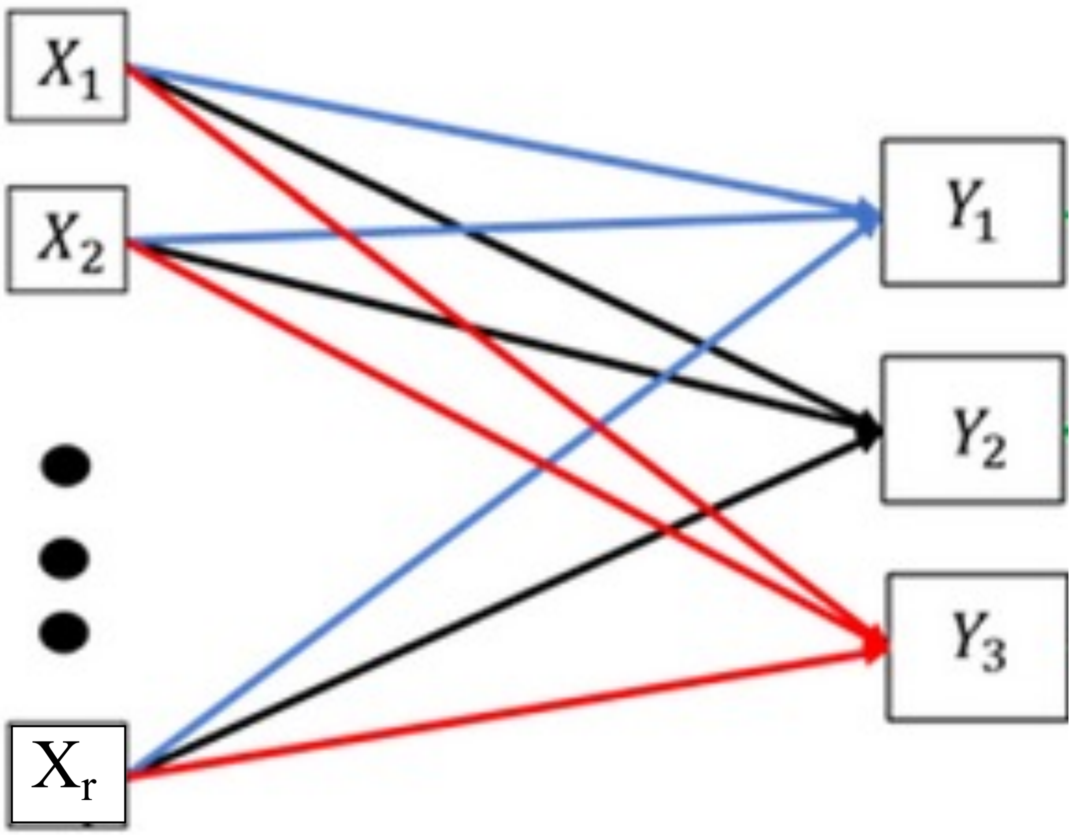
Species inference



Beak length, mass

Species inference

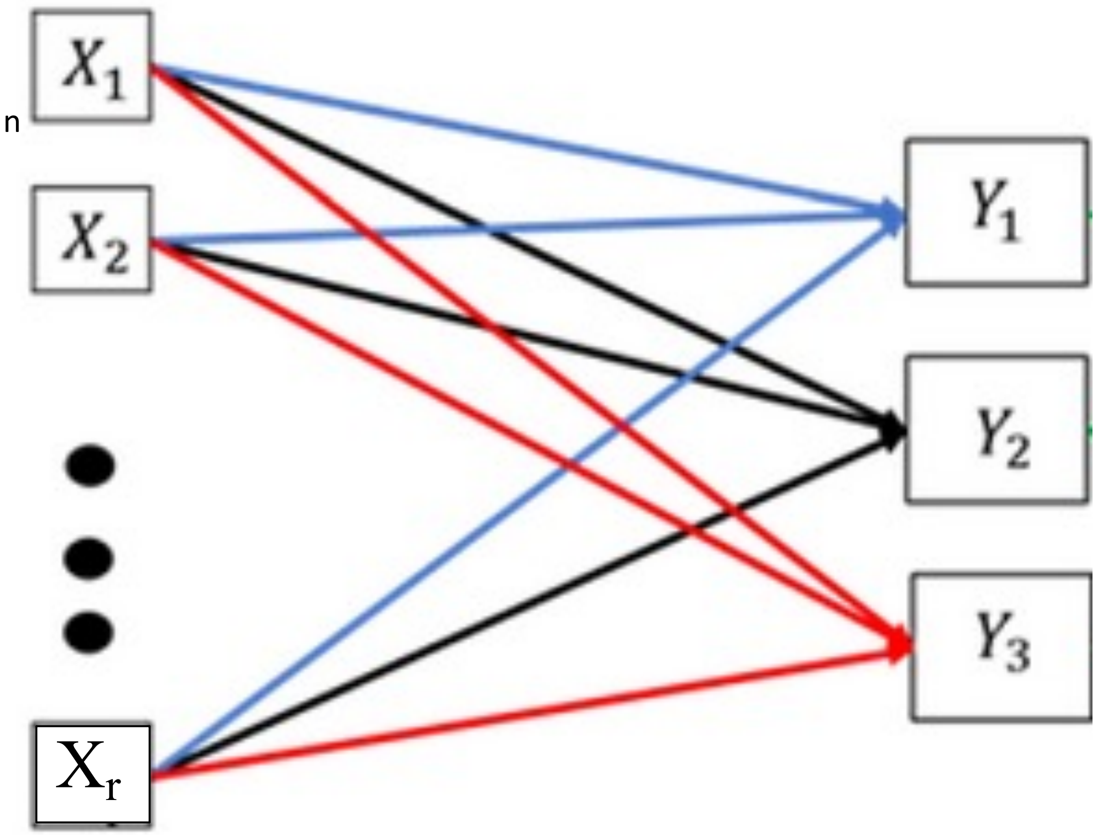
$$y = X w + b$$

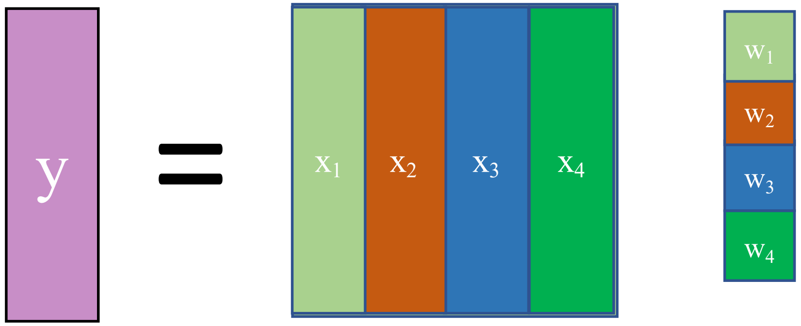


Beak length, mass

Species inference

n



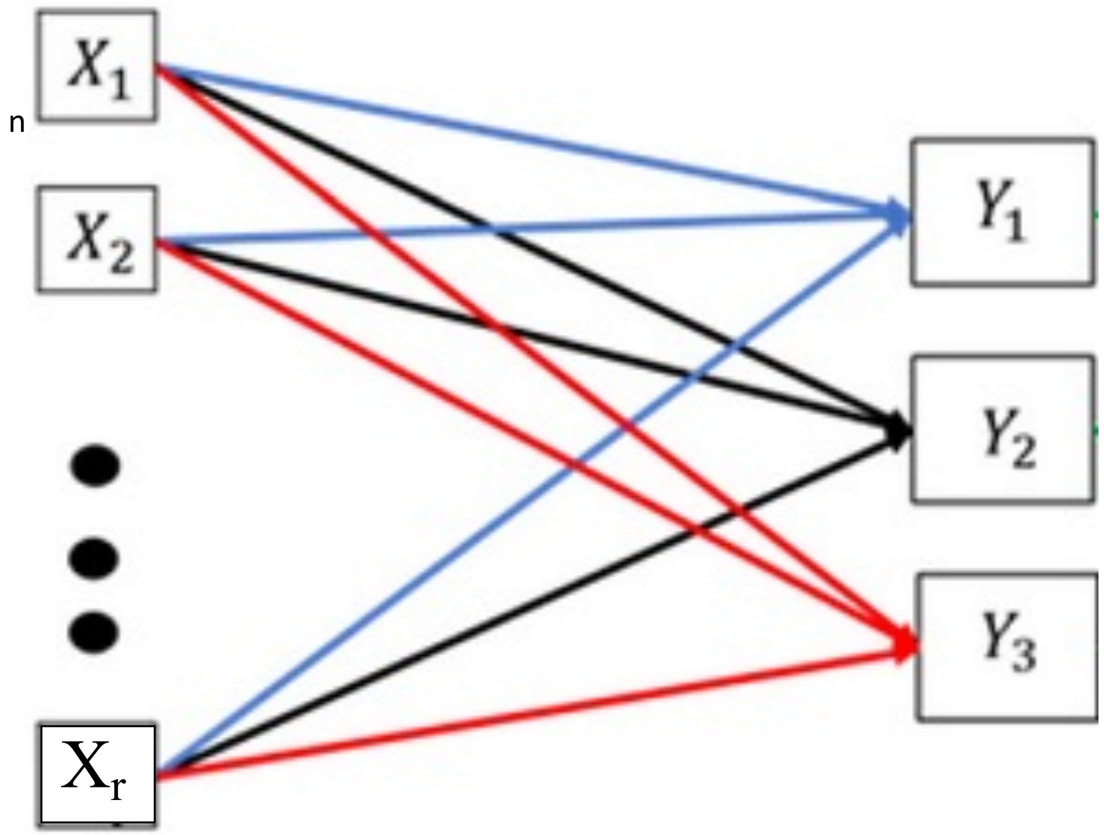


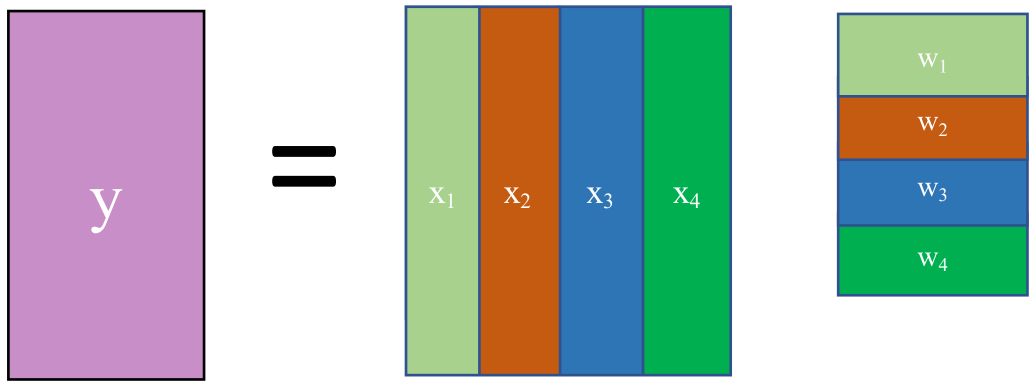
(n x 1) (n x r) (r x 1)

Beak length, mass

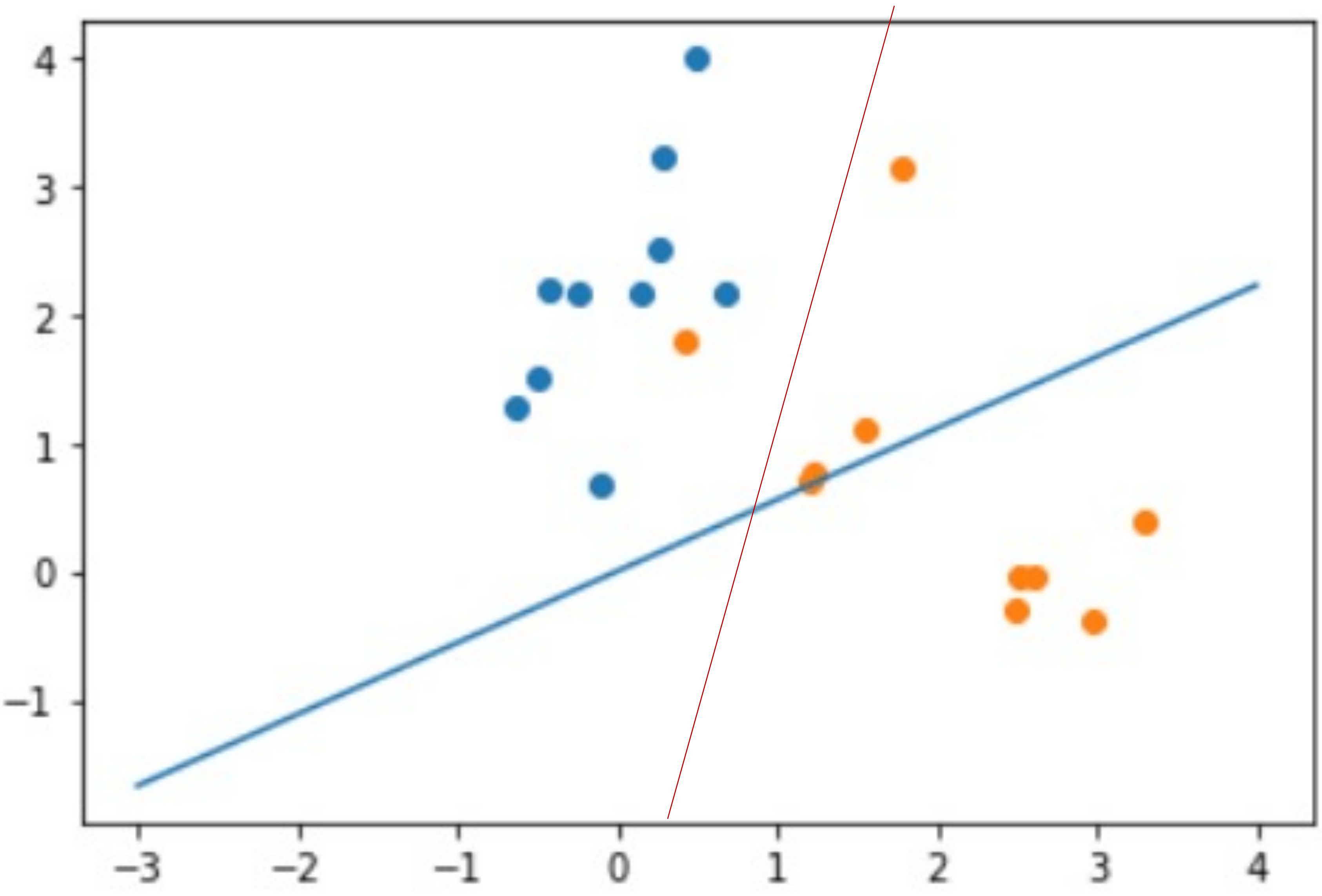
Species inference

n





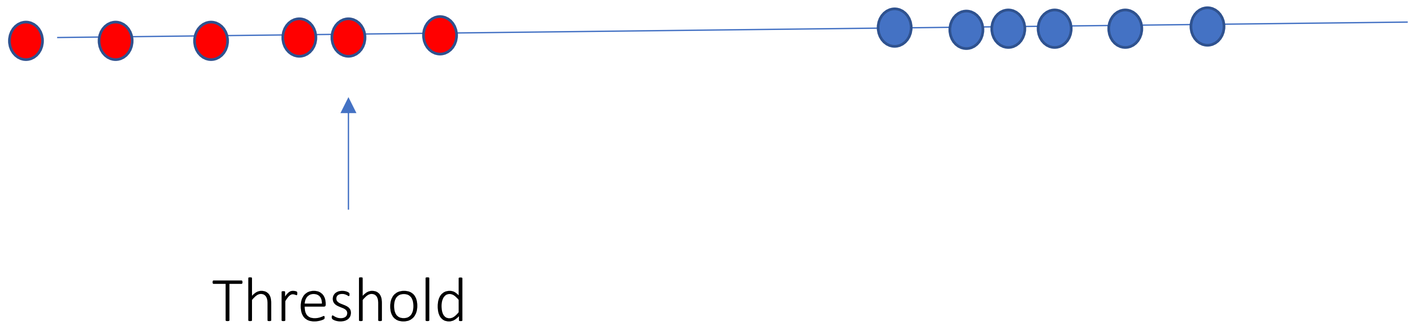
(n x p) (n x r) (r x p)



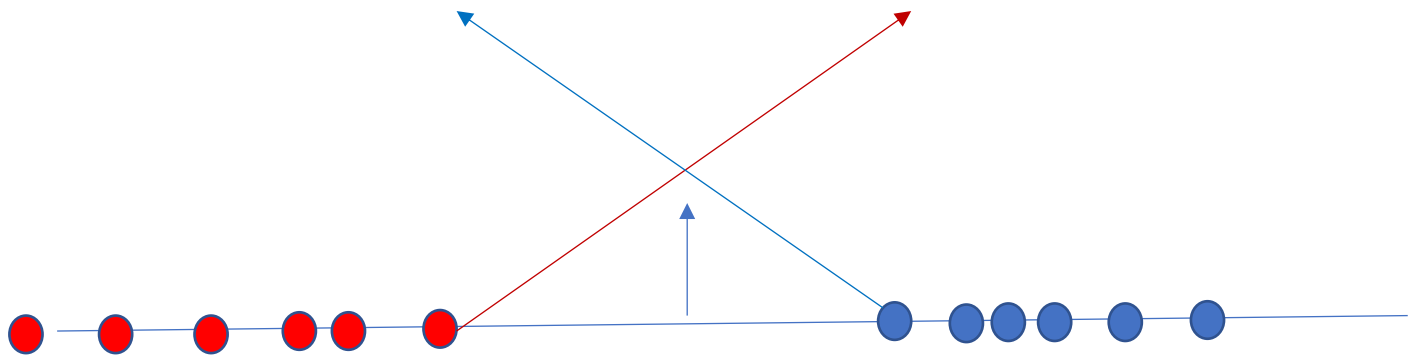
The thick line is the linear regression line.

The red line makes only 1 misclassification

Special case: perfectly separable features

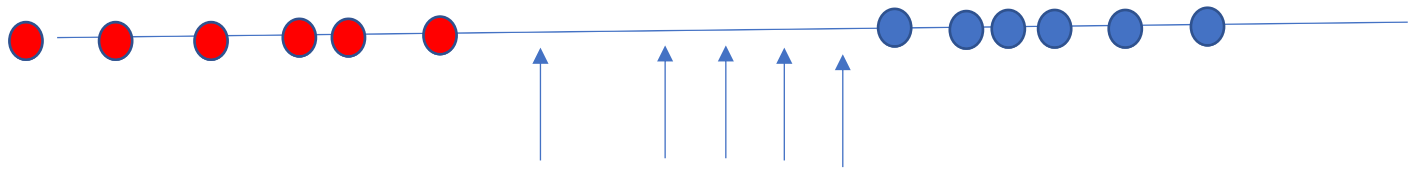


Can you find a threshold that separates the classes perfectly?



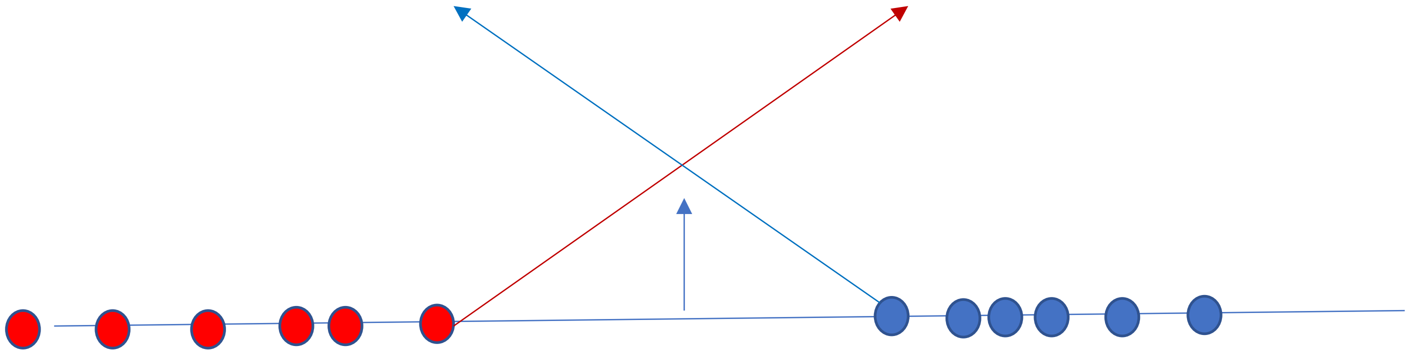
"Maximum margin" threshold maximizes the sum of the minimum distances to the decision boundary

Can you find a threshold that separates the classes perfectly?



Threshold?

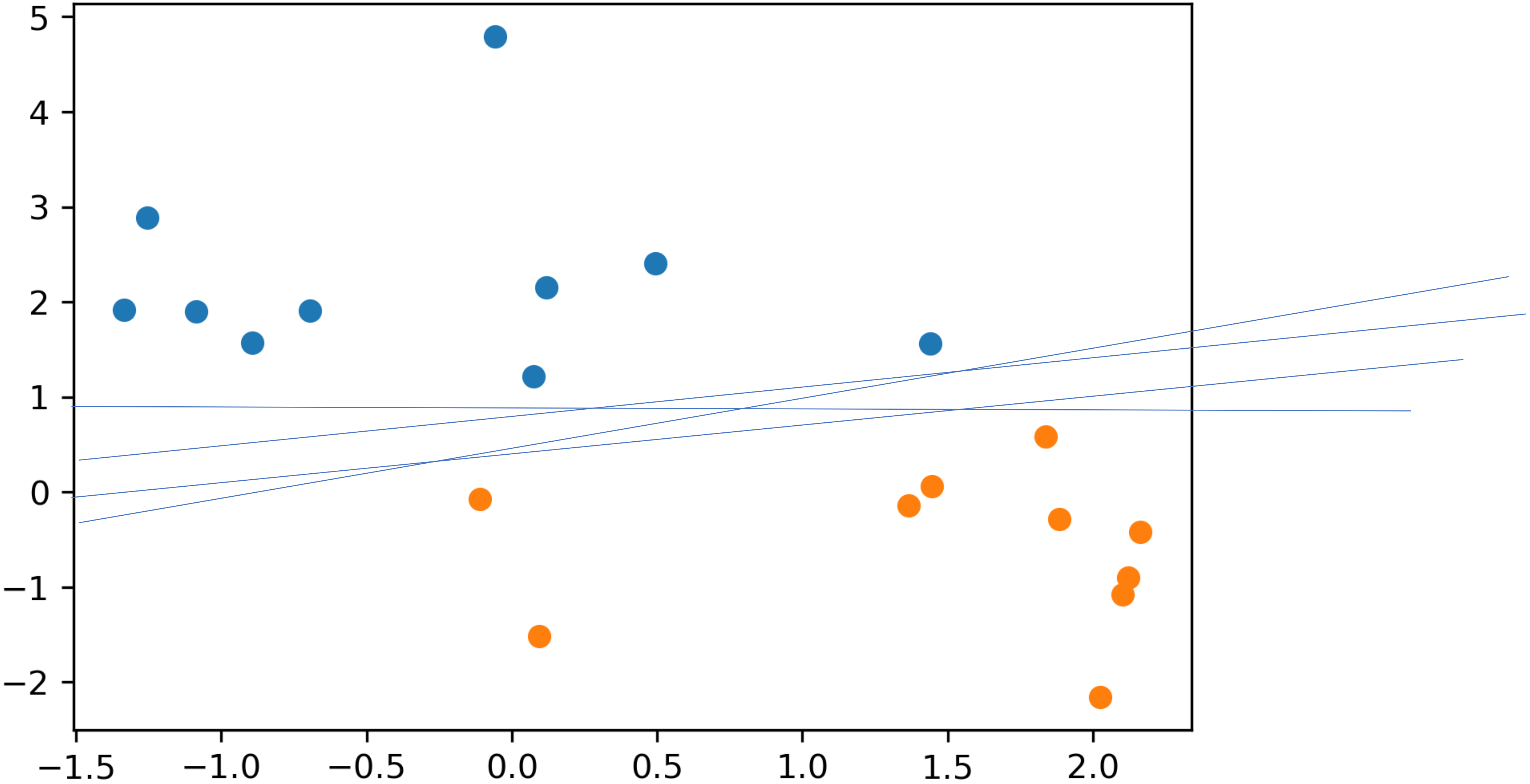
Can you find a threshold that separates the classes perfectly?

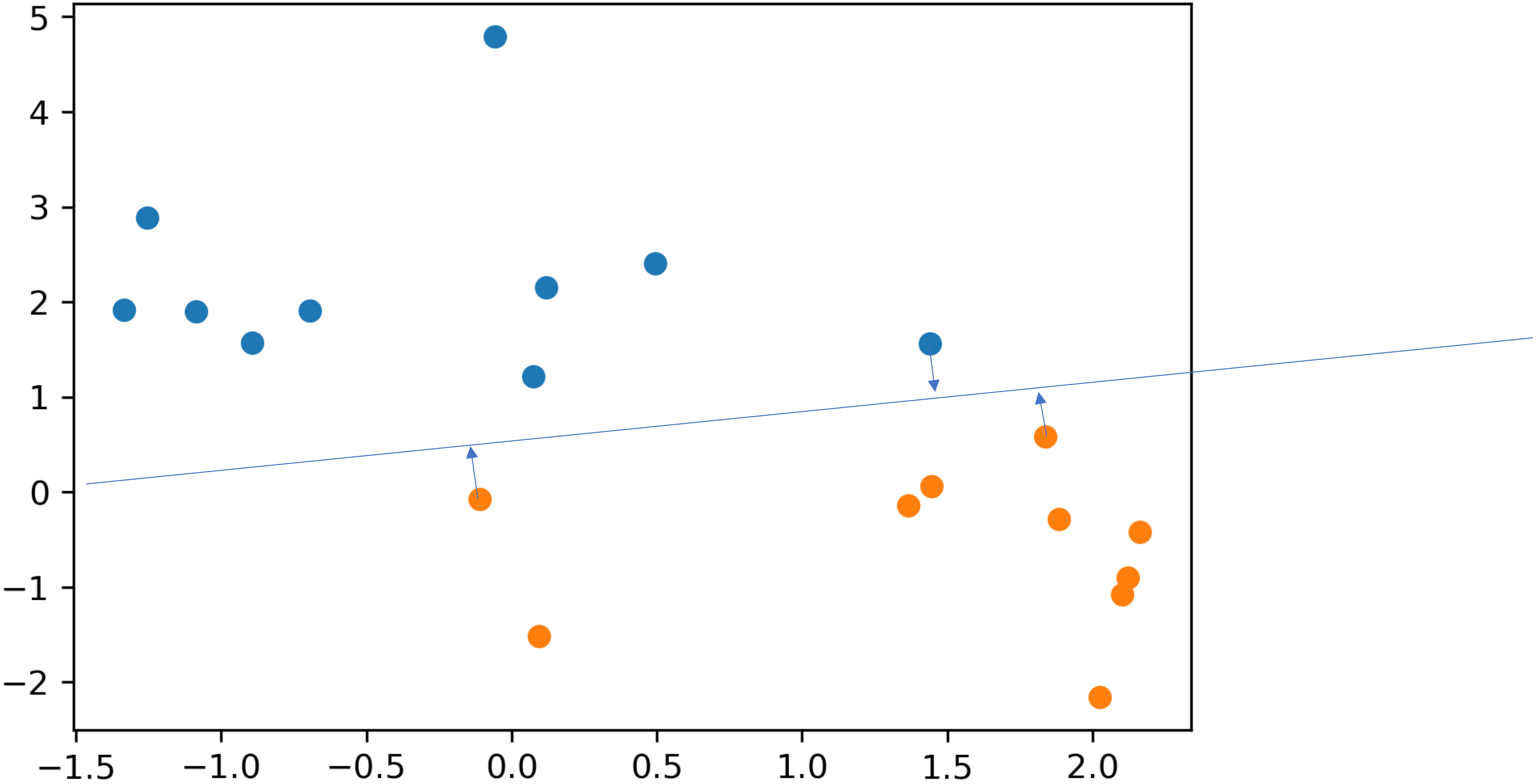


“Maximum margin” threshold maximizes the sum of the minimum distances to the decision boundary

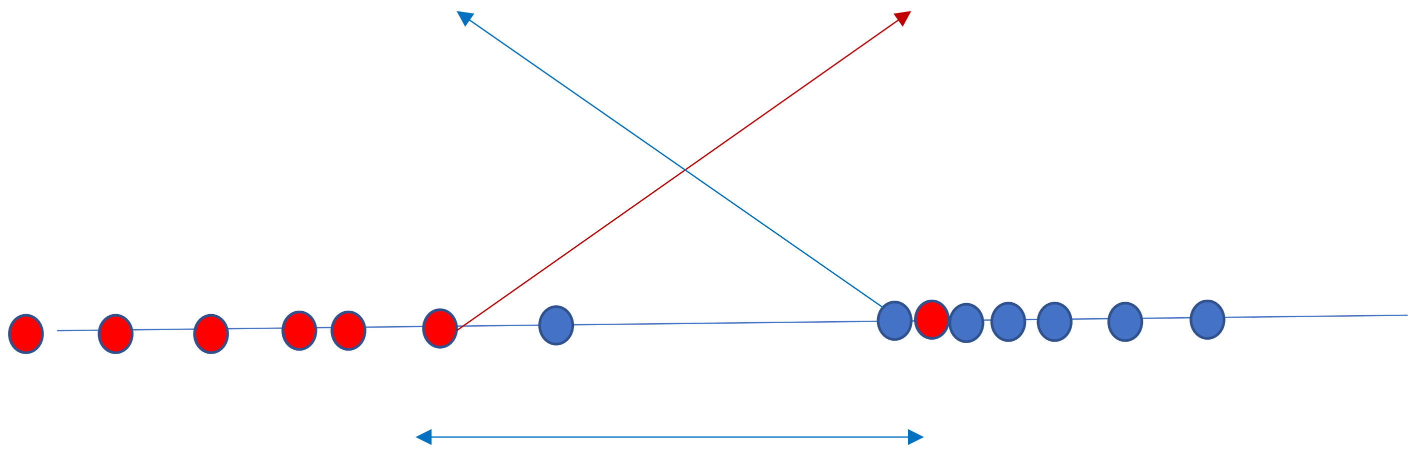
NOTE: The placement of the decision threshold depends here on just two points, not a sum over all the data.   The correctly classified points have zero weight in computing the threshold.

One vs. many...



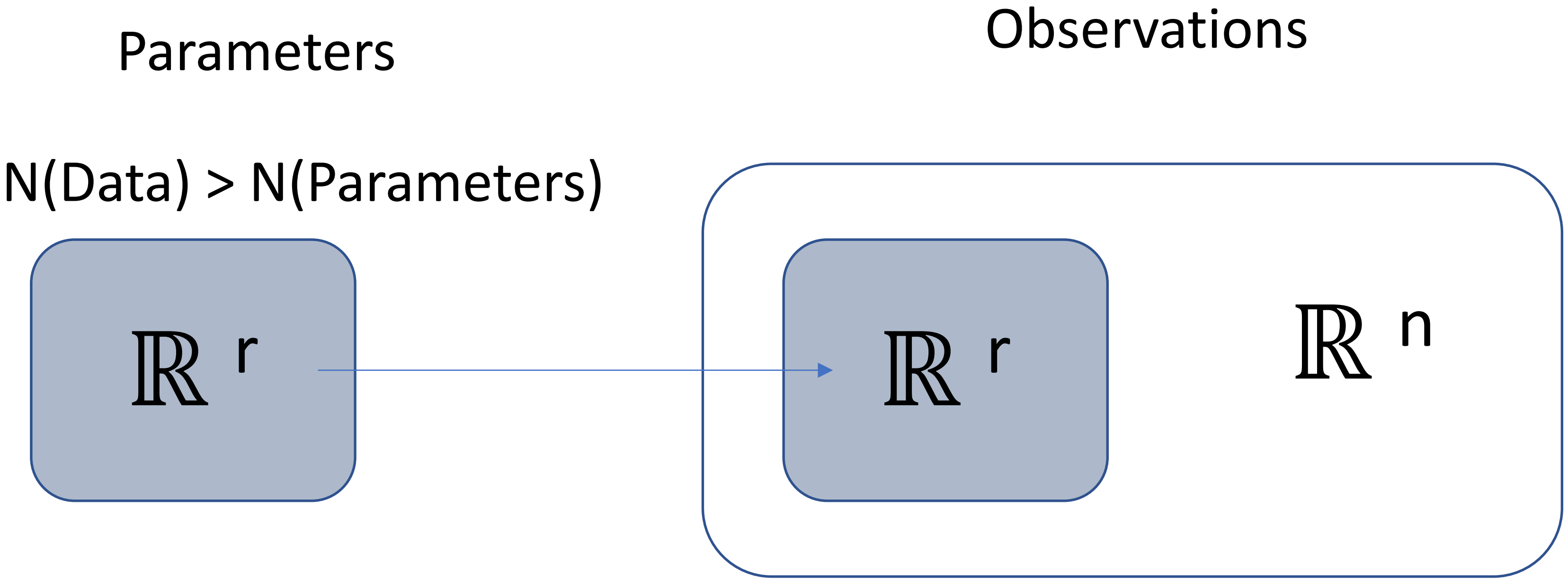

5
4
3
2
1
0
−1
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

“Maximum margin” doesn’t apply when some points are misclassified… so a compromise was found



What did the engineers come up with? A fudge factor for the loss function.

Overdetermined
Underparameterized
Normal ”fitting”,
least-squares
or not, is like this.

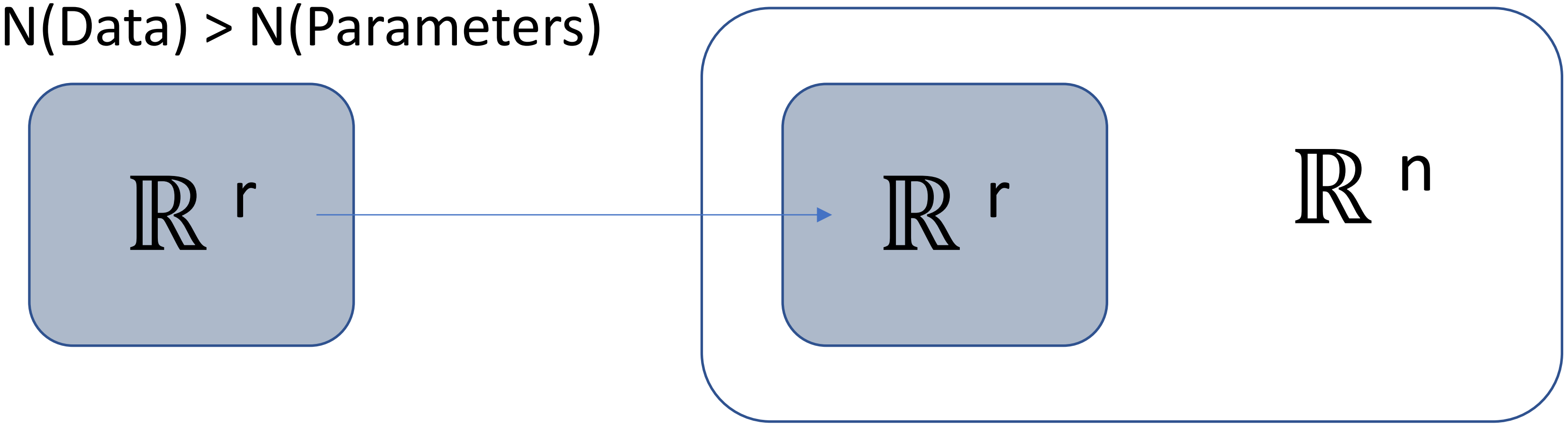


Parameters

Observations

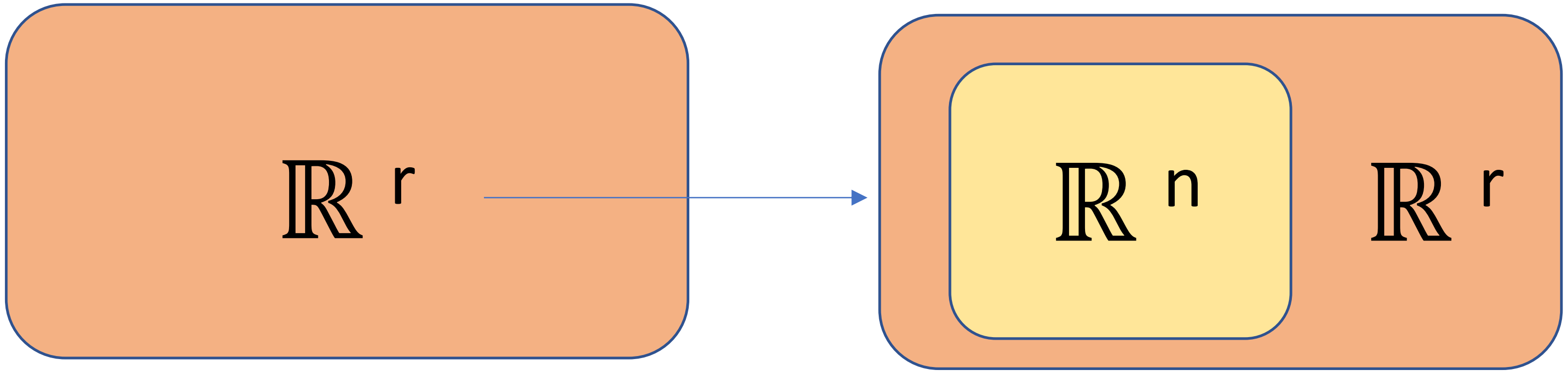
N(Data) > N(Parameters)

Overdetermined
Underparameterized
Normal "fitting",
least-squares
or not, is like this.

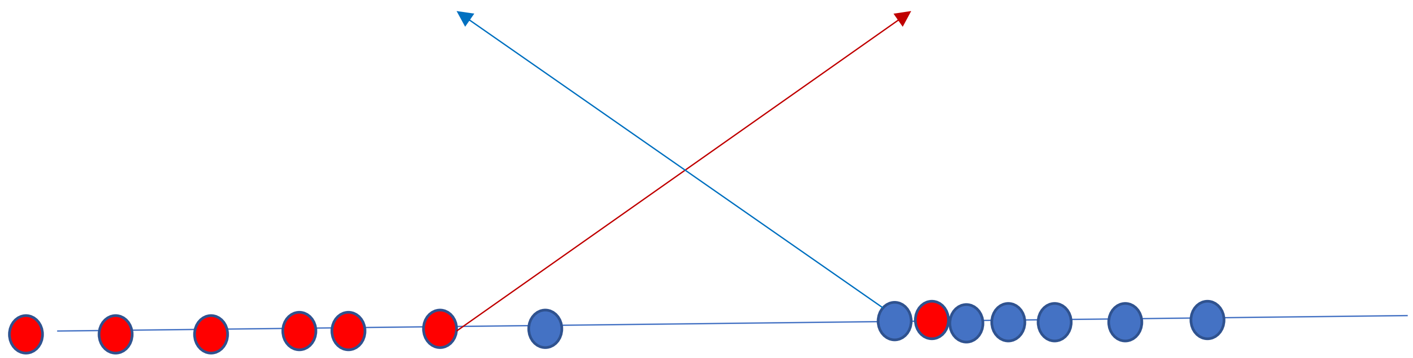


N(Parameters0) > N(Data)

Underdetermined
Overparameterized

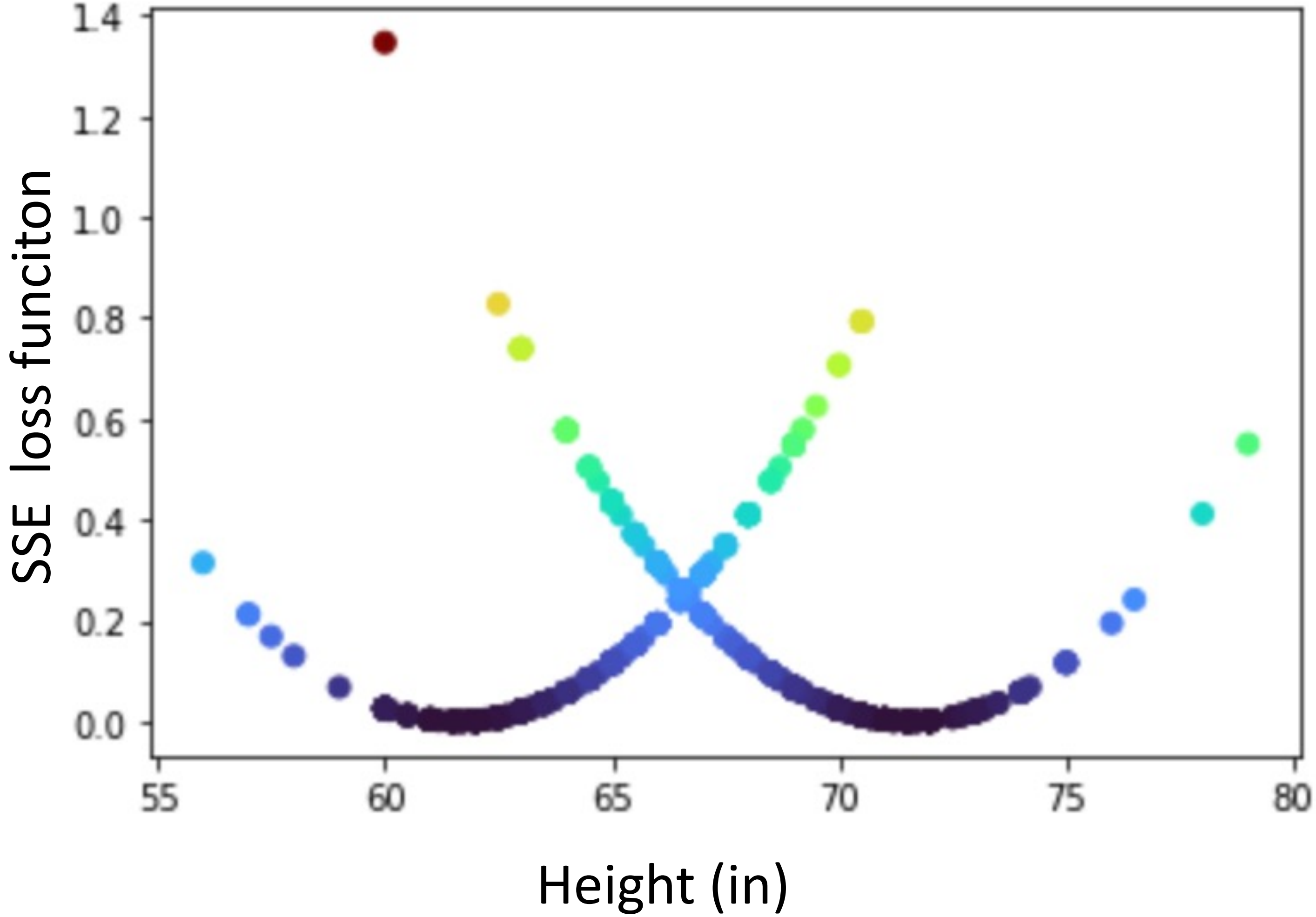


“Maximum margin” doesn’t apply when some points are misclassified… so a compromise was found

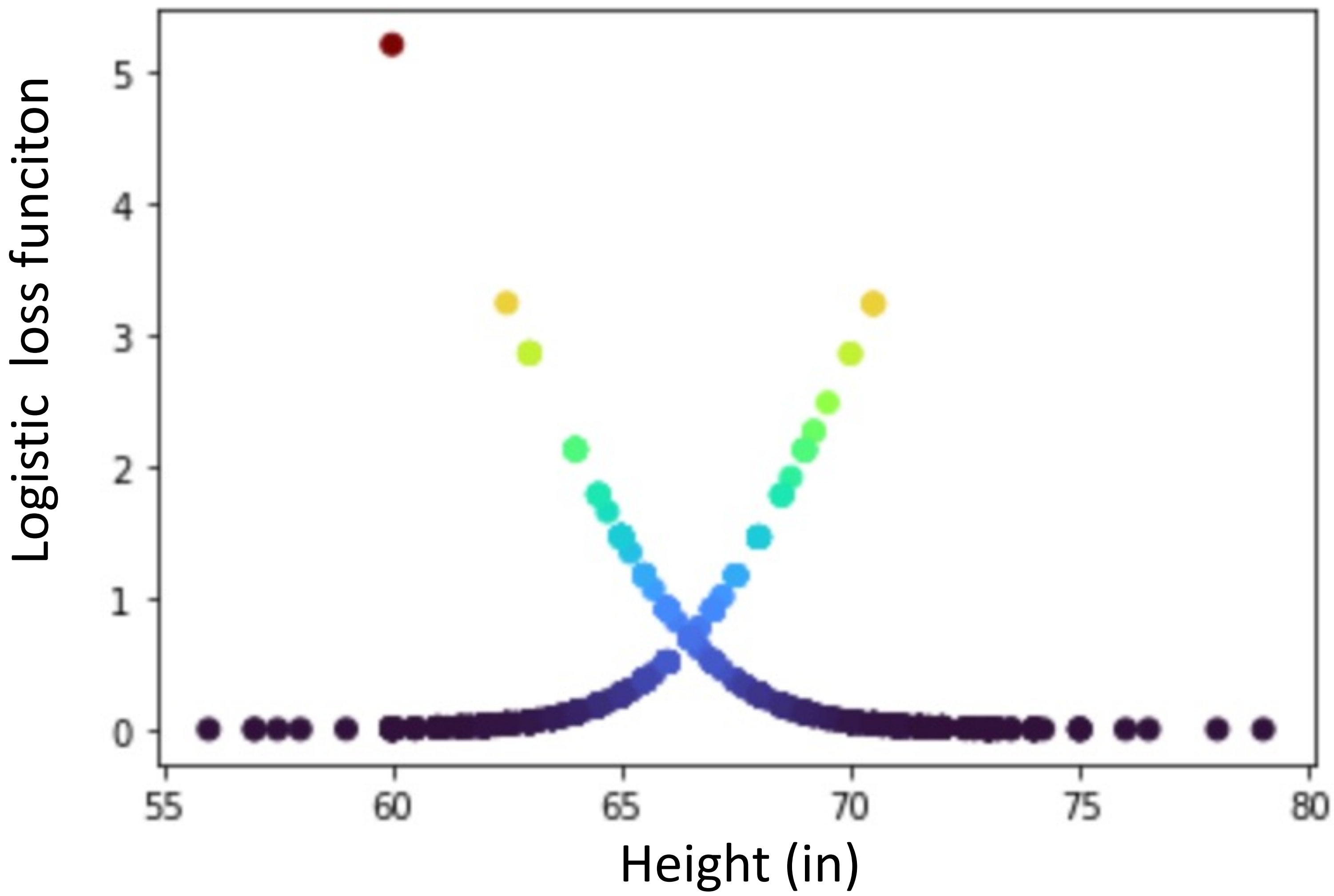


What did the engineers come up with? A fudge factor for the loss function.
“Margin” is a hyperparameter that defines a band around the decision threshold.
Ignore all correctly classified points;
Permit but penalize misclassifications within the margin

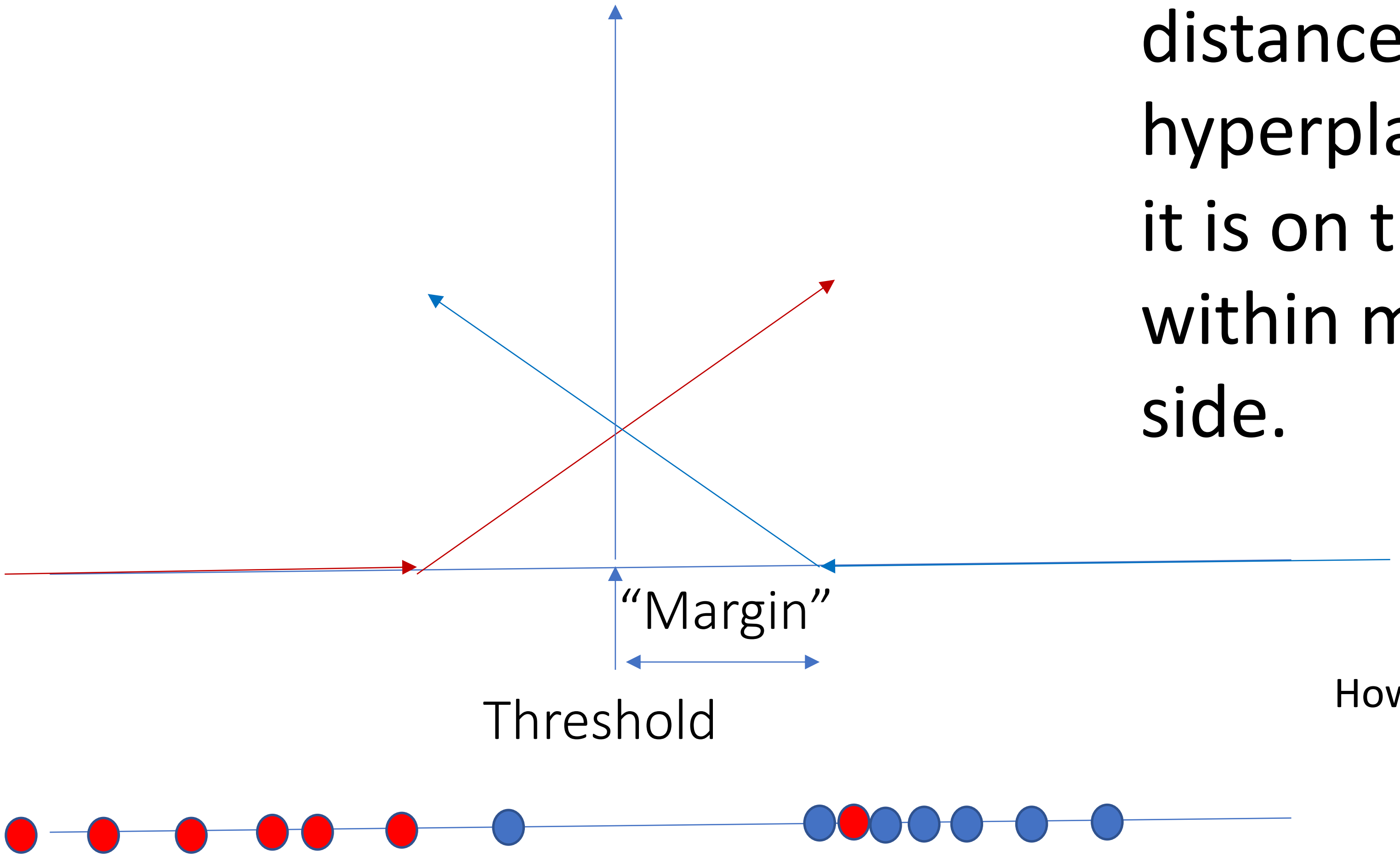
SSE loss function for Galton height data



Logistic loss function on Galton height data



Hinge loss function

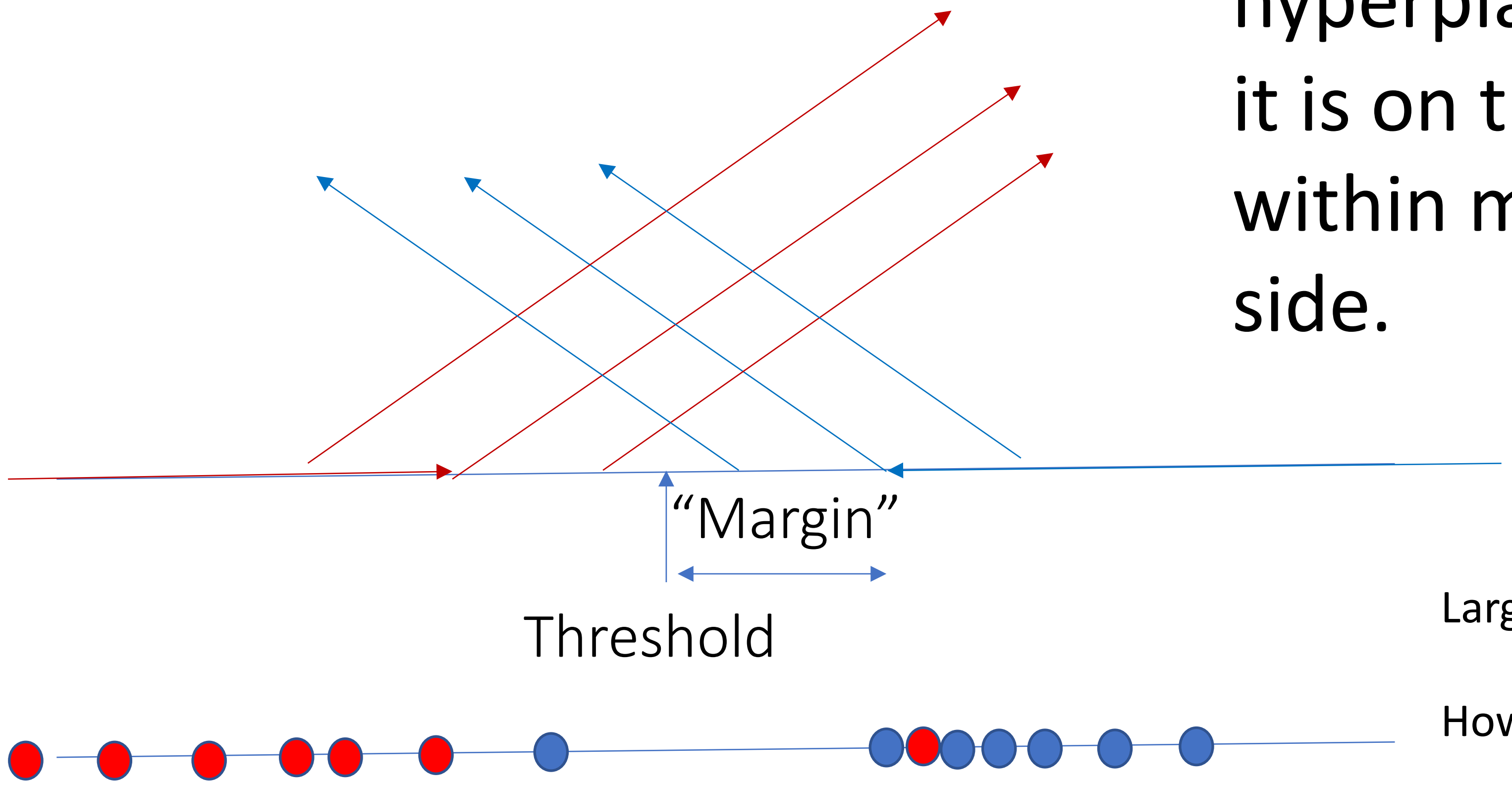


The loss function counts the absolute value of the distance to the threshold hyperplane, but only when it is on the wrong side, or within margin of the right side.

How to set margin?

Hinge loss function

The loss function counts the absolute value of the distance to the threshold hyperplane, but only when it is on the wrong side, or within margin of the right side.



Large margins look at more points.

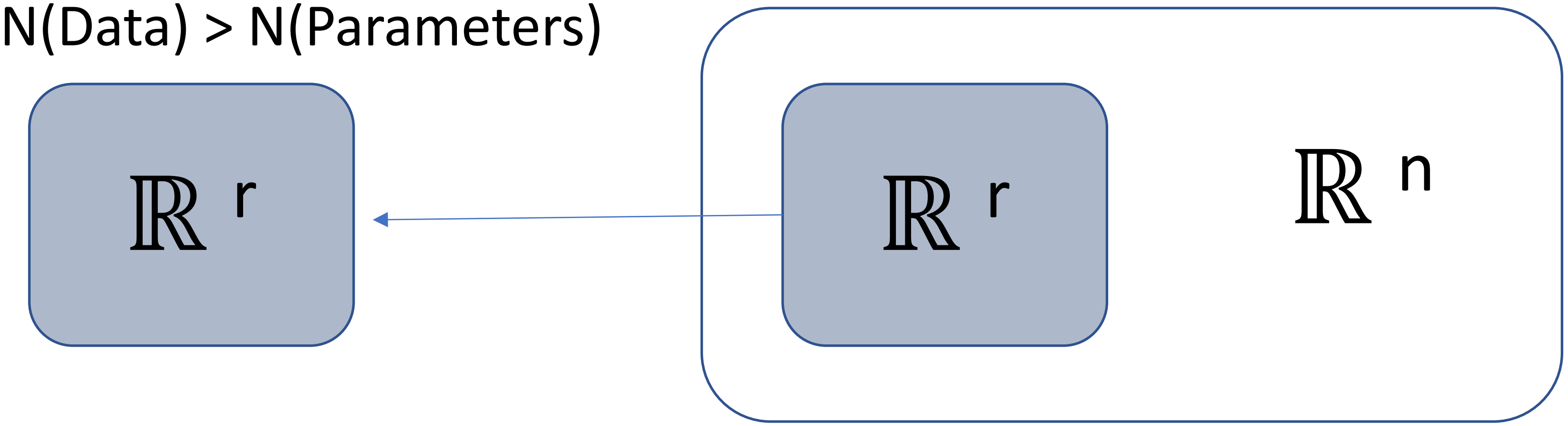
How to set margin?

Parameters

Observations

Overdetermined
Underparameterized
Normal ”fitting”,
least-squares
or not, is like this.

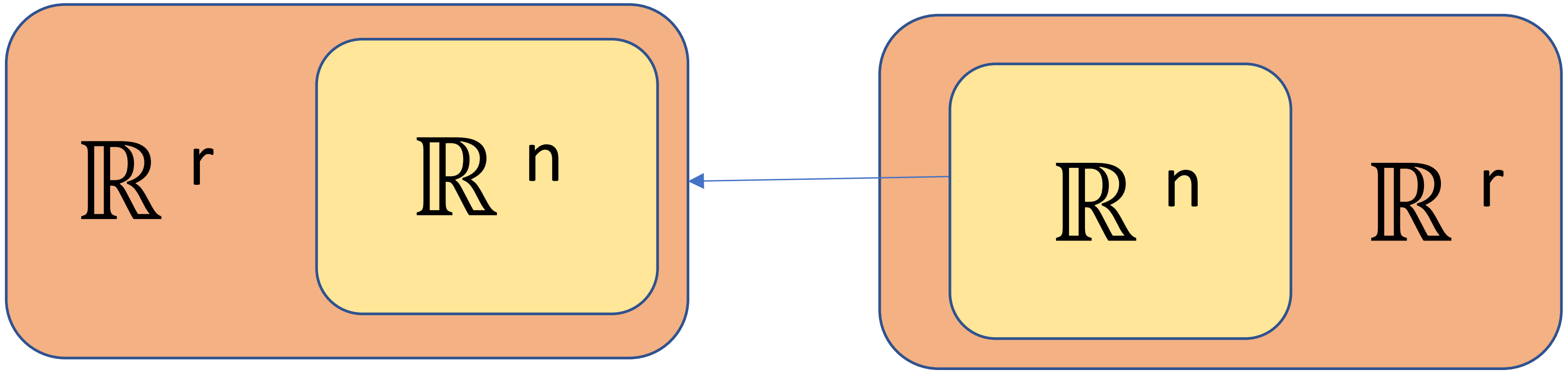
N(Data) > N(Parameters)



Underdetermined
Overparameterized

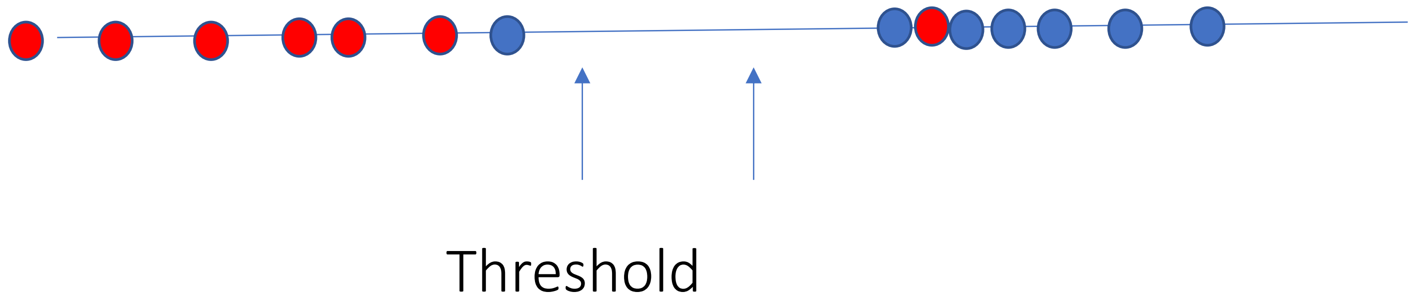
Iterating over n
is easier than
iterating over r

N(Parameters0) > N(Data)



Many dimensions must
be filled with information
from prior / regularization

You worry too much about loss functions.
Set threshold for maximum accuracy?
Doesn't work very well…



Demo