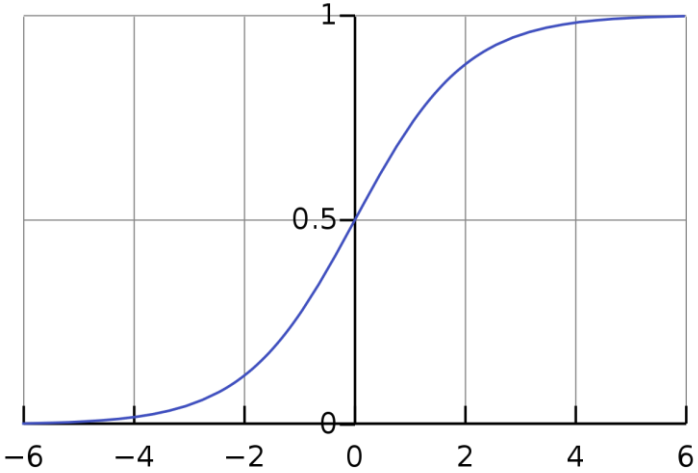
# Logistic Regression

JrPhy

# Introduction

- If the dataset is linear separable, then the dataset can be separate perfectly. But how about the noisy data ?
- Suppose a data $x$, the correct probability is $P(1|x) = 0.9$, incorrect probability is $P(-1|x) = 0.1$
- The score is not 0 or 1, instead of $0 \leq \text{score} \leq 1$, so we use a flexible function to determine, it's logistic regression.

$$h(s = w^T x) = \frac{e^{w^T x}}{1 + e^{w^T x}} = \frac{1}{1 + e^{-w^T x}}$$



# Property

- Here we also want to do binary classification, but use a probability to determine it is 1 or 0. Suppose $f(x) = P(1|x)$, then $f(y) = P(-1|x) = 1 - P(1|x)$, and use logistic function

$$P(1|x) = \frac{e^{w^T x}}{1+e^{w^T x}} = \frac{1}{1+e^{-w^T x}}$$

$$P(0|x) = 1 - P(1|x) = 1 - \frac{e^{w^T x}}{1+e^{w^T x}} = \frac{1}{1+e^{w^T x}} = -P(-1|x)$$

# Cross-Entropy Error

- Consider a dataset D = {($x_1$,o), ($x_2$,x),…, ($x_i$,x)}, $f(x_i) = 0$ or 1, suppose $f(s_i) \sim h(s_i)$,is called likelihood, so the probability is

$$P(x_1) f(s_1) \times P(s_2)(1 - f(s_2)) \times \ldots \times P(x_n)(1 - f(s_n))$$

$$\sim P(x_1) f(s_1) \times P(s_2)(1 - h(s_2)) \times \ldots \times P(x_n)(1 - h(s_n))$$

$$= P(x_1) f(s_1) \times P(s_2) h(-s_2) \times \ldots \times P(x_n) h(-s_n)$$

$$= \prod_{i=1}^{n} P(x_i) h(y_i x_i) = \prod_{i=1}^{n} P(x_i) h(y_i w^T x_i)$$

- Next step is to maximize $\boldsymbol{w}$

# Optimize

- But it's hard to calculate the maximum, so we take log before it

$$\max_w \prod_{i=1}^{n} P(x_i)\theta(y_i x_i) \rightarrow \max_w \prod_{i=1}^{n} \theta(y_i w^T x_i) \rightarrow \max_w \left( \ln \prod_{i=1}^{n} \theta(y_i w^T x_i) \right)$$

$$\max_w \left( \ln \prod_{i=1}^{n} \theta(y_i w^T x_i) \right) = \max_w \left( \ln \theta(y_1 w^T x_1)\theta(y_2 w^T x_2)...\theta(y_n w^T x_n) \right)$$

$$= \max_w \left( \sum_{i=1}^{n} \ln \theta(y_i w^T x_i) \right) = \min_w \left( \sum_{i=1}^{n} -\ln \theta(y_i w^T x_i) \right)$$

$$\boxed{= \min_w \left( \sum_{i=1}^{n} \ln \left( 1 + \exp(-y_i w^T x_i) \right) \right)} \boxed{= \min_w E_{in}(w, x_i, y_i)}$$

# Optimize

$$\nabla_i E_{in}(w_i, x_i, y_i) = \frac{\partial}{\partial w_i} \ln(1 + \exp(-y_i w_i^T x_i)) = \frac{\exp(-y_i w_i^T x_i)}{1 + \exp(-y_i w_i^T x_i)} (-y_i x_i)$$

$$\nabla E_{in}(w_i, x_i, y_i) = \frac{1}{N} \sum_{i=1}^{N} \theta(-y_i w^T x_i)(-y_i x_i) = 0$$

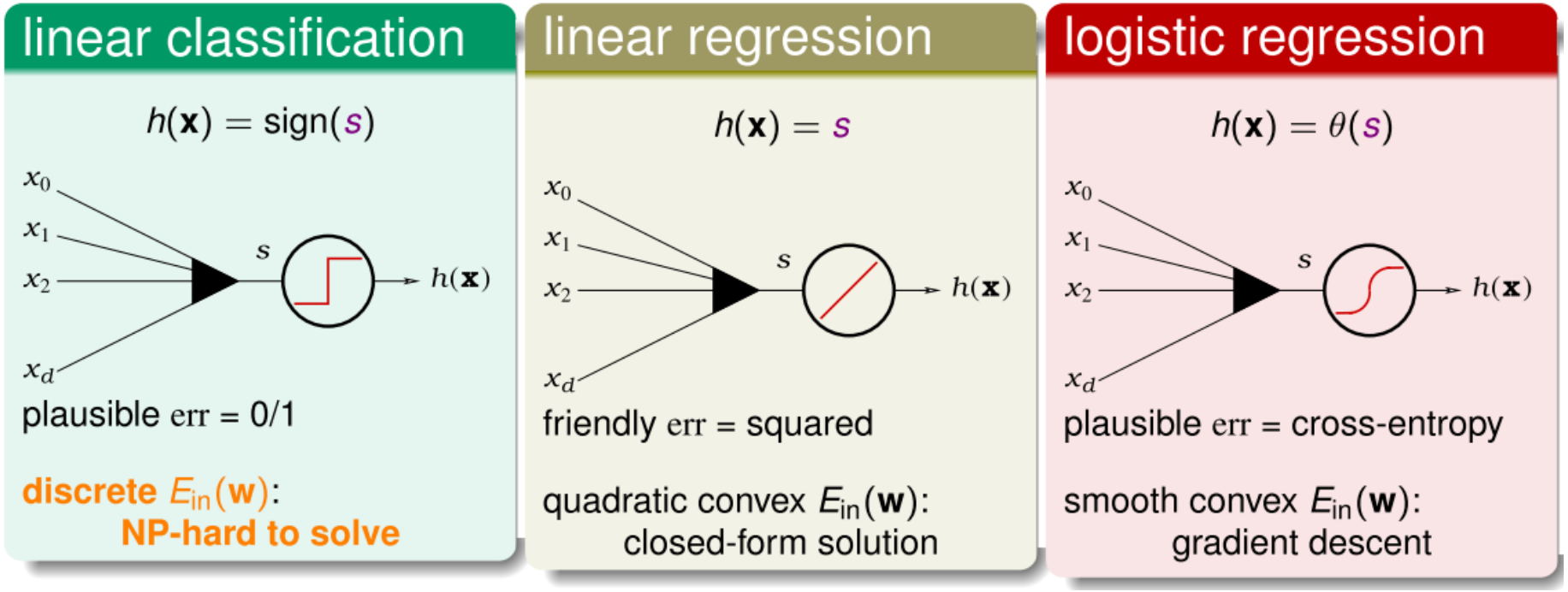
- It's hard to prove that there is only one extreme value and it's minimum the, but it's a truth, so I just use it and find the w such that the $E_{in}$ is minimized.
- So we can apply gradient descent on this problem.

# Optimize

- Let's see where the gradient will equal to 0, by the property of exp, only when $y_i w^T x_i >> 0$, then $\exp(- y_i w^T x_i) \sim 0$, this means the dataset is linear separable.
- The other possibility is the summation equals to 0, but this means the dataset is not linear separable, and it's not the linear function, so that we can just find the approximated solution.

# Regression for classification

- So far I've introduced linear classification, linear regression, and logistic regression, what I want to do is classification, so can regression help us classify?

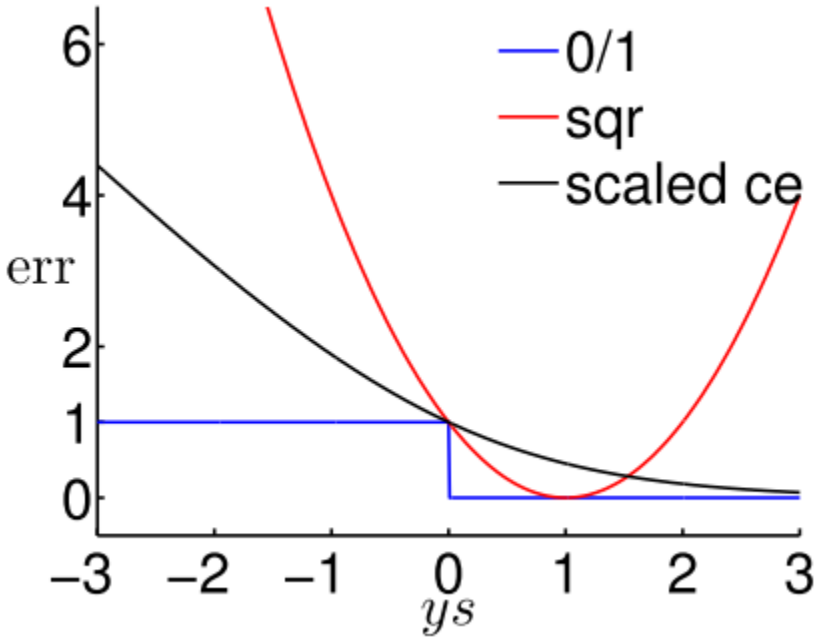



Hsuan Tien Lin, mlfound/211_handout

# Regression for classification

- Let's plot *error* versus *ys*, so that we can see the error of logistic regression is smaller as *ys* is bigger, by using gradient descent, we can find the error approach to 0.

$$
\begin{aligned}
0/1 \quad & \mathrm{err}_{0/1}(s,y) = [\![\mathrm{sign}(ys) \neq 1]\!] \\
\mathrm{sqr} \quad & \mathrm{err}_{\mathrm{SQR}}(s,y) = (ys-1)^2 \\
\mathrm{ce} \quad & \mathrm{err}_{\mathrm{CE}}(s,y) = \ln(1+\exp(-ys)) \\
\text{scaled ce} \quad & \mathrm{err}_{\mathrm{SCE}}(s,y) = \log_2(1+\exp(-ys))
\end{aligned}
$$



# Regression for classification

- So in practical problems, dataset is usually not linear separable, so logistic regression is most using for classification.

**PLA**

- pros: **efficient + strong guarantee if lin. separable**
- cons: works only if lin. separable, otherwise needing **pocket** heuristic

**linear regression**

- pros: **'easiest' optimization**
- cons: loose bound of $\text{err}_{0/1}$ for large $|ys|$

**logistic regression**

- pros: **'easy' optimization**
- cons: loose bound of $\text{err}_{0/1}$ for very negative $ys$

# Square error for logistic regression

- Why doesn't logistic regression use square error? Let's apply square error on it and take a look its property.

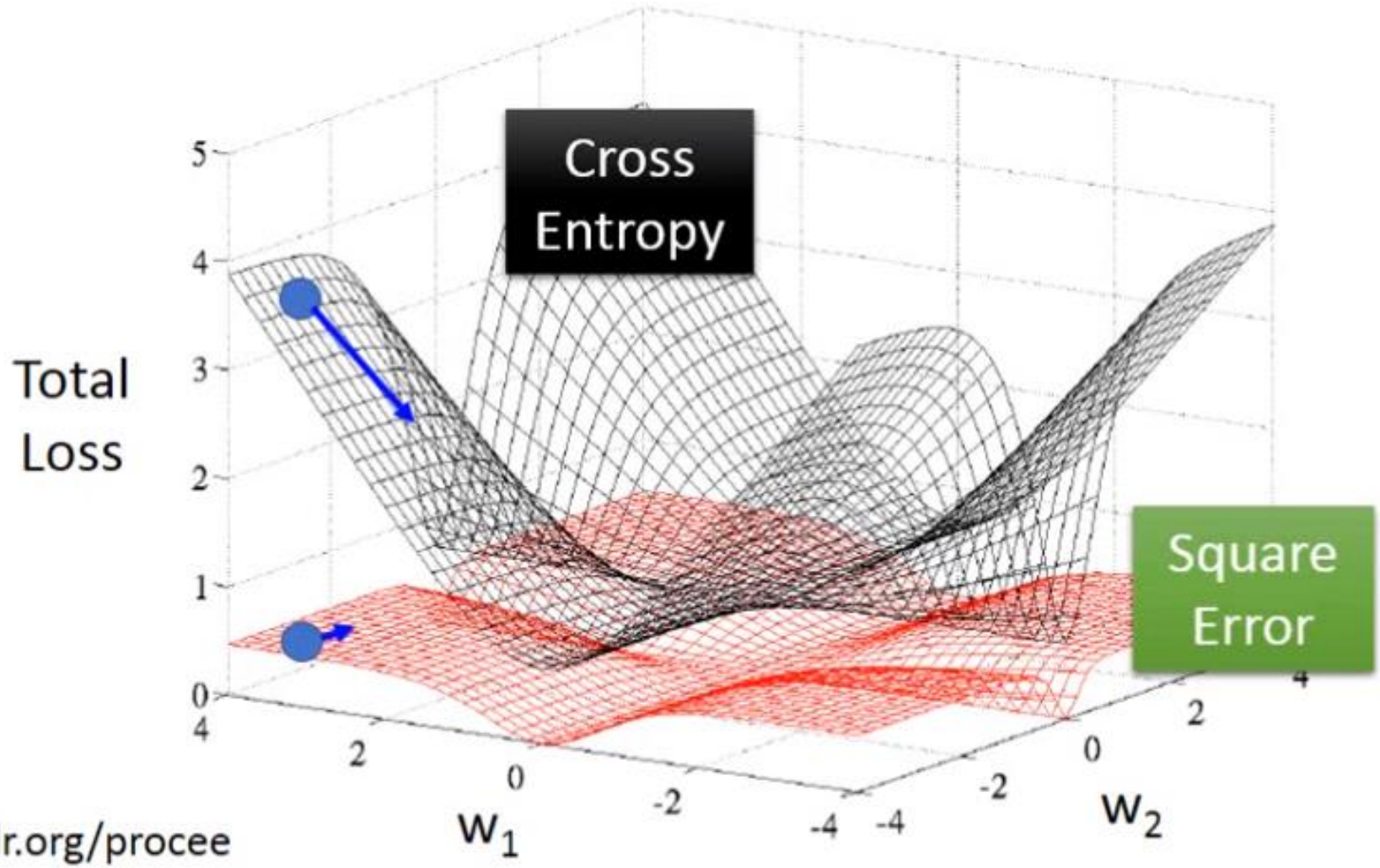
$$\theta(s) = \theta\left(\sum_{i=0}^{n} w_i x_i\right) \qquad E(w) = \frac{1}{2}\left[\hat{y} - \theta\left(\sum_{i=0}^{n} w_i x_i\right)\right]^2$$

$$\frac{\partial E(w)}{\partial w} = \left[\hat{y} - \theta\left(w^T x\right)\right]\left[1 - \theta\left(w^T x\right)\right]\theta\left(w^T x\right)x = 0$$

- If $\theta(\boldsymbol{w^T x}) = 1$ → close to the target, $\theta(\boldsymbol{w^T x}) = 0$ → far from the target,
- No matter it's close or far from the target, both steps are small, we just want the bigger step as it far from the target, and smaller step as it close to the target.

# Cross Entropy v.s. Square Error

https://www.youtube.com/watch?v=hSXFuypLukA&t=2009s




http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf