Statistics and Data Science 365 / 565

Data Mining and Machine Learning

February 15

Yale

Outline

- Recap: Empirical risk minimization
- Geometric interpretations: least squares and margin
- Maximum likelihood
- Notebook

Recap

Minimize population risk

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

Can't do that. Do empirical risk minimization

coz don't know E(y)

$$\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

when data identically distributed

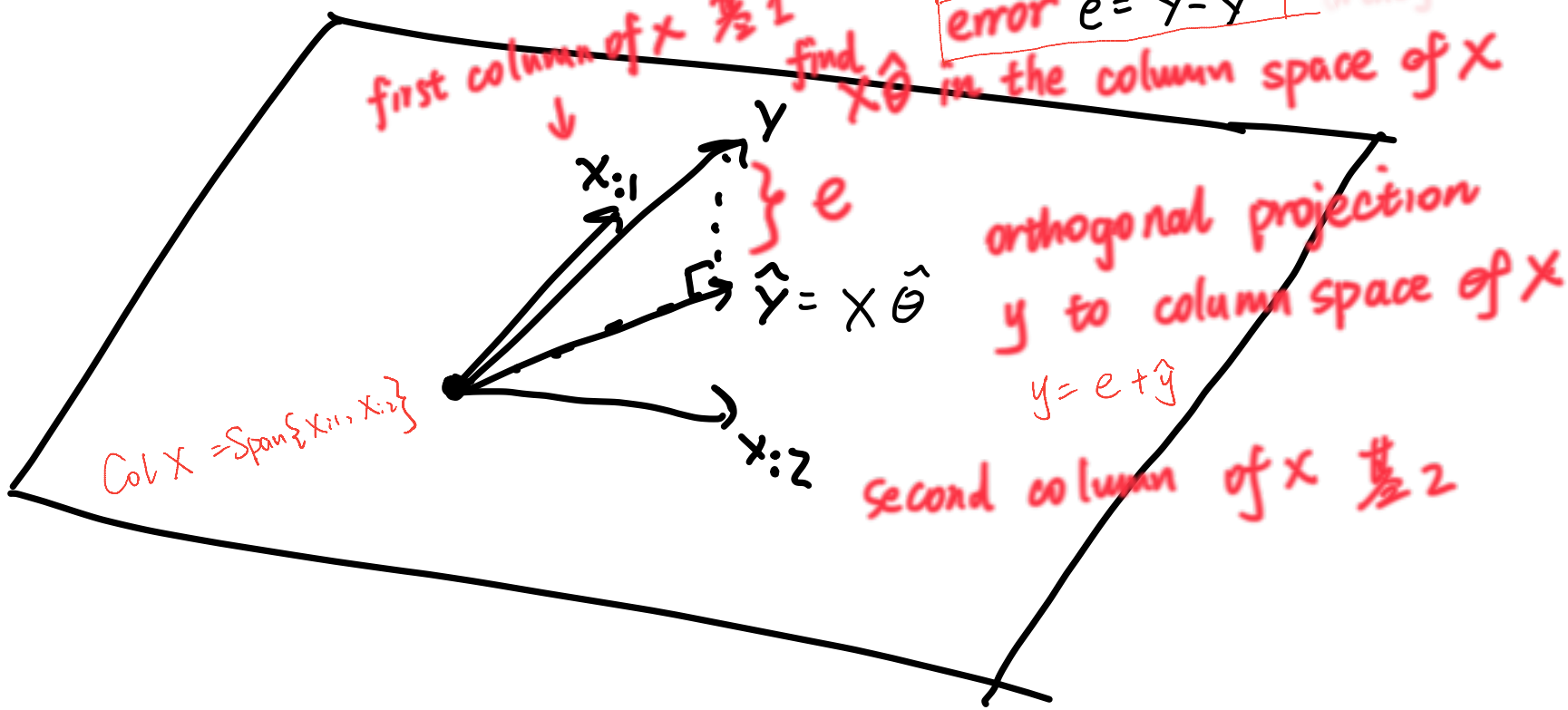
$E[\hat{R}(f)] = R(f)$

Binary classification surrogate losses: Boosting, logistic, hinge

Regression losses: squared, absolute, (there are more)

Geometric interpretation

Linear Regression: $y = X\theta + w$



Geometric interpretation: margin

focus on classification

$= y\,f(x)$ ↓ how "certain"

↓ label y $x \in \mathbb{R}^{n\times 2}$

$x_{(1)}$

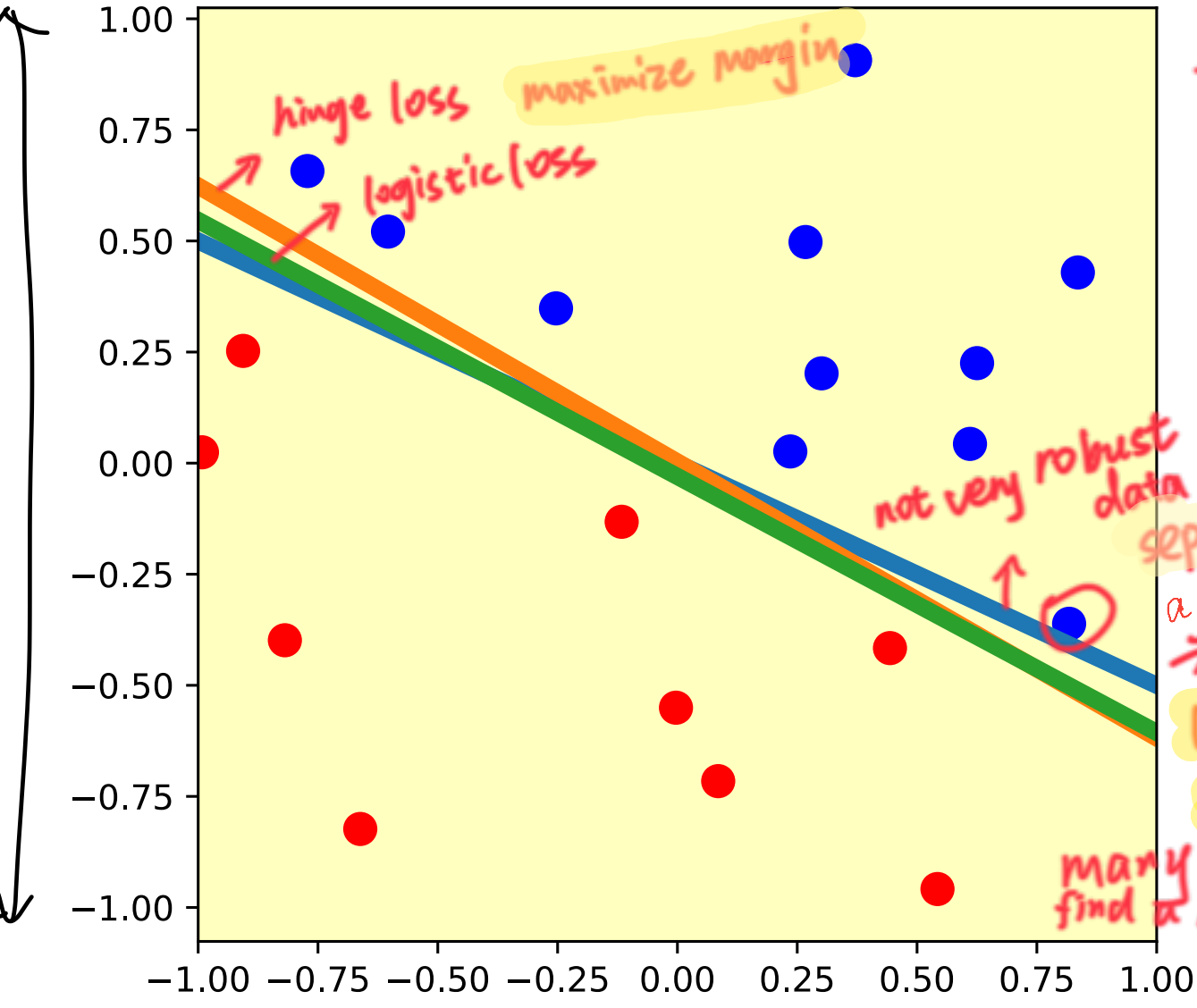
$x:1$

{ blue: +1
red: -1

label

$x:2$

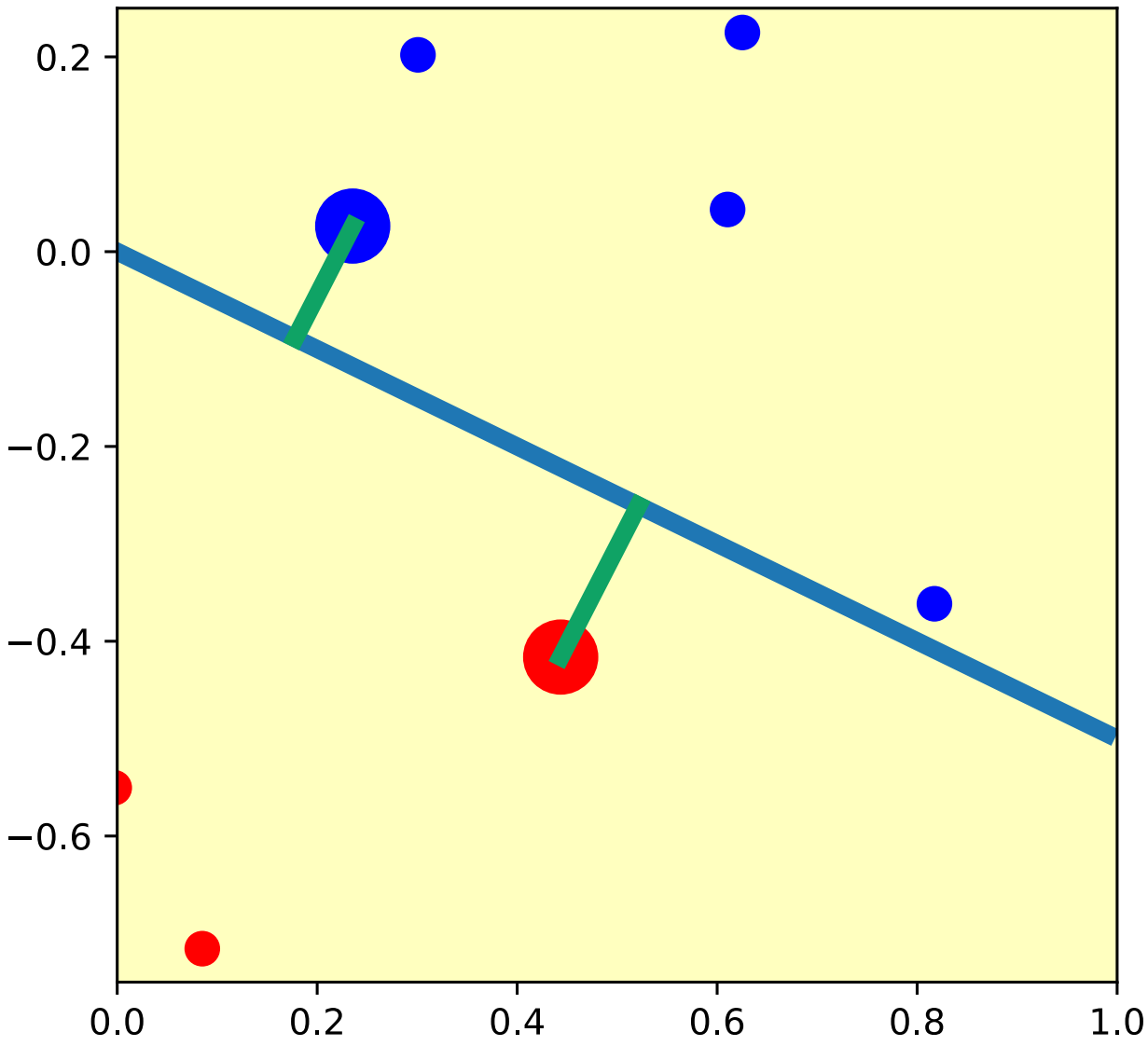
$x_{(2)}$



my function think it should be for one value or another value for binary classification

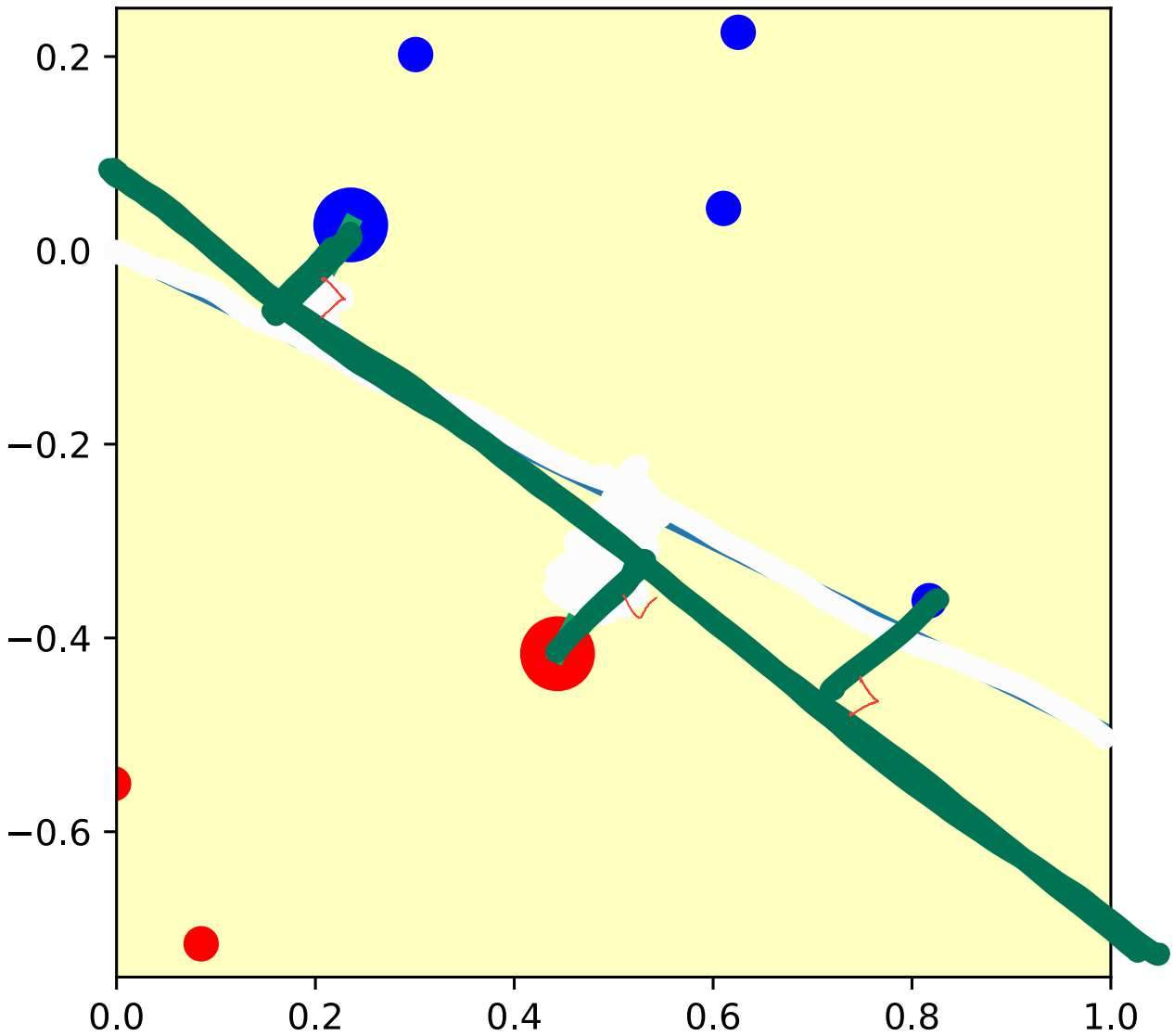
Geometric interpretation: margin

zoom in
green line: margin distance of point between decision boundry



Geometric interpretation: margin

a more prefesable decision boundry



Geometric interpretation: margin

if we don't have $\|\theta\|_2$, take θ and θ_0 very large. margin $\to +\infty$
make margin formla scaling free
decision boundary is dictated 决定 by θ, but don't change with size of θ
does change is "certainty", as θ, θ_0 gets larger means we are more certain

Take $\mathcal{Y} = \{-1, +1\}$, $x_i \in \mathbb{R}^d$, $\theta \in \mathbb{R}^d$, $\theta_0 \in \mathbb{R}$ talk calibration 校准 later

θ_{null} (θ_0): intercept of θ is real number

inner product of x_1, θ

$$\text{margin}(\theta) \equiv \min_i \frac{y_i(\langle x_i, \theta \rangle + \theta_0)}{\|\theta\|_2}$$

$\|\theta\|_2$: l2 norm of θ

Hw 3 proof for perfect separable data

$\frac{y_i(\langle x_i, \theta \rangle + \theta_0)}{\|\theta\|_2}$ is the smallest distance between any point and decision line

$\|\theta\|_2$: renormalization

Probability interpretation: classification

to be easier

$$\mathcal{Y} = \{0, 1\}$$

Hamming loss

$$R(f) = \mathbb{E}\mathbb{1}(f(x) \neq y)$$

$$f^* = \arg\min_f R(f).$$

assume we know x

$$f^*(x) = \mathbb{1}\Big(\mathbb{P}(y = 1|x) > 0.5\Big)$$

Rule: $\mathbb{E}[\mathbb{I}(\cdots)] = P(\cdots)$ expected value of indicator = Probability of ~

$$\mathbb{E}\mathbb{1}(f(x) \neq y) = \mathbb{P}(f(x) \neq y)$$

law of iterated expectation $E(Y) = E[E(Y|X)]$ 条件概率 given x

$$= \mathbb{E}\Big[\mathbb{E}[\mathbb{1}(f(x) \neq y)|x]\Big]$$

y = {0, 1}

$$= \mathbb{E}\Big[[\mathbb{1}(f(x) \neq 1)]\mathbb{P}(y = 1|x) + [\mathbb{1}(f(x) \neq 0)]\mathbb{P}(y = 0|x)\Big]$$

penalty ① ① + ② = 1 penalty ②

Focus on the inside. of E coz when inside is smallest E[] is also smallest

无论 f(x) 是否 = 1 或 0，总要 pay 2 个 penalty 中的一个，因为有 noise，可能分类错

try to pick the smallest penalty of 2 penalties eg(0.4, 0.6) pick 0.4

$$R(f) = \begin{cases} \mathbb{P}(y=1|x) & f(x)=0 \\ \mathbb{P}(y=0|x) & f(x)=1 \end{cases}$$

Probability interpretation: classification

$f^* = \arg\min_f R(F)$.

$$f^*(x) = \mathbb{1}\Big(\mathbb{P}(y = 1|x) > 0.5\Big)$$

Hamming loss

$$\begin{aligned}
\mathbb{E}\mathbb{1}(f(x) \neq y) &= \mathbb{P}(f(x) \neq y) \\
&= \mathbb{E}\Big[\mathbb{E}[\mathbb{1}(f(x) \neq y)|x]\Big] \\
&= \mathbb{E}\Big[[\mathbb{1}(f(x) \neq 1)]\mathbb{P}(y = 1|x) + [\mathbb{1}(f(x) \neq 0)]\mathbb{P}(y = 0|x)\Big]
\end{aligned}$$

Focus on the inside.

In general, But

Define $\eta(x) = \mathbb{P}(y = 1|x)$. Don't have in general.

So assume true model is this θ^*: true θ

Create a model:

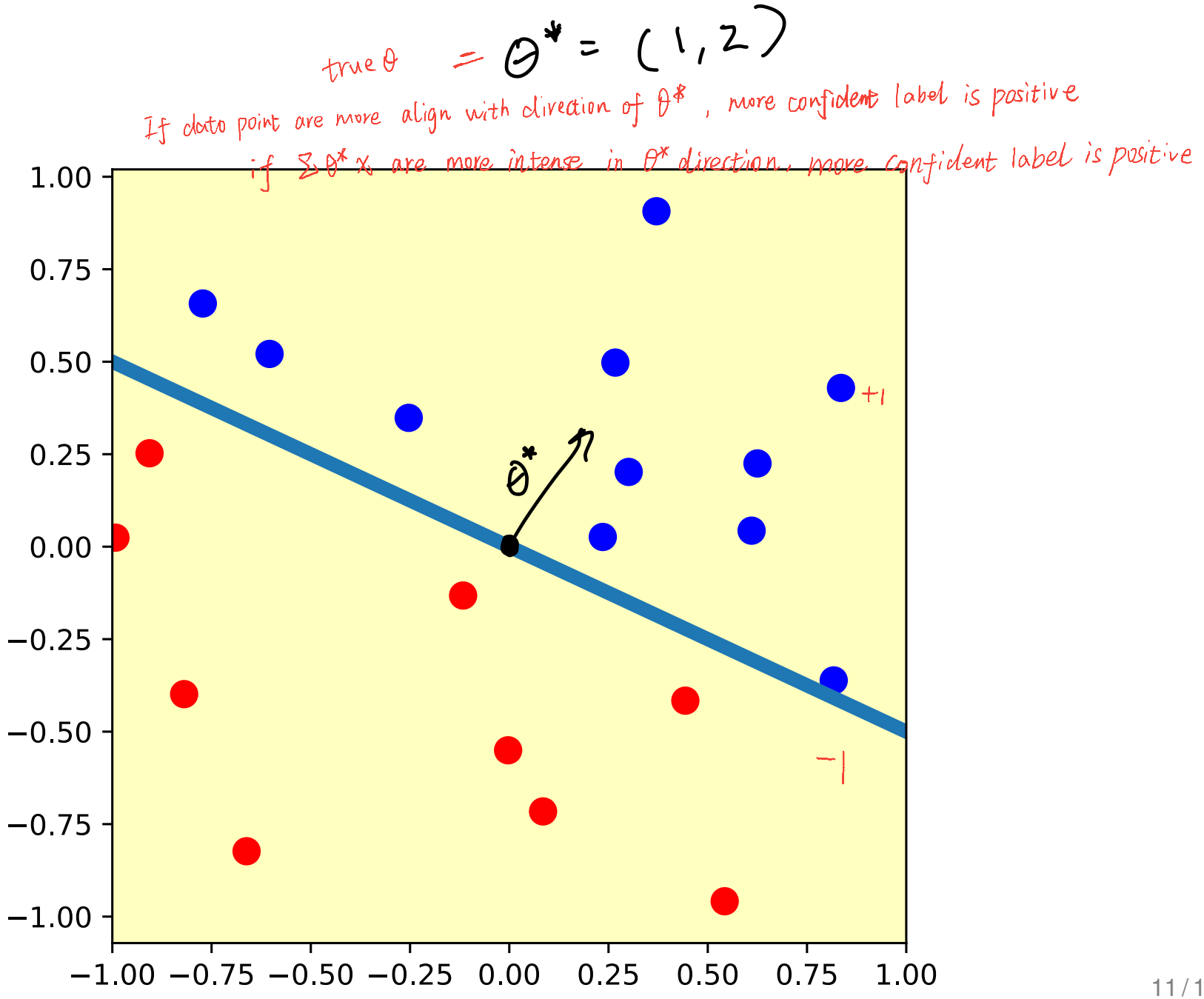
$$\eta(x) = \frac{\exp(\theta^{*T} x)}{1 + \exp(\theta^{*T} x)}$$

$\theta^{*T} x = \langle \theta^*, x \rangle$

inner product

assuming ignore intercept θ_0 or θ_0 automatically built in

Logistic


true θ = θ* = (1, 2)
If data to point are more align with direction of θ* , more confident label is positive
if Σ θ* x are more intense in θ* direction, more confident label is positive
θ*
+1
−1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

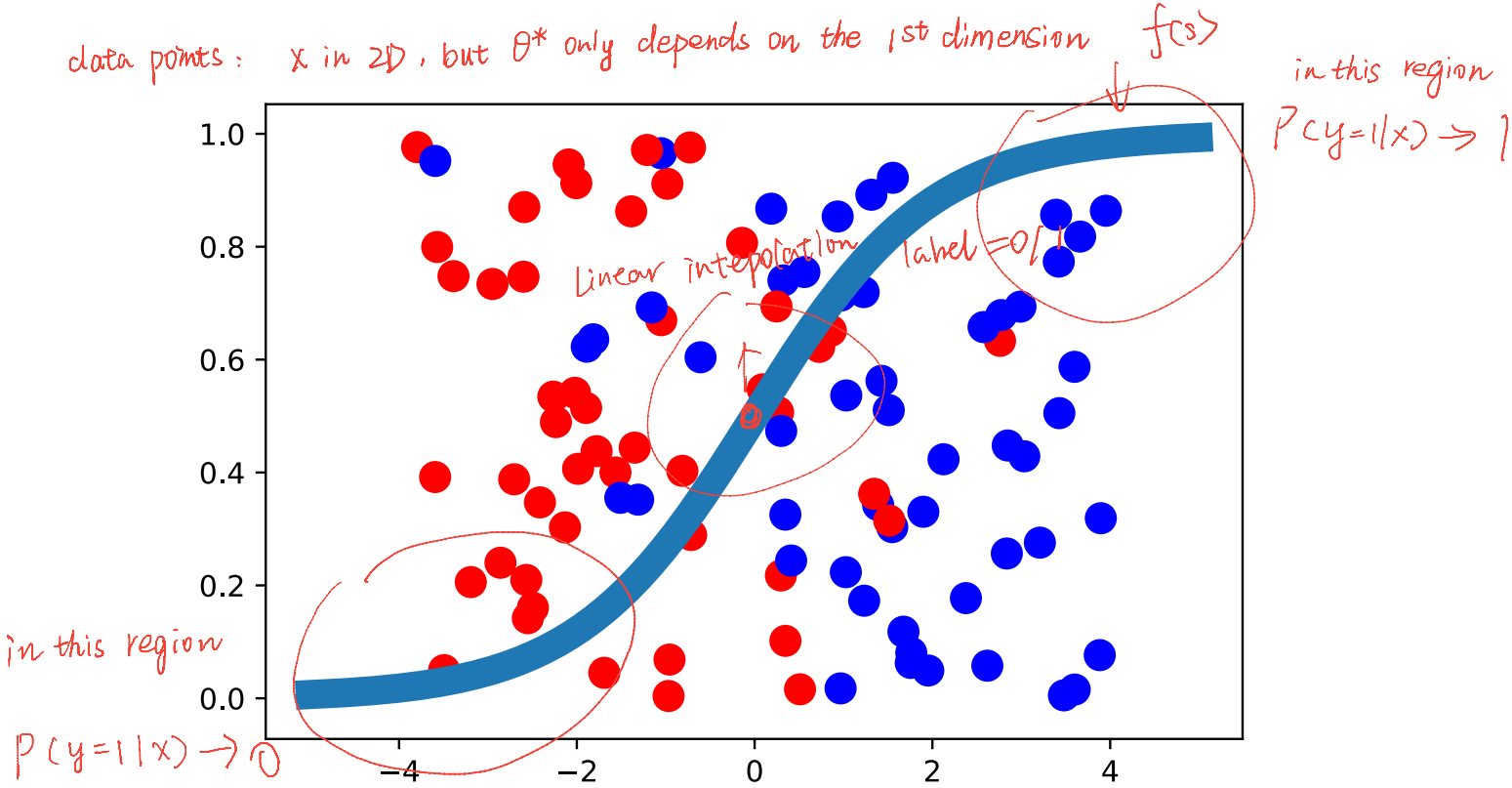
Logistic

Sigmoid function $f(s) = \dfrac{1}{1+\exp(-s)} = \dfrac{\exp(s)}{1+\exp(s)}$ $= P(y=1|x)$

probability that label $= +1$

meaning of noise

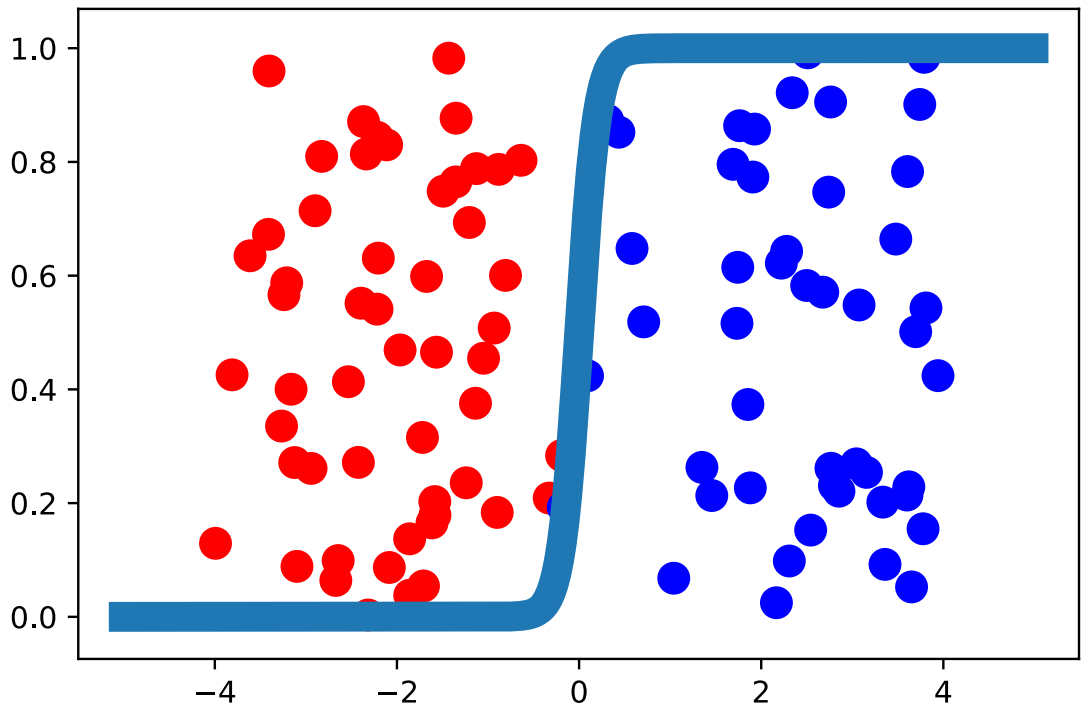
data points: x in 2D, but θ^* only depends on the 1st dimension



Logistic

Scaling will decrease noise

more certain where label actually are



PyTorch's Doc

NLLLOSS

CLASS `torch.nn.NLLLoss(weight: Optional[torch.Tensor] = None, size_average=None, ignore_index: int = -100, reduce=None, reduction: str = 'mean')` [SOURCE]

The negative log likelihood loss. It is useful to train a classification problem with *C* classes.

If provided, the optional argument `weight` should be a 1D Tensor assigning weight to each of the classes. This is particularly useful when you have an unbalanced training set.

The *input* given through a forward call is expected to contain log-probabilities of each class. *input* has to be a Tensor of size either $(minibatch, C)$ or $(minibatch, C, d_1, d_2, ..., d_K)$ with $K \geq 1$ for the *K*-dimensional case (described later).

Obtaining log-probabilities in a neural network is easily achieved by adding a *LogSoftmax* layer in the last layer of your network. You may use *CrossEntropyLoss* instead, if you prefer not to add an extra layer.

The *target* that this loss expects should be a class index in the range $[0, C-1]$ where *C = number of classes*; if *ignore_index* is specified, this loss also accepts this class index (this index may not necessarily be in the class range).

The unreduced (i.e. with `reduction` set to `'none'`) loss can be described as:

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^\top, \quad l_n = -w_{y_n} x_{n,y_n}, \quad w_c = \text{weight}[c] \cdot 1\{c \neq \text{ignore\_index}$$

where x is the input, y is the target, w is the weight, and N is the batch size. If `reduction` is not `'none'` (default `'mean'`), then

Maximum likelihood/minimum negative log-likelihood

Assumption of Logistic Regression
① X are i.i.d

Need to estimate θ^* from (assume iid) data $\{(x_i, y_i)\}$

② β 服从 Bernoulli (p_i)

Candidate parameter β ← $\hat{\theta}$

$P(y=1|x)$ for logistic $f = f(s)$

Example i we have guess $p_i = f(x_i^T \beta)$. Likelihood of guess (also called Loss)

probability Mass Function of Bernoulli Distribution

assume β is independent of x_i

$$L(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} = \prod_{i=1}^{n} P_\beta(Y_i = y_i | x_i)$$

i.i.d product

probability { pdf continuous data; pmf discrete data

$= P_\beta(\{x_i, y_i\}_{i=1}^n)$

$= P(\{x_i, y_i\}_{i=1}^n | \{x_i\}_{i=1}^n, \beta)$

$= P(\{x_i, y_i\}_{i=1}^n | \{x_i\}_{i=1}^n, \beta)$

$L(\beta)$: probability of my observing data as I have seen if β was the true parameter

find β maximize that probability ($L(\beta)$)

Negative log-likelihood

i.e. minimize NLL

$$= \begin{cases} \prod_{i=1}^{n} p_i & (y_i = 1) \\ \prod_{i=1}^{n} (1-p_i) & (y_i = 0) \end{cases}$$

$$\ell(\beta) = \sum_{i=1}^{n} \underbrace{-y_i \log(p_i) - (1-y_i)\log(1-p_i)}_{\text{This is the } \textit{loss for example } i}$$

Let

$$\widehat{\theta} = \arg\min_{\beta} \ell(\beta)$$

(MLE)

$\hat{\theta}$ is the maximum likelihood estimate for θ^* (true model)

Max likelihood/min negative log-likelihood fits into loss/risk framework

$$\frac{\exp(\widehat{\theta}^T x)}{1 + \exp(\widehat{\theta}^T x)}$$

Not necessarily a good estimate for $\mathbb{P}(y = 1|x)$ i.e. not good

Calibration → how good is my estimate as a estimate of the true probability

Good classification performance

Logistic Regression

In logistic model, i.e. sigmoid function $\mathbb{P}_\beta(y=1|x) = \frac{\exp(x^T\beta)}{1+\exp(x^T\beta)} \Rightarrow \mathbb{P}_\beta(y=0) = \frac{1}{1+\exp(x^T\beta)}$

$L(\beta)$ 可以简化

$$\mathbb{P}_\beta(y|x) = \frac{\exp(yx^T\beta)}{1+\exp(x^T\beta)} = \begin{cases} \frac{\exp(x^T\beta)}{1+\exp(x^T\beta)} & y=1 \\ \frac{1}{1+\exp(x^T\beta)} & y=0 \end{cases}$$

(Next time:) $P(y|x) = \frac{\exp(y\,g(x))}{1+\exp(g(x))}$

Likelihood

$$\prod_{i=1}^{n} \frac{\exp(y_i x_i^T\beta)}{1+\exp(x_i^T\beta)} = \prod_{i=1}^{n} \left(\frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)}\right)^{y_i} \times \left(\frac{1}{1+\exp(x_i^T\beta)}\right)^{1-y_i}$$

Log-likelihood

loss for label $y \in \{0,1\}$

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

Last lecture for $\mathcal{Y} = \{-1, +1\}$ logistic loss

$$\ell(\beta^T x, y) = \log(1 + \exp(-2y\beta^T x))/\log(2)$$

same as log likelihood

See supplementary notes `Logistic Regression.ipynb` to verify they are (up to scaling) the same

Conclusion

Logistic regression as maximum likelihood estimation

Given features x_i, the larger probability of get y_i ($P(y_i|x_i)$), the more accurate our model

Geometric interpretation of least squares and margin

Next time:

generative modeling

non-linear decision boundaries

exponential families Lec 7

optimization