UCB - CS189
Introduction to Machine Learning
Fall 2015
Lecture 6: Logistic regression

Isabelle Guyon

ChaLearn

Come to my office hours…

Wed 2:30-4:30 Soda 329

Last time

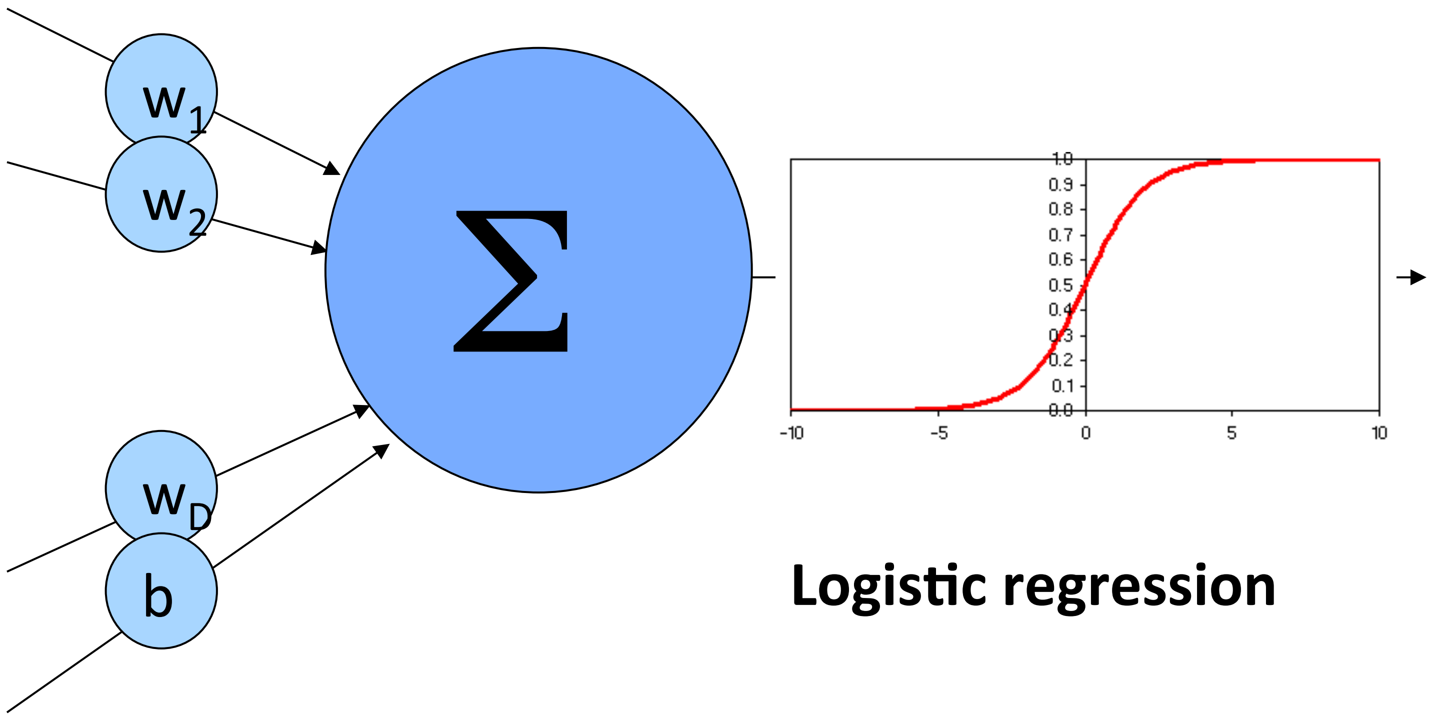
- High complexity models may "**overfit**":
  - Fit perfectly training examples
  - Generalize poorly to new cases
- **SRM solution**: organize the models in nested subsets such that in every structure element

  complexity < θ
- **Regularization:** Formalize learning as a constrained optimization problem, minimize

  regularized risk = training error + λ complexity
- Both formulations are equivalent via the use of Lagrange multipliers.
- θ and λ are hyperparamenters, which can be optimized by cross-validation.

Come to my office hours…
Wed 2:30-4:30 Soda 329

Today

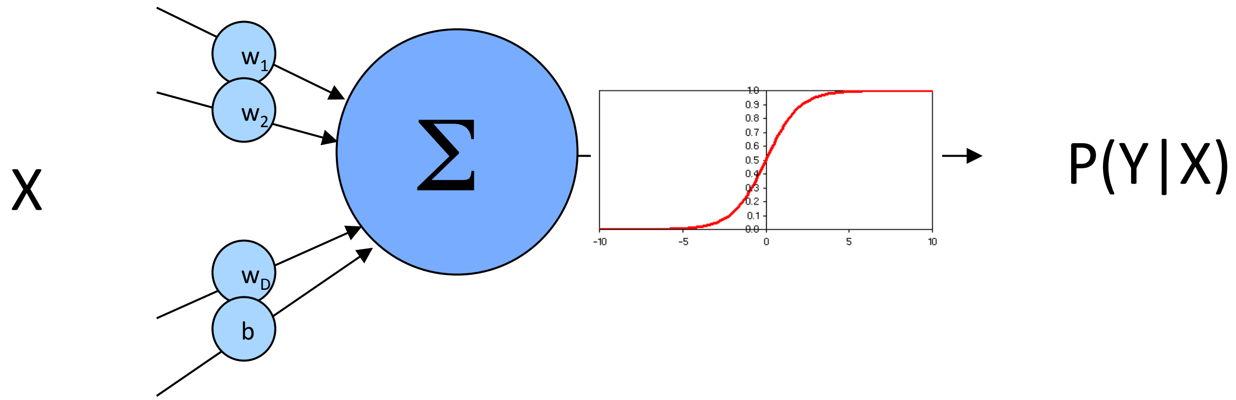


Logistic regression

Math prerequisites

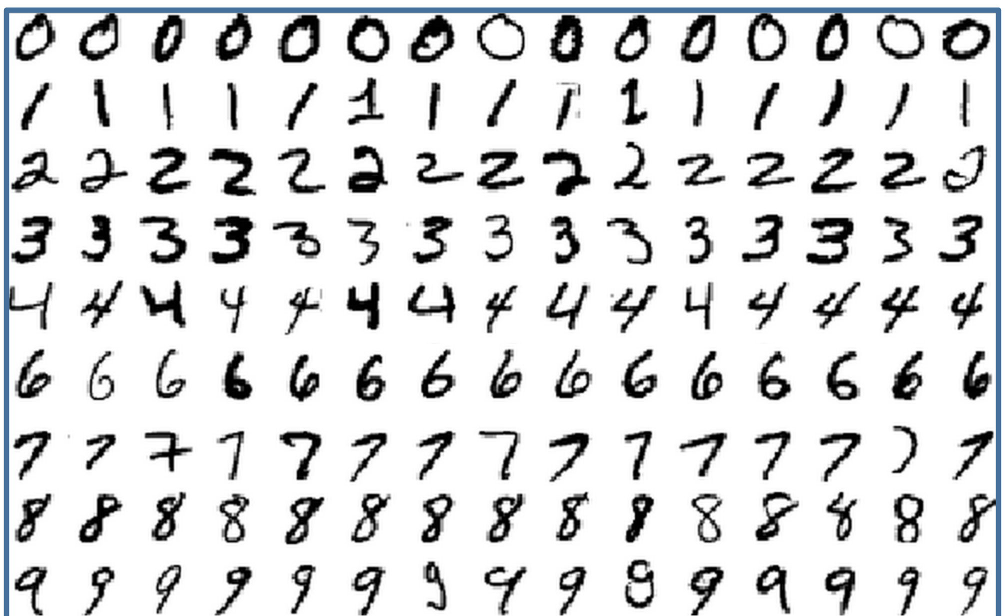
- Random variable, probability distribution
- CDF, PDF
- Normal Law
- Bernouilli trials, Binomial law
- Maximum likelihood
- Bayes rule

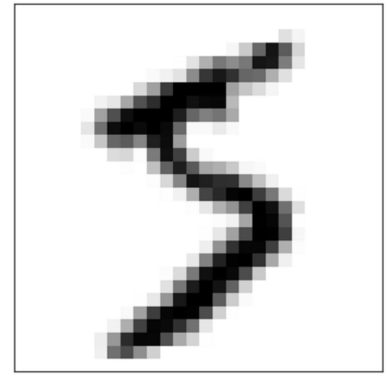
Probabilistic output

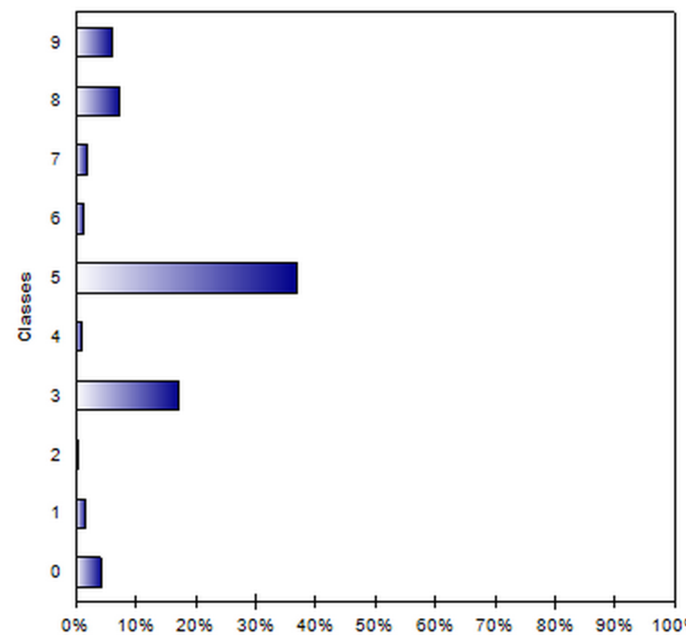


- **Advantages:**
  1. <u>Soft decisions:</u> output more informative than hard decision.
  2. <u>Modular decisions</u>: easier to integrate as part of big decision system.
  3. <u>Flexible decisions:</u> (still) possible to monitor tradeoff between false positive and false negative.
- **Disadvantages:**
  – Need to estimate probabilities (never solve a harder problem than you need).

Benefit 1: Soft decisions







Benefit 2: Modular decisions

- The “HaHa” machine:

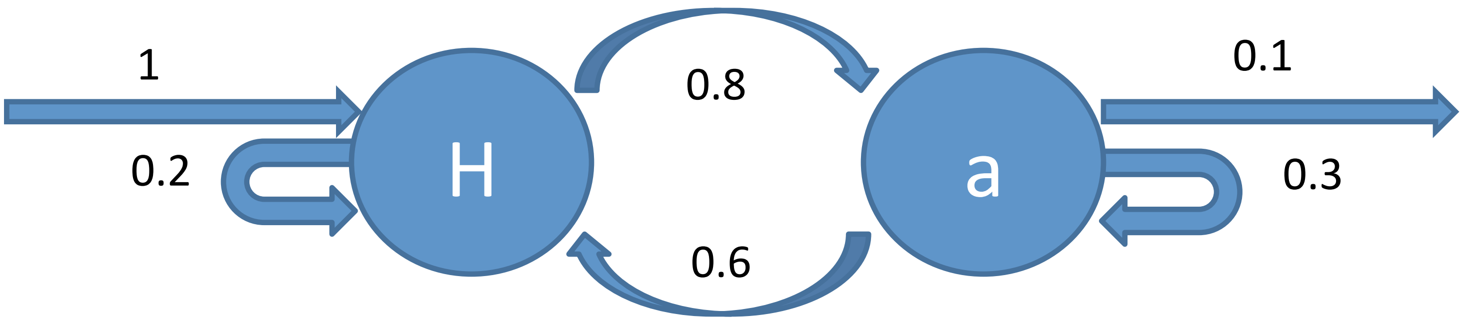
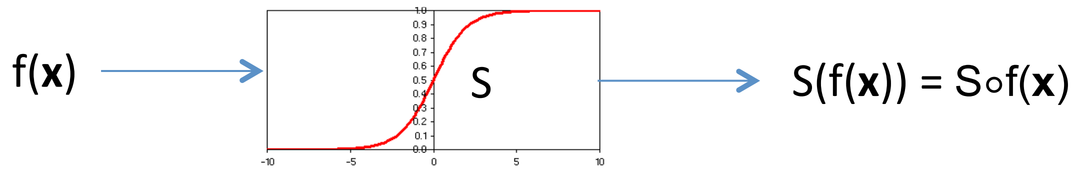


Photo: Bravel’s bucket

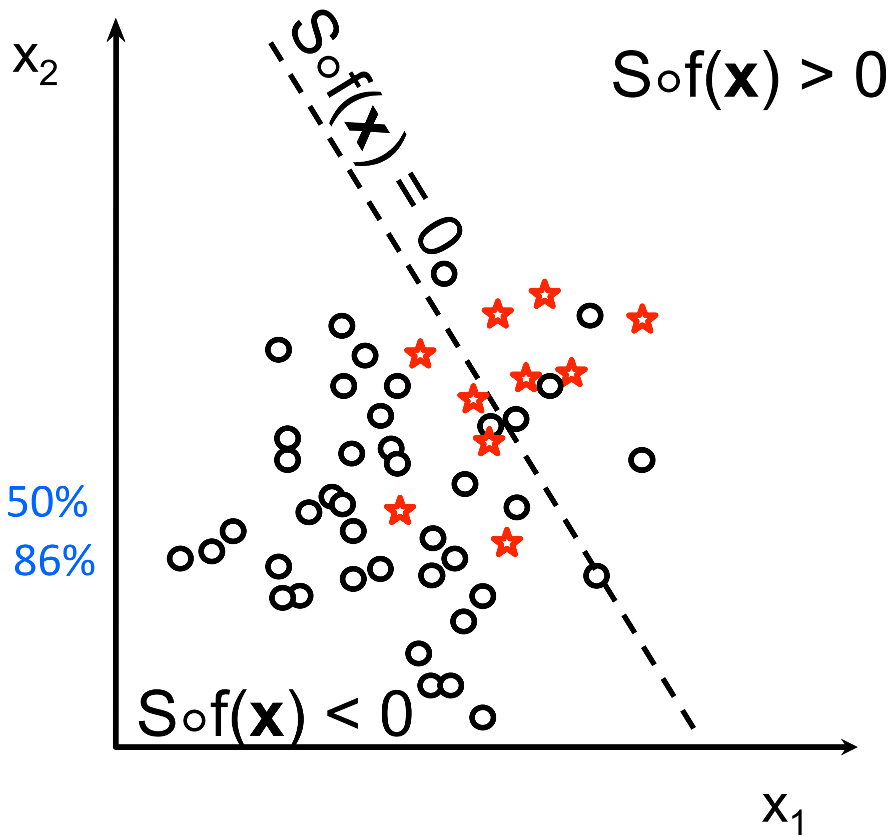
Combine Proba(H | ) coming from the handwriting recognizer with the current state in the “HaHa” machine, which provides a “prior” Proba(H | previous state).

Benefit 3: Flexible decisions



10 errors
5 FP
5 FN

Success rate = 79%
Sensitivity = TP/pos = 50%
Specificity = TN/neg = 86%
BAC = 68%

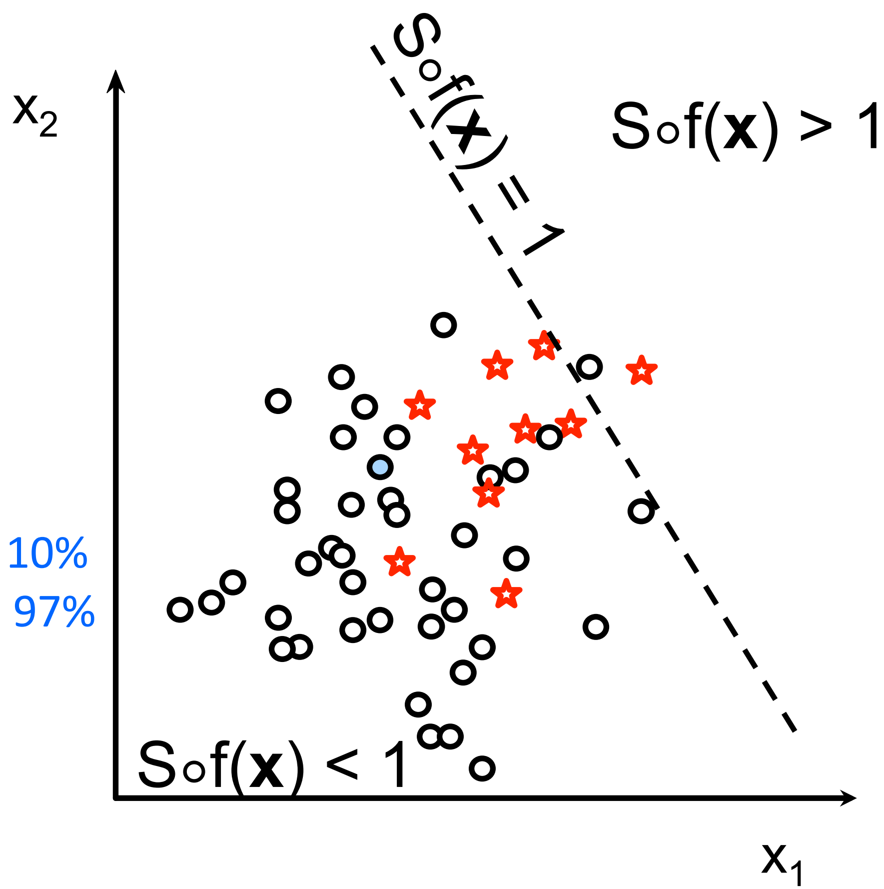


Since S is a smooth monotonically increasing function, varying a threshold on S∘f(**x**) allows us to monitor the fraction of FP and FN, just like for f(**x**).

Benefit 3: Move up the decision boundary → fewer False Positive

10 errors
1 FP
9 FN

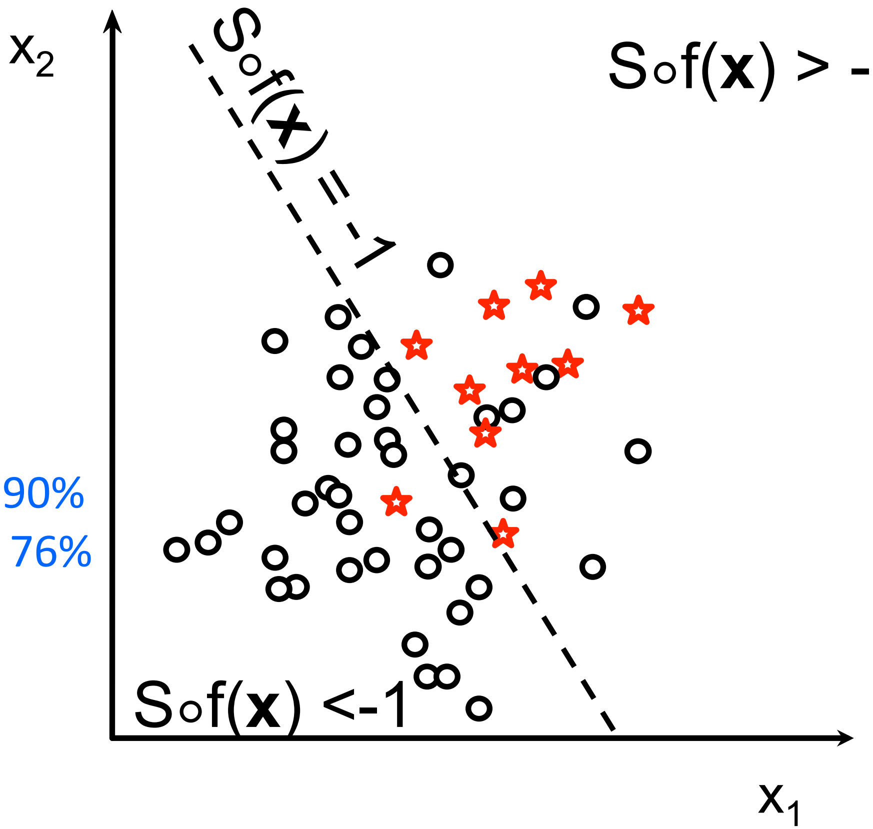
Success rate = 79%
Sensitivity = TP/pos = 10%
Specificity = TN/neg = 97%
BAC = 53%



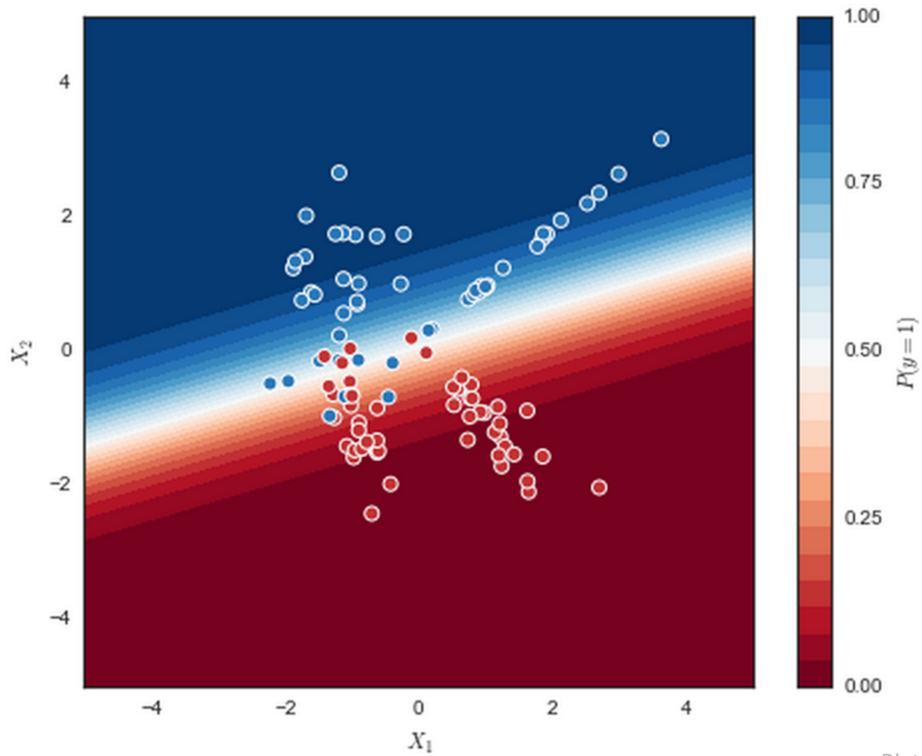
Benefit 3: Move up the decision boundary → fewer False Negative

10 errors
9 FP
1 FN

Success rate = 79%
Sensitivity = TP/pos = 90%
Specificity = TN/neg = 76%
BAC = 83%



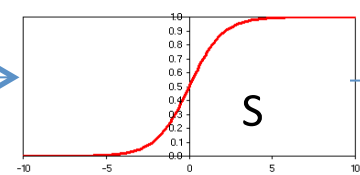
Soft boundary



Picture: Stackoverflow mwaskom

$f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b$

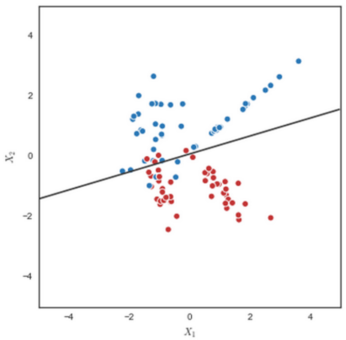
$f(\mathbf{x}) = \Sigma_i\, w_i\, x_i + b$



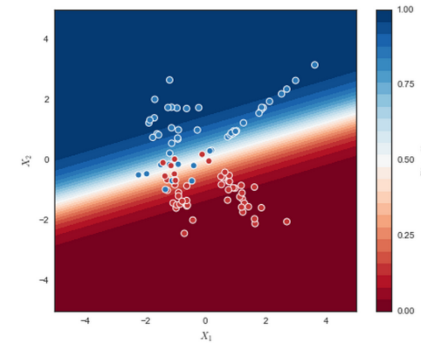
$\underbrace{S(f(\mathbf{x})) = P_f(Y=1 \mid X=\mathbf{x})}_{p}$

$f(\mathbf{x}) = g(p)$

$S^{-1} = g =$ link function

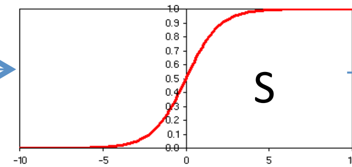


Which function S?



f(**x**) = **w** • **x** + b

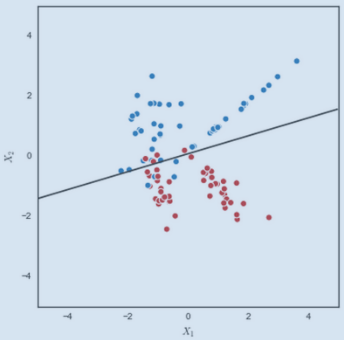
f(**x**) ∈ [-∞, +∞]



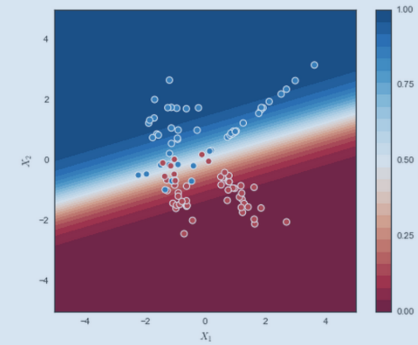
S(f(**x**)) = P(Y=1 | X=**x**)

S(f(**x**)) ∈ [0, 1]

How do we map [-∞, +∞] to [0, 1]?

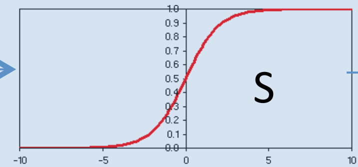


Which function S?



$f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b$

$f(\mathbf{x}) \in [-\infty, +\infty]$



$S(f(\mathbf{x})) = P(Y=1 \mid X=\mathbf{x})$

$S(f(\mathbf{x})) \in [0, 1]$

How do we map [-∞, +∞] to [0, 1]?

$P_f(Y=1 \mid X=\mathbf{x}) = p(\mathbf{x}) \quad \in [0, 1]$

$P_f(Y=-1 \mid X=\mathbf{x}) = 1 - p(\mathbf{x}) \quad \in [0, 1]$

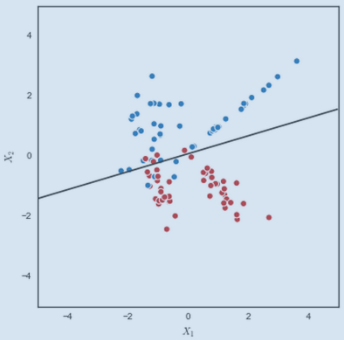
$\log p(\mathbf{x}) \quad \in [-\infty, 0]$

$-\log(1 - p(\mathbf{x})) \in [0, +\infty]$

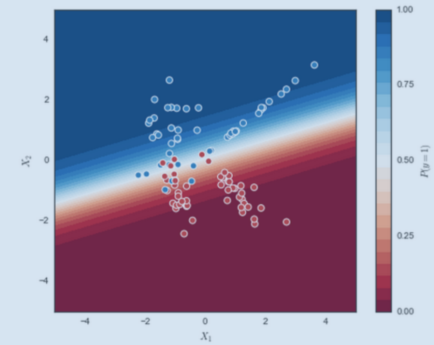
$$f(\mathbf{x}) = \underbrace{\log [\ \underbrace{p(\mathbf{x}) / (1-p(\mathbf{x}))}_{\text{odds ratio}}\ ]}_{g(p)} \in [-\infty, +\infty]$$

$g(p) = \log(p/(1-p))$

logit link function

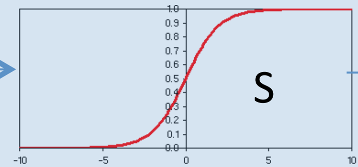


Which function S?



$f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b$

$f(\mathbf{x}) \in [-\infty, +\infty]$



$S(f(\mathbf{x})) = P(Y=1 \mid X=\mathbf{x})$

$S(f(\mathbf{x})) \in [0, 1]$

How do we map $[-\infty, +\infty]$ to $[0, 1]$?

$P_f(Y=1 \mid X=\mathbf{x}) = p(\mathbf{x}) \quad \in [0, 1]$

$P_f(Y=-1 \mid X=\mathbf{x}) = 1 - p(\mathbf{x}) \quad \in [0, 1]$

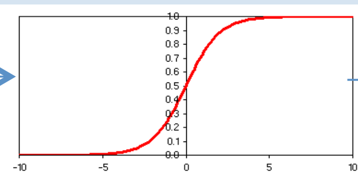
$\log p(\mathbf{x}) \quad \in [-\infty, 0]$

$-\log(1 - p(\mathbf{x})) \in [0, +\infty]$

$$f(\mathbf{x}) = \log [\, \underbrace{p(\mathbf{x}) / (1-p(\mathbf{x}))}_{\text{odds ratio}} \,] \in [-\infty, +\infty]$$

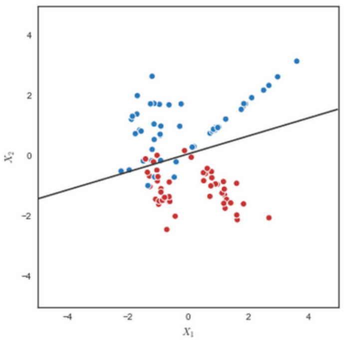
$\boxed{\begin{array}{l} g(p) = \log(p/(1-p)) \\ \text{logit link function} \end{array}}$

$f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b$

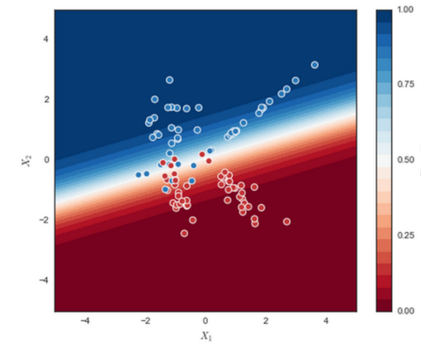


$P_f(Y=1 \mid X=\mathbf{x}) = 1 / (1 + e^{-f(\mathbf{x})})$

$\boxed{S(t) = g^{-1}(t) = 1 / (1 + e^{-t}) \quad \text{logistic function}}$



Assumptions made



- <u>Linear logistic regression</u>:

The log odds ratio (logit) is a linear function of **x**

$\log [P_f(Y=1|X=\mathbf{x}) \,/\, P_f(Y=-1|X=\mathbf{x})] = \mathbf{w} \bullet \mathbf{x} + b$

- <u>Non-linear logistic regression</u>:

The log odds ratio (logit) is any function of **x**

$\log [P_f(Y=1|X=\mathbf{x}) \,/\, P_f(Y=-1|X=\mathbf{x})] = f(\mathbf{x})$

(think of the kernel trick).

Generalized linear models

- Logistic regression belongs to the GLM family:

The GLM consists of three elements:

1. A probability distribution from the exponential family.
2. A linear predictor $\eta = \mathbf{X}\boldsymbol{\beta}$.
3. A link function g such that $\mathrm{E}(\mathbf{Y}) = \boldsymbol{\mu} = g^{-1}(\eta)$.

Notations:

$\beta \leftrightarrow \mathbf{w}$

$\mu \leftrightarrow p$

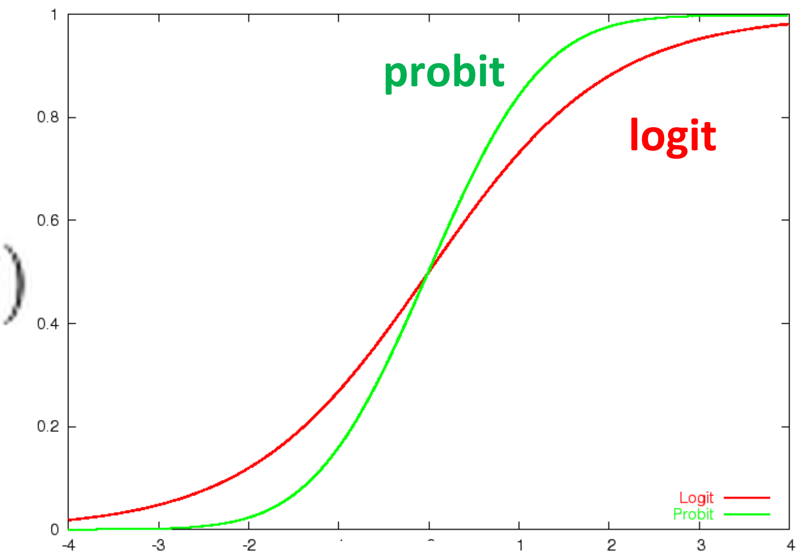
$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = g(p)$

Distribution	Support of distribution	Typical uses	Link name	Link function **g**	Mean function **S=g[-1]**
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$	$\mu = \mathbf{X}\boldsymbol{\beta}$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$	$\mu = -(\mathbf{X}\boldsymbol{\beta})^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-2}$	$\mu = (-\mathbf{X}\boldsymbol{\beta})^{-1/2}$
Poisson	integer: $0, 1, 2, \ldots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$	$\mu = \exp(\mathbf{X}\boldsymbol{\beta})$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$ **logit**	$\mu = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$ **logistic**
Binomial	integer: $0, 1, \ldots, N$	count of # of "yes" occurrences out of N yes/no occurrences			
Categorical	integer: $[0, K)$ K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1	outcome of single K-way occurrence			
Multinomial	K-vector of integer: $[0, N]$	count of occurrences of different types (1 .. K) out of N total K-way occurrences			

Source: Wikipedia

Other choices of link function

$p = S\ (X'\beta)$



$X'\beta = g(p)$

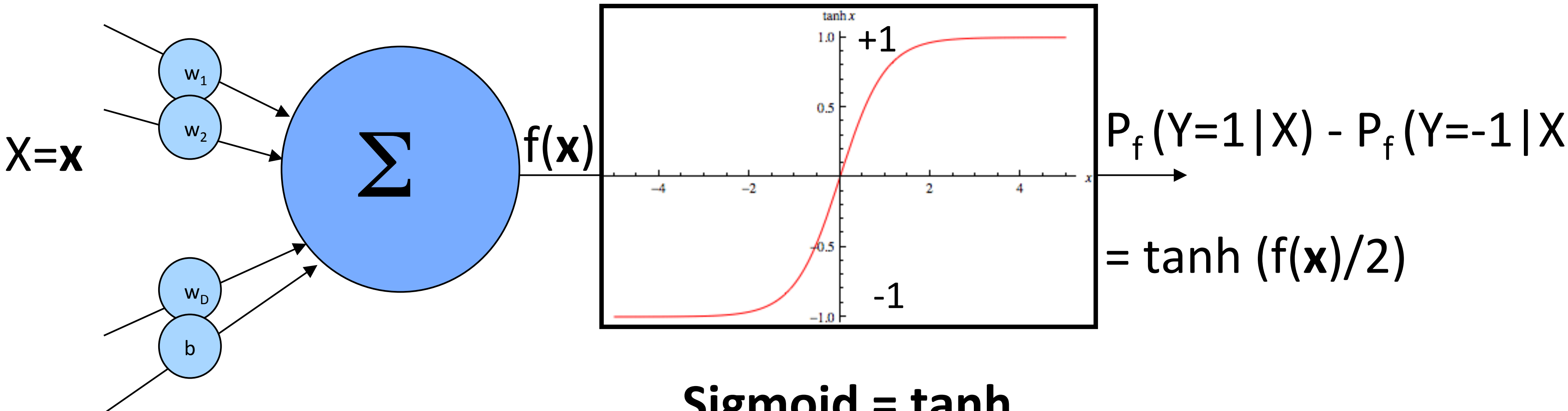
Probit: g^{-1} =

S is the Cumulative Distribution Function (CDF) of the standard normal distribution.

Reminder: $\beta \leftrightarrow w$

Source: Wikipedia

Similitude with the sigmoid neuron



Sigmoid = tanh

Relationship with Bayes optimal discriminant classifier

- Imagine we knew the true probability distribution $P(X, Y)$, then…

decide: $y=+1$, if $P(Y=1|X=\mathbf{x}) > P(Y=-1|X=\mathbf{x})$

$y=-1$, otherwise

would give the smallest error rate (irreducible error).

- $P(Y=1|X=\mathbf{x}) = P(Y=-1|X=\mathbf{x})$ is the Bayes optimal decision boundary.
- Valid Bayes optimal discriminant:
  - $f^*(\mathbf{x}) = P(Y=1|X=\mathbf{x}) - P(Y=-1|X=\mathbf{x})$ sigmoid neuron
  - $f^*(\mathbf{x}) = \log P(Y=1|X=\mathbf{x}) - \log P(Y=-1|X=\mathbf{x})$ logit

How to define a risk functional?

- $f^*(\mathbf{x}) = \log P(Y=1|X=\mathbf{x}) - \log P(Y=-1|X=\mathbf{x})$ Bayes optimal
- $f(\mathbf{x}, \mathbf{w}) = \mathbf{w} \bullet \mathbf{x} + b$ Proposed estimator (learning machine)
- How do we get the target values? All we have are data samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_N, y_N)$.
- Idea: compare the predicted distribution $P_f(Y=y^k|X=\mathbf{x}^k)$ with the empirical distribution.
- Distance between distributions: Use KL divergence or the related "cross-entropy":

$$\boxed{R(f) = (1/N) \Sigma_{\mathbf{k=1:N}} - \log P_f(Y=y^k|X=\mathbf{x}^k)}$$

Link to Maximum Likelihood (ML)

People who know about ML will recognize that min(cross-entropy) $\Leftrightarrow$ max(likelihood):

- Cross-entropy:

  $R(f) = (1/N) \sum_{\mathbf{k=1:N}} - \log P_f\,(Y=y^k|X=\mathbf{x}^k)$

- Likelihood:

  $P(\text{data}|f) = \prod_{\mathbf{k=1:N}} P_f\,(Y=y^k|X=\mathbf{x}^k)$

- Cross-entropy = negative log likelihood.

Logistic loss = cross-entropy loss

$P(Y=1 \mid X=\mathbf{x}) = 1 / (1+ e^{-f(\mathbf{x})})$

$P(Y=-1 \mid X=\mathbf{x}) = 1 / (1+ e^{f(\mathbf{x})})$

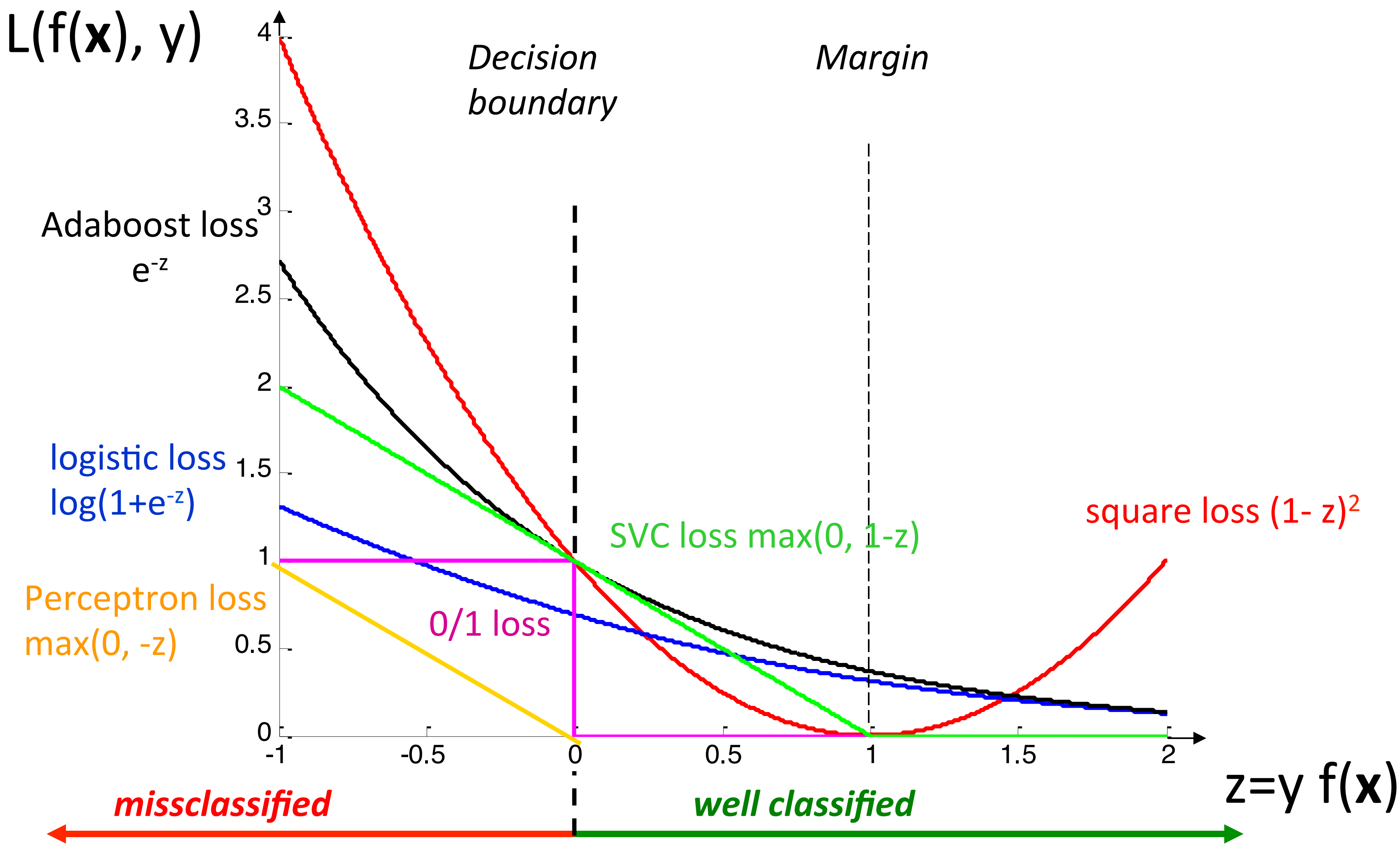
$z = y\, f(\mathbf{x})$ functional margin $(y = \pm 1)$

$P(Y=y \mid X=\mathbf{x}) = 1 / (1+ e^{-z})$

$$\boxed{- \log P(Y=y \mid X=\mathbf{x}) = \log (1+ e^{-z})}$$

Loss Functions

The risk is the average of the loss.



Dual learning machines

PARAMETRIC

$f(\mathbf{x}) = \mathbf{w} \bullet \Phi(\mathbf{x})$

$\mathbf{w} = \sum_k \alpha_k \Phi(\mathbf{x}^k)$

Perceptron algorithm

$\mathbf{w} \leftarrow \mathbf{w} + y_k \Phi(\mathbf{x}^k)$ if $y_k f(\mathbf{x}^k)<0$

(Rosenblatt 1958)

Minover (optimum margin)

$\mathbf{w} \leftarrow \mathbf{w} + y_k \Phi(\mathbf{x}^k)$ for min $y^k f(\mathbf{x}^k)$

(Krauth-Mézard 1987)

LMS regression

$\mathbf{w} \leftarrow \mathbf{w} + \eta \, (y_k - f(\mathbf{x}_k)) \, \Phi(\mathbf{x}^k)$

(Widrow-Hoff 1960)

NON PARAMETRIC

$f(\mathbf{x}) = \sum_k \alpha_k \, k(\mathbf{x}^k, x)$

$k(\mathbf{x}^k, \mathbf{x}) = \Phi(\mathbf{x}^k).\Phi(\mathbf{x})$

Potential Function algorithm

$\alpha_k \leftarrow \alpha_k + y_k$ if $y_k f(\mathbf{x}^k)<0$

(Aizerman et al 1964)

Dual minover

$\alpha_k \leftarrow \alpha_k + y^k$ for min $y_k f(\mathbf{x}^k)$

(ancestor of SVM 1992,
similar to kernel Adatron, 1998,
and SMO, 1999)

Dual LMS

$\alpha_i \leftarrow \alpha_i + \eta \, (y_i - f(\mathbf{x}^k))$

Exercise: Gradient Descent

- Linear discriminant $f(\mathbf{x}) = \Sigma_i\, w_i\, x_i$
- Functional margin $z = y\, f(\mathbf{x})$, $y=\pm 1$
- Compute $\partial z/\partial w_i$
- Derive the learning rules $\Delta w_i = -\eta\, \partial L/\partial w_i$ corresponding to the following loss functions:

square loss
$L=(1- z)^2$

SVC loss
$L=\max(0, 1-z)$

Adaboost loss
$L=e^{-z}$

Perceptron loss
$L=\max(0, -z)$

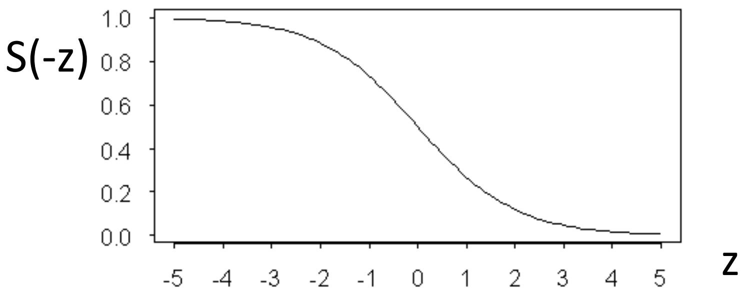
logistic loss
$L=\log(1+e^{-z})$

Logistic regression

Cox, 1958

- $f(\mathbf{x}) = \Sigma_i \, w_i \, x_i$
- $\boxed{z = y \, f(\mathbf{x})} = \Sigma_i \, w_i \, y \, x_i$
- $\partial z/\partial w_i = y \, x_i$ Hebb's rule update
- $L_{logistic} = \log(1+e^{-z})$
- $L/\partial z = - \, e^{-z} / (1+e^{-z}) = - \, S(-z)$ Loss variation
- $\Delta w_i = -\eta \, \partial L/\partial w_i = -\eta \, \partial L/\partial z \, . \, \partial z/\partial w_i$

$$\boxed{\Delta w_i = \eta \, S(-z) \, y \, x_i}$$



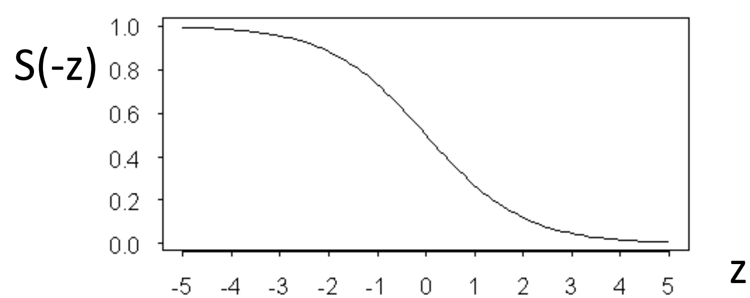
Like Hebb's rule but weighted: misclassified examples count more.

Logistic regression in $\Phi-$space

Cox, 1958

- $f(\mathbf{x}) = \Sigma_i \, w_i \, \Phi_i \, (\mathbf{x})$
- $\boxed{z = y \, f(\mathbf{x})} = \Sigma_i \, w_i \, y \, \Phi_i \, (\mathbf{x})$
- $\partial z / \partial w_i = y \, \Phi_i \, (\mathbf{x})$
- $L_{logistic} = \log(1+e^{-z})$
- $L/\partial z = - e^{-z} / (1+e^{-z}) = - S(-z)$
- $\Delta w_i = -\eta \, \partial L/\partial w_i = -\eta \, \partial L/\partial z \, . \, \partial z/\partial w_i$

$$\boxed{\Delta w_i = \eta \, S(-z) \, y \, \Phi_i \, (\mathbf{x})}$$



Like Hebb's rule but weighted: misclassified examples count more.

Kernel "Trick"

- $f(\mathbf{x}) = \sum_k \alpha_k \, k(\mathbf{x}^k, \mathbf{x})$
- $k(\mathbf{x}^k, \mathbf{x}) = \mathbf{\Phi}(\mathbf{x}^k) \bullet \mathbf{\Phi}(\mathbf{x})$

NON PARAMETRIC

$\Updownarrow$ Dual forms

- $f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{\Phi}(\mathbf{x})$
- $\mathbf{w} = \sum_k \alpha_k \, \mathbf{\Phi}(\mathbf{x}^k)$

PARAMETRIC

Kernel logistic regression

- $f(\mathbf{x}) = \Sigma_i \, w_i \, \Phi_i(\mathbf{x}) = \Sigma_h \, \alpha_h \, k(\mathbf{x}^h, \mathbf{x})$
- Φ−space version: $\Delta w_i = \eta \, S(-z) \, y \, \Phi_i(\mathbf{x})$
- For example $(\mathbf{x}^k, y^k)$:

$$\Delta \mathbf{w} = \eta \, S(-z) \, y^k \, \Phi(\mathbf{x}^k)$$

- $\mathbf{w} = \Sigma_k \, \alpha_k \, \Phi(\mathbf{x}^k)$, $\Delta \mathbf{w} = \Delta\alpha_k \, \Phi(\mathbf{x}^k)$

$$\boxed{\Delta\alpha_k = \eta \, S(-z) \, y^k}$$

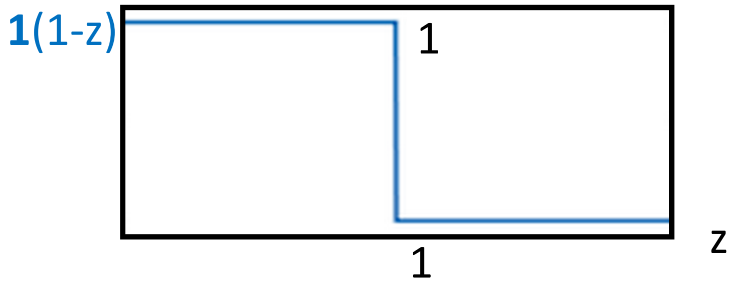
Note: the $\Delta\alpha$ is different when $\partial L / \partial \alpha$ is computed directly.

Comparison with hinge loss

- $f(\mathbf{x}) = \Sigma_i\, w_i\, \Phi_i(\mathbf{x}) = \Sigma_h\, \alpha_h\, k(\mathbf{x}^h, \mathbf{x})$
- $z = y\, f(\mathbf{x})$
- Maximum margin (hinge loss max(0, 1-z)):

$\Delta w_i = \eta\, \mathbf{1}(1\text{-}z)\; y^k\; \Phi_i(\mathbf{x}^k)$

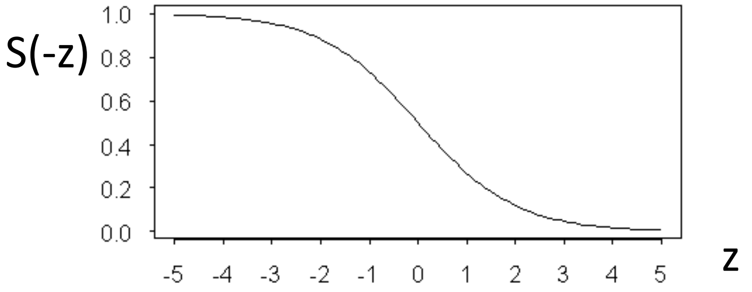
$\Delta \alpha_k = \eta\, \mathbf{1}(1\text{-}z)\; y^k$



- Logistic regression (logistic loss $\log(1+e^{-z})$):

$\Delta w_i = \eta\, S(-z)\; y^k\; \Phi_i(\mathbf{x}^k)$

$\Delta \alpha_k = \eta\, S(-z)\; y^k$



Adding shrinkage

- $f(\mathbf{x}) = \Sigma_i\, w_i\, \Phi_i(\mathbf{x}) = \Sigma_h\, \alpha_h\, k(\mathbf{x}^h, \mathbf{x})$
- We present example k for learning.
- Logistic regression without shrinkage:

  $\Delta w_i = \eta\, S(-z)\, y^k\, \Phi_i(\mathbf{x}^k)$

  $\Delta \alpha_k = \eta\, S(-z)\, y^k$
- Logistic regression with shrinkage:

  $\Delta w_i = -\gamma\, w_i + \eta\, S(-z)\, \Phi_i(\mathbf{x}^k)$

  $w_i = \Sigma_k\, \alpha_k\, \Phi_i(\mathbf{x}^k)$

  $\Delta \alpha_k = -\gamma\, \alpha_k + \eta\, S(-z)\, y^k$ for example k

  $\Delta \alpha_h = -\gamma\, \alpha_h$ for the other examples

Summary

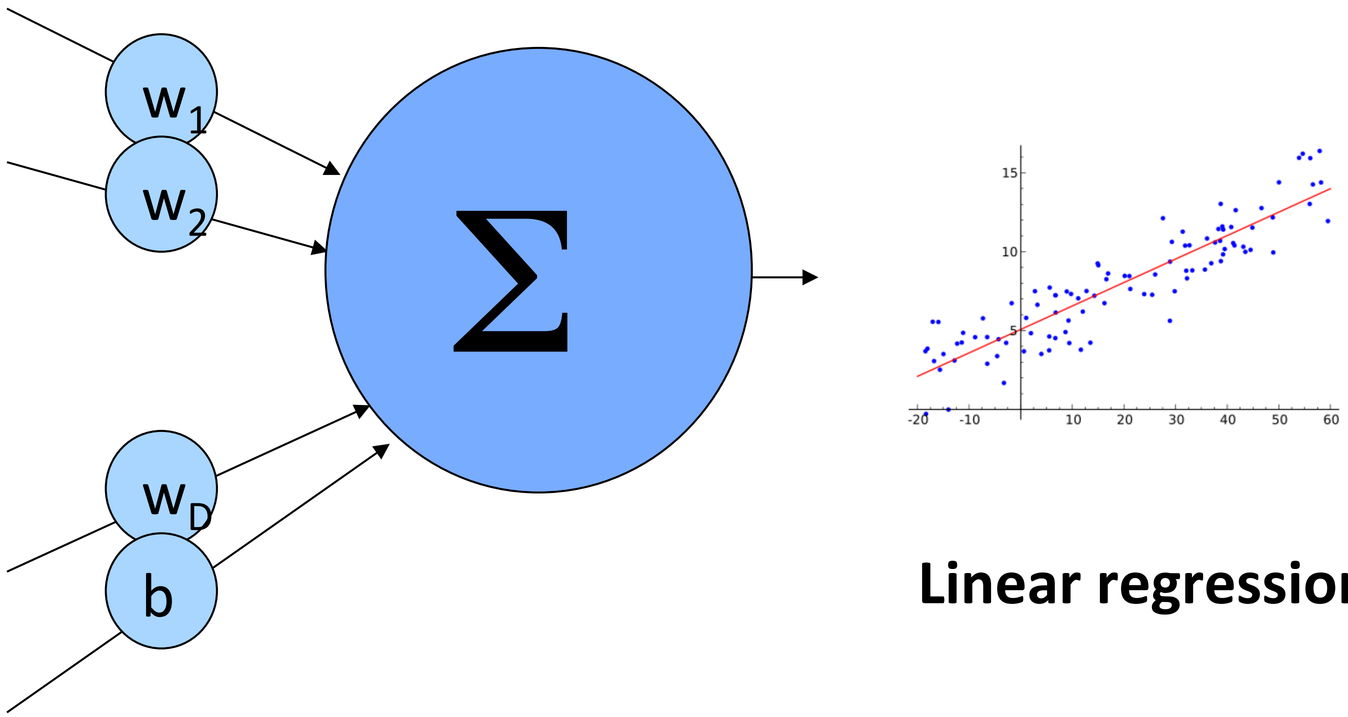
- To map a discriminant function $f(\mathbf{x})$ to probabilities $p = P(Y=1|X=\mathbf{x})$ use a link function:

  $$f(\mathbf{x}) = g(p)$$

- Using the logit link $g(p) = p/(1-p)$ leads to logistic regression; $S(z) = g^{-1}(z) = 1/(1+e^{-z})$ is the logistic (or sigmoid) function.
- $P(Y=\mathbf{y}|X=\mathbf{x}) = S(z)$ with $z = y\, f(\mathbf{x})$.
- The logistic loss is $-\log S(z) = \log(1+ e^{-z})$.
- The Hebb's rule update is multiplied by $S(-z)$.

Come to my office hours…
Wed 2:30-4:30 Soda 329

Next time



Linear regression