Statistics and Data Science 365 / 565

Data Mining and Machine Learning

February 15

Yale

Outline

- Recap: supervised learning goal
- Recap: Function classes
- Classification
- Surrogate Losses
- Population Risk
- Notebook

Recap: Supervised learning goals

hat means estimate of sth

Create a function $\widehat{f} : \mathcal{X} \mapsto \mathcal{Y}$ by "learning" from data $z_i = (x_i, y_i)$.

Probably want $\widehat{f}(x_i) \approx y_i$ (Notation: $\approx$ means approximately)

Regression: $\mathcal{Y}$ is continuous/ordered (prices)

Classification: $\mathcal{Y}$ is discrete/unordered (labels: hot dog v not hot dog)

Focus on classification today.

Recap: Function Classes 函数的类型

① **Linear functions:** Take $x \in \mathbb{R}^d$

$$\mathcal{F} = \{f \mid f(x) = \theta^T x \text{ for some } \theta \in \mathbb{R}^d\}$$

(weights: θ; features: x)

Learning: Optimize loss for weights

Meaning of weight: how much a change in input will affect change in output

sometimes weight and features are correlated (colinearity)

② **Nearest neighbors:** Make predictions based on neighbors

Learning: ① Training example proximity. ② Number of neighbors. ③ Choice of distance.

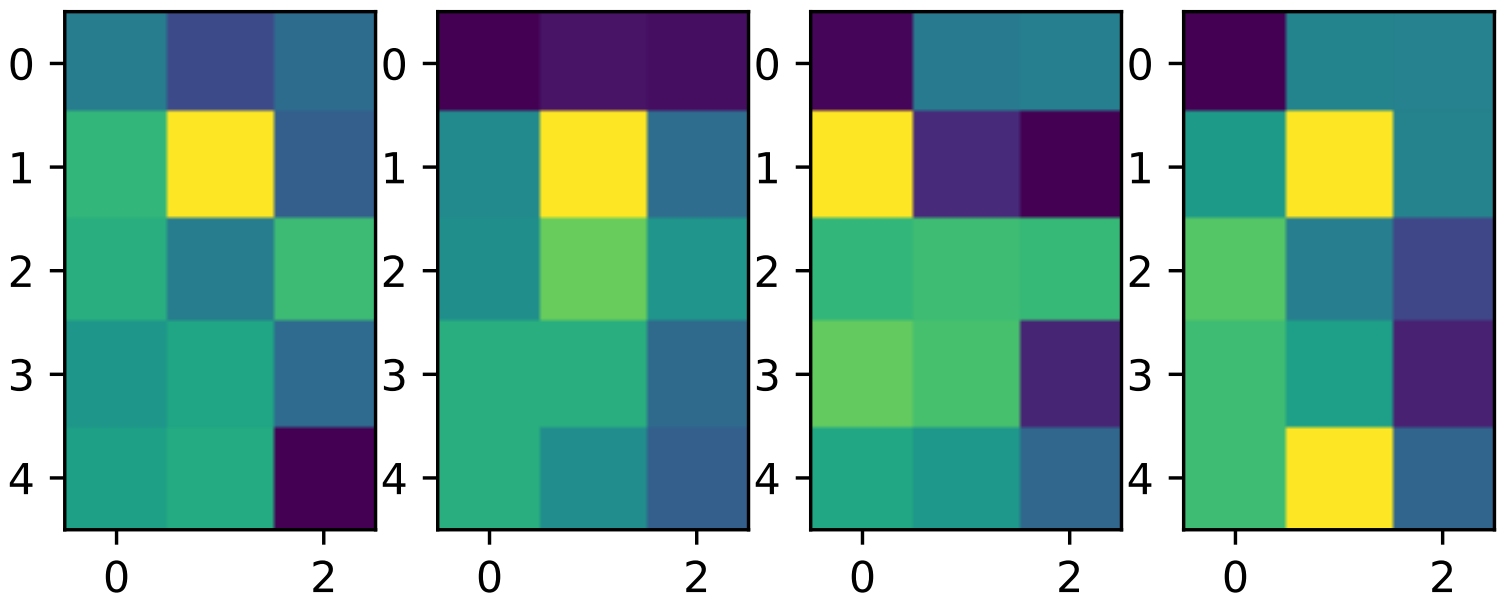
(① who were the k nearest neighbors; ② k; ③ Often use Euclidean distance (l2 norm))

Linear function weights: classification

$y = -1$ one label

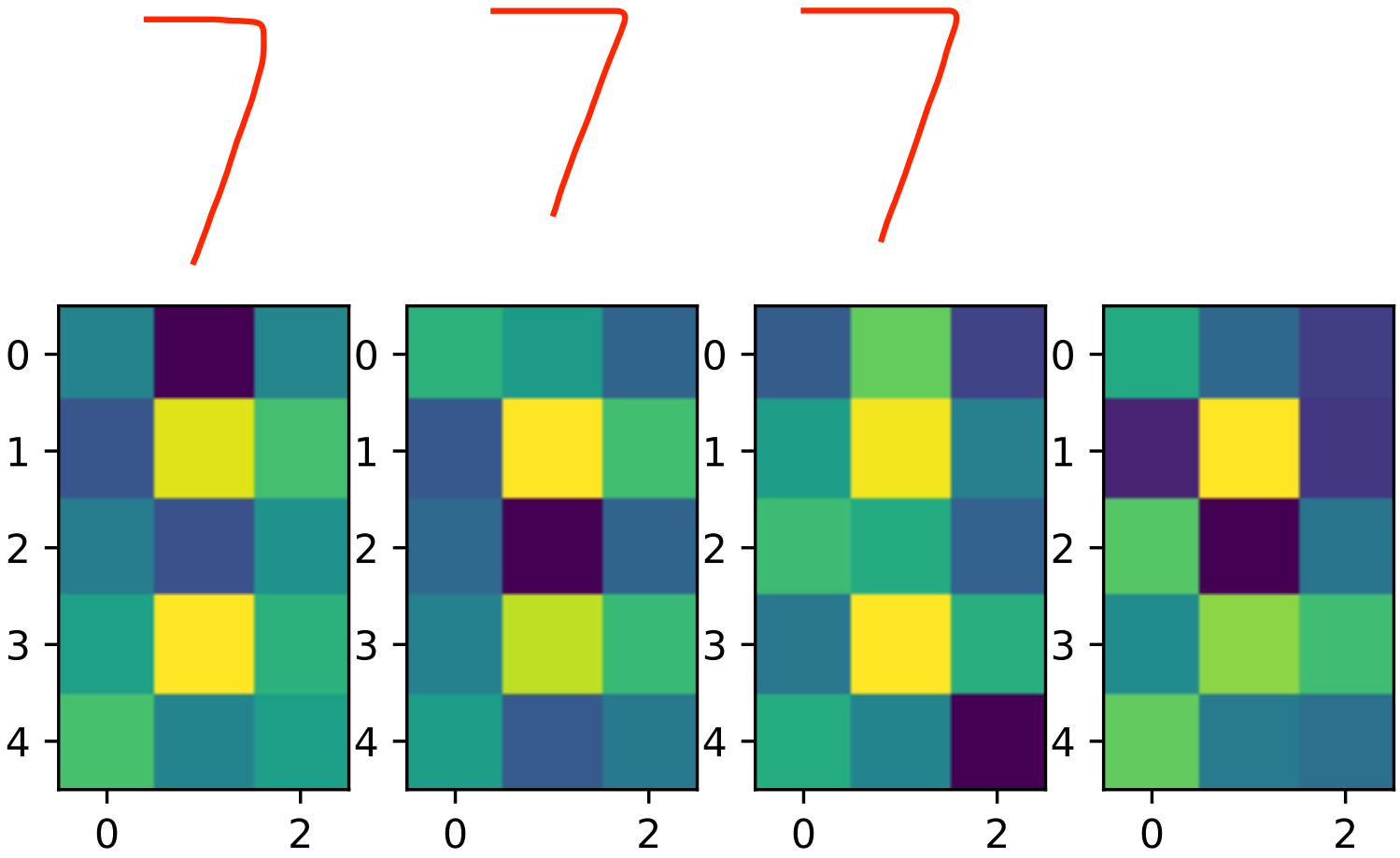
data set: images 7 and 8 with Gaussian noise

4 Eg. of X with label -1



Linear function weights: classification

$y = 1$ another label



Linear function weights

label $y = \{-1, +1\}$ $\hat{y}_i = 5.5 > 1$ → +1 label

Discriminative learning R → binary classification

x θ weight

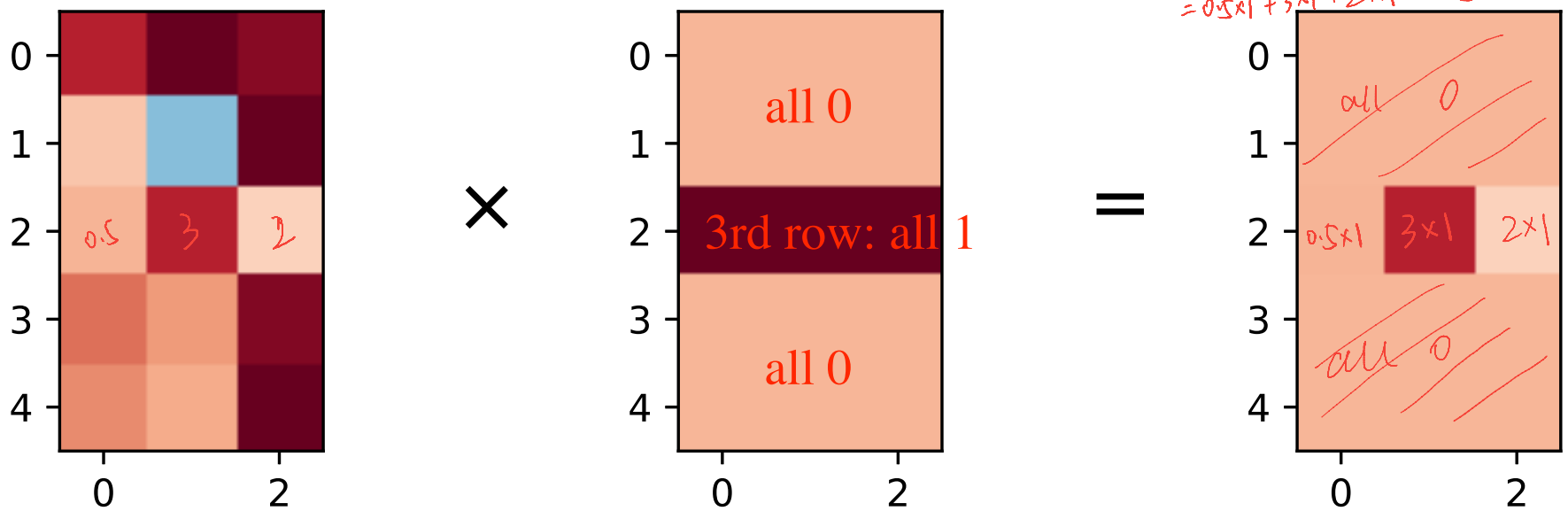
scale x by weight θ

point-wise multiplication $x \cdot \theta$

$\sum x_i \theta_i$ = a real number

$= 0.5 \times 1 + 3 \times 1 + 2 \times 1 = 5.5$

5x3 pixels = 15 dimensions



Linear function weights

This is a classification problem.

Above choice has mean Hamming error 0.128. Can we do better?

find optimal value of Hamming error

if $f(x)$ is linear function $\Rightarrow$ square loss $\neq$ Hamming loss

error	y	$f(x)$	interpretation
81	+1	too positive (10)	$f(x)$ very confident y is a positive thing
121	+1	too negative (-10)	bigger error

This is a classification problem. Directly optimizing Hamming loss is hard. NP hard computational intractable

$$\text{sign}(f(x)) = \begin{cases} +1 & f(x) > 0 \\ 0 & f(x) = 0 \\ -1 & f(x) < 0 \end{cases}$$

Will just used squared loss. Add an intercept.

① Hamming loss $\Rightarrow \mathbb{1}(y \neq f(x))$ ($y \in \pm 1$, $f(x) \in \mathbb{R}$)

hard to have $y=f(x)$

$\Rightarrow$ instead $\mathbb{1}(y \neq \text{sign}(f(x)))$

easier

② Squared loss: $(y - f(x))^2$

$y = 0, \pm 1$

if $f(x)$ is discrete

error	y	$f(x)$
0	+1	1
4	+1	-1
4	-1	1

square loss $\div 4$ = Hamming loss

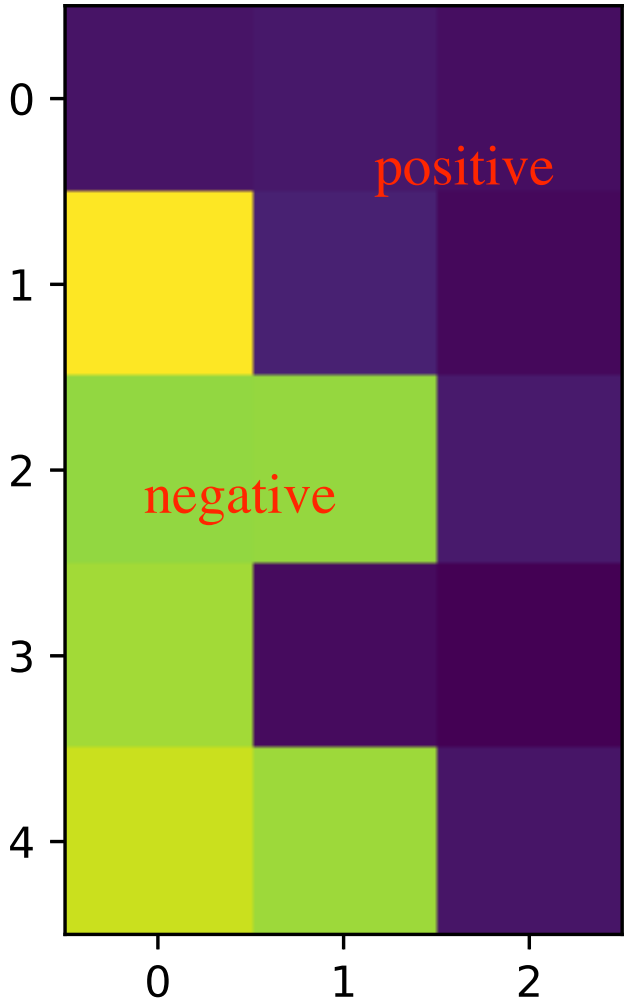
argmin don't care scaling

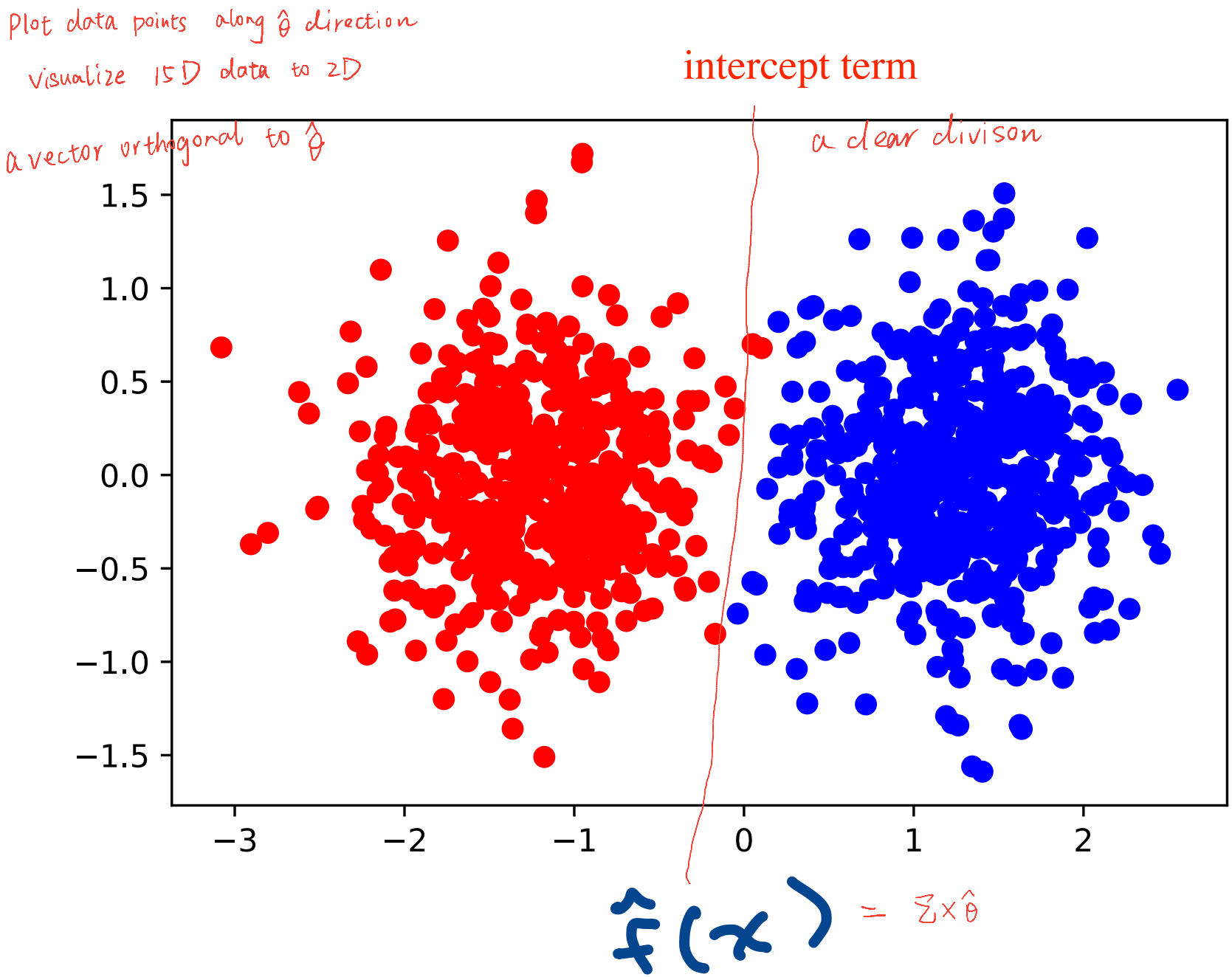
Achieves 1% training error

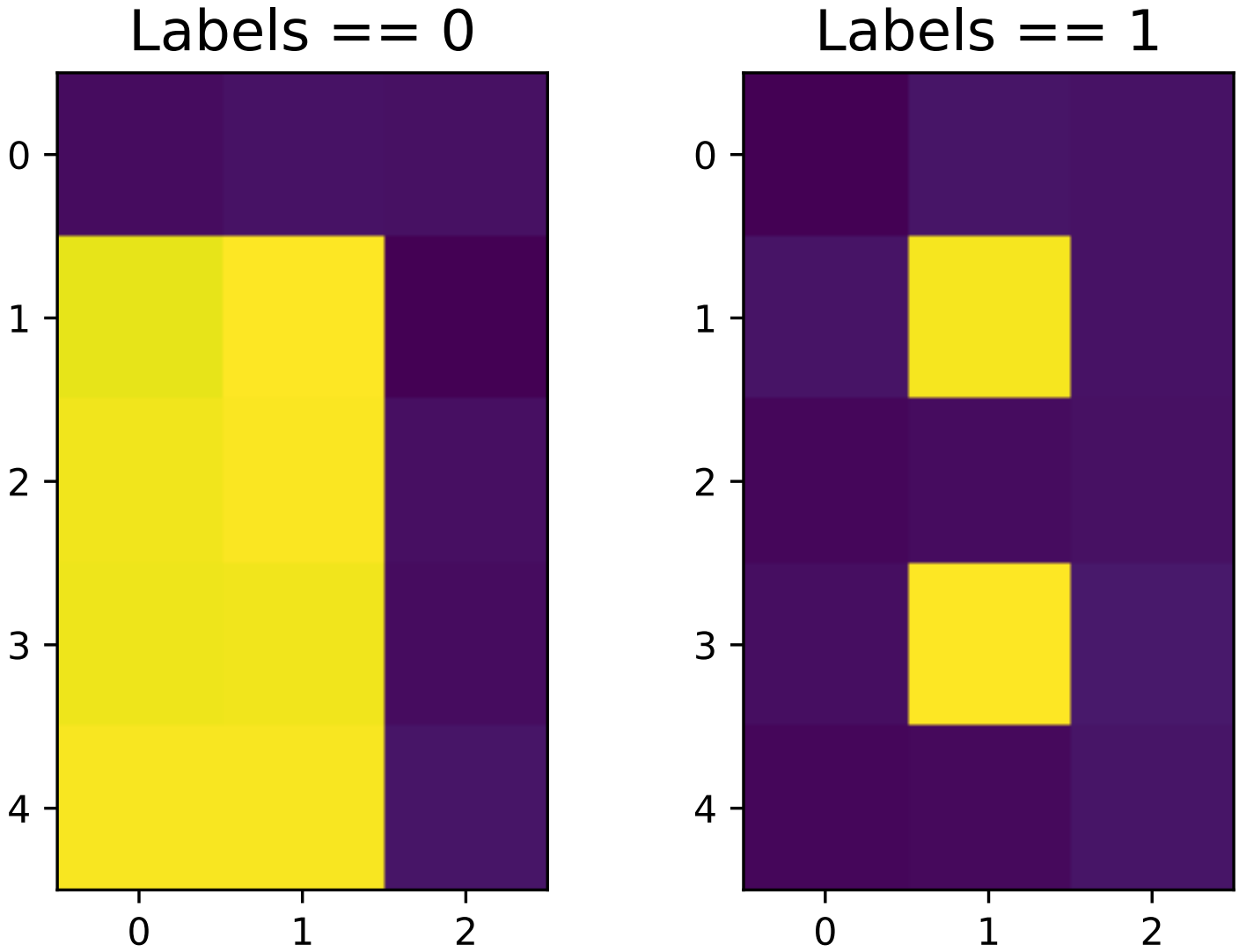
(Hamming loss)

Optimize square loss can also achieve very small (nice) Hamming loss

$\hat{\theta}$




Plot data points along $\hat{\theta}$ direction
visualize 15D data to 2D
intercept term
a vector orthogonal to $\hat{\theta}$
a clear divison
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−3
−2
−1
0
1
2
$f(x) = \Sigma x\hat{\theta}$



average of data point with label == 0 is image 7

Classification

Directly optimizing Hamming loss is hard

Also not always a great idea

Rely on surrogate losses

easier to select parameters

- squared loss above
- logistic loss
- hinge loss
- boosting loss

Surrogate Losses

Let's take $\mathcal{Y} = \{-1, +1\}$

$= \mathbb{1}(y \neq \text{sign}(f(x)))$

Hamming loss: $\mathbb{1}(y \neq f(x)) = \mathbb{1}(f(x)y \leq 0)$. Why? $f(x)$和y异号说明 $f(x) \neq y$

← optional: often <

squared loss: $(y - f(x))^2 = y^2(1 - f(x)y)^2$

若y不是binary而是 continous

$(y - f(x))^2$

$= [y(1 - \frac{f(x)}{y})]^2$

if $f(x) = \{\pm 1\}$, then sq loss = Ham loss

logistic loss: $\log(1 + \exp(-2f(x)y))/\log(2)$

if $f(x) = \{-1, +1\}$, then logistic loss = hamming loss

use to plot nice figure,

arg min don't care scaling 2, log(2)

hinge loss: $(1 - yf(x))_+$ Notation:

if $f(x) = \{-1, +1\}$, then hin loss = Ham loss

$$(v)_+ = \begin{cases} v & v > 0 \\ 0 & v \leq 0 \end{cases}$$

positive parts of a number

boosting loss: $\exp(-yf(x))$

Surrogate Losses

All of the losses above function of $f(x)y$

In classification could learn function $f : \mathcal{X} \mapsto \{-1, +1\}$ Hamming loss

Hard. Usually learn $f : \mathcal{X} \mapsto \mathbb{R}$ surrogate loss

Then take $\text{sign}(f)$

margin: $f(x)y$–how correct or wrong very positive: very correct very negative: very wrong $f(x)$与y同号 异号

If $\mathcal{Y} = \{0, 1\}$ then take $\mathbb{1}(f(x) > 0)$

Linear logistic regression

to solve binary classification

most popular way

$\hat{f}: X \to \mathbb{R}$

optimize over θ

$$\arg\min_{\theta} \sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T \theta))$$

logistic loss

Function class is linear functions

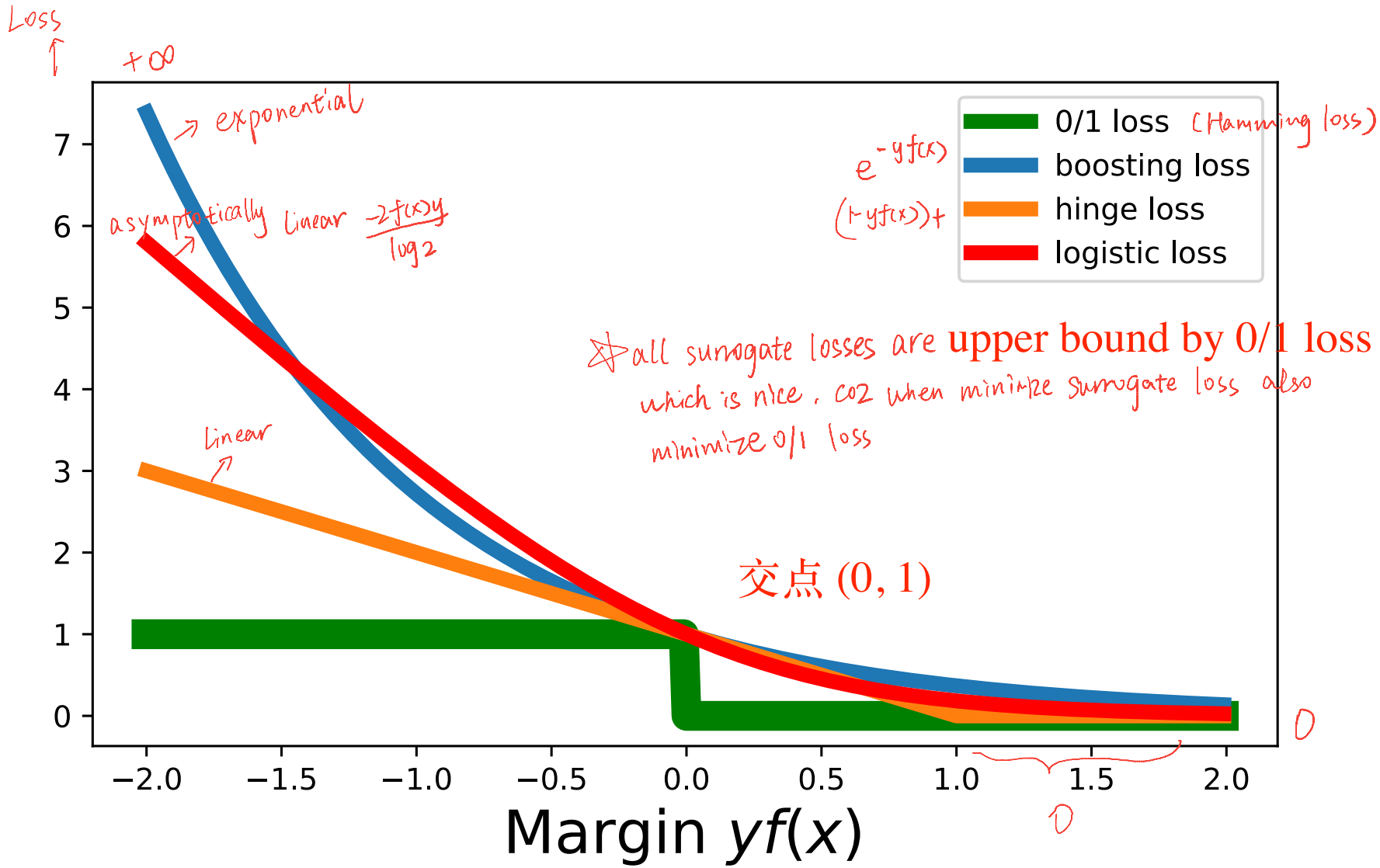
Easier to optimize over the above than to optimize

$$\arg\min_{\theta} \sum_{i=1}^{n} \log(1 + \exp(-y_i \text{sign}(x_i^T \theta)))$$

non-linearity challenge with training neural network

no closed form solution: but can use least squares solution and other methods

Surrogate Losses



Loss motivation

of both regression and classification

Have been optimizing

$$\arg\min_{f \in \mathcal{F}} \widehat{R}(f)$$

$\mathcal{F}$: function classes
(linear function, kNN, neural network, etc)

Motivation:

$$f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}\ell(f(x), y)$$

where $x, y \sim \mathbb{P}(x, y)$

x, y 服从某个P分布

i.e. be good in expected value

Define population Risk or just Risk to be

$$R(f) = \mathbb{E}\ell(f(x), y)$$

Risk of function f is the expected loss of that function
expectation is of pop, not sample

For fixed f if data identically distributed (not necessarily independent)

randomly uniformly pick a sample from a pop

coz expectation is linear

$$\mathbb{E}\widehat{R}(f) = R(f)$$

expected value of empirical risk = pop risk

Hope that with enough data $\widehat{R}(f)$ is close to $R(f)$

Use central limit theorem, emprical process theory, Glivenko Cantelli, ...

general specific

Caveats

- Data might not be independent (false sense of security for large n) eg. google flu large n doesn't mean good estimate eg. time series data correlate with adjcent data
- Weighting data all the same (bias against minorities)
- Choice of loss might be inappropriate for problem (robustness) ↓ e.g. Covid positive case

actually a paper prove different choice of loss is fine

Conclusion

Classification similar to regression, but with different losses

Use surrogates instead of Hamming loss

Empirical risk aims to approximate population risk

$\hat{R}(f)$ large sample $R(f)$

if data i.d

$E[\hat{R}(f)] = R(f)$