上海科技大学
ShanghaiTech University

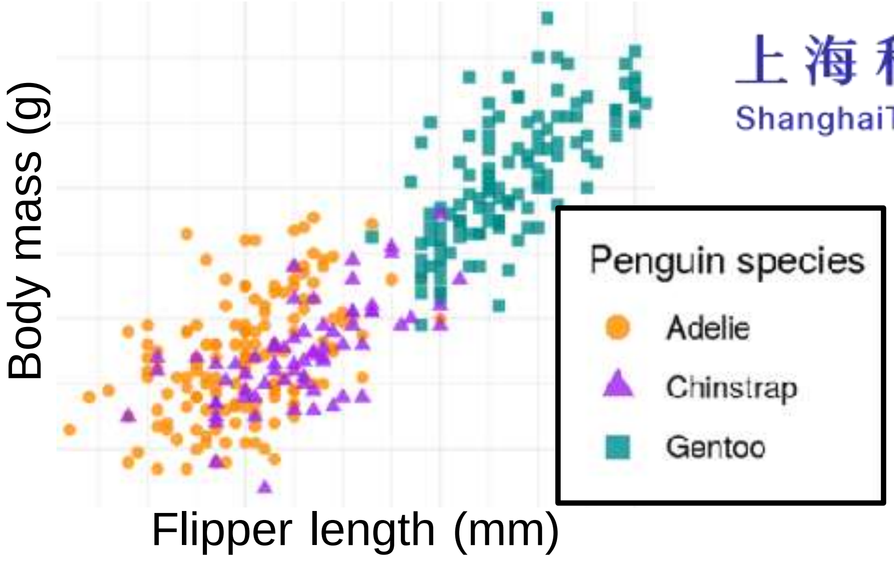
CS182: Introduction to Machine Learning – Logistic Regression

Yujiao Shi
SIST, ShanghaiTech
Spring, 2025

Slides Courtesy of MIT, CMU

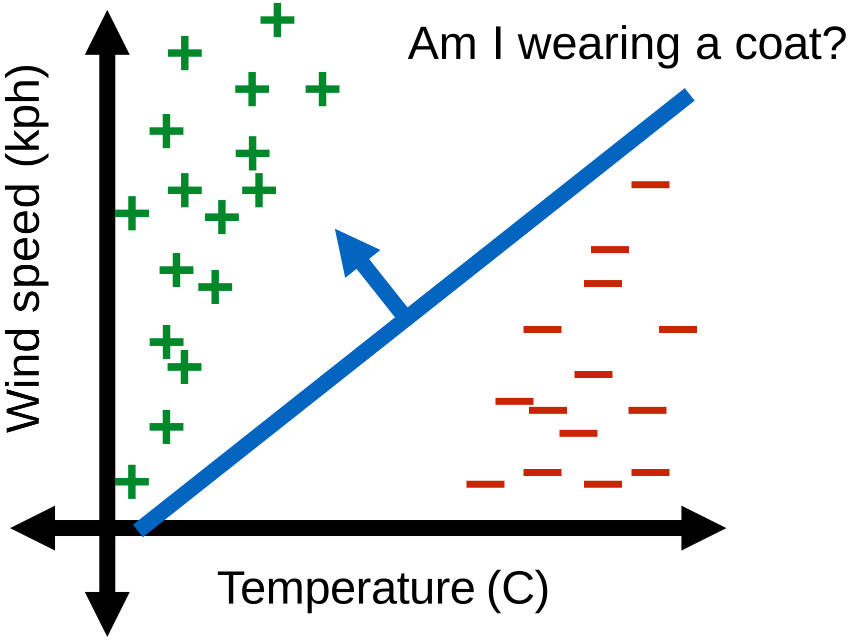
Recall

- Perceptron struggles with data that's not linearly separable



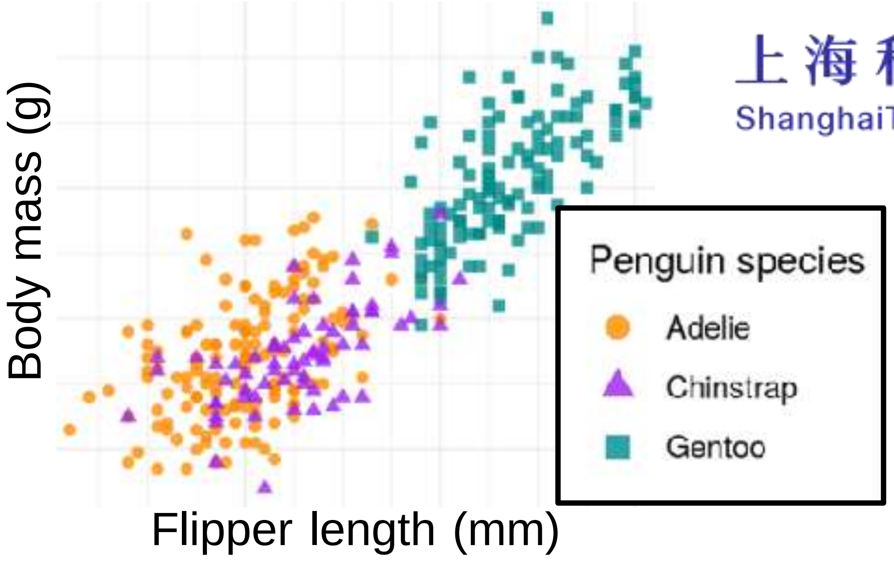
Notice

- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)



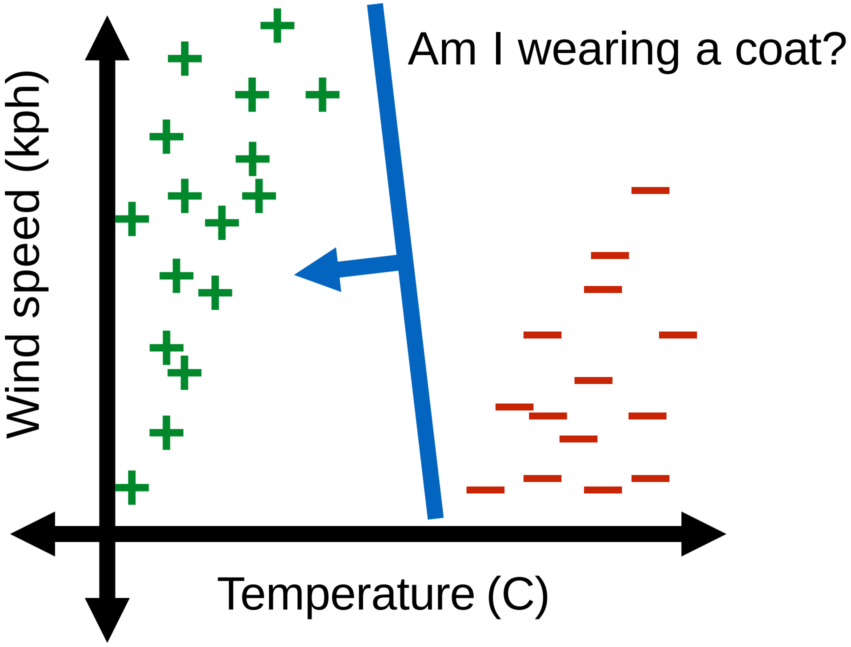
Recall

- Perceptron struggles with data that's not linearly separable



Notice

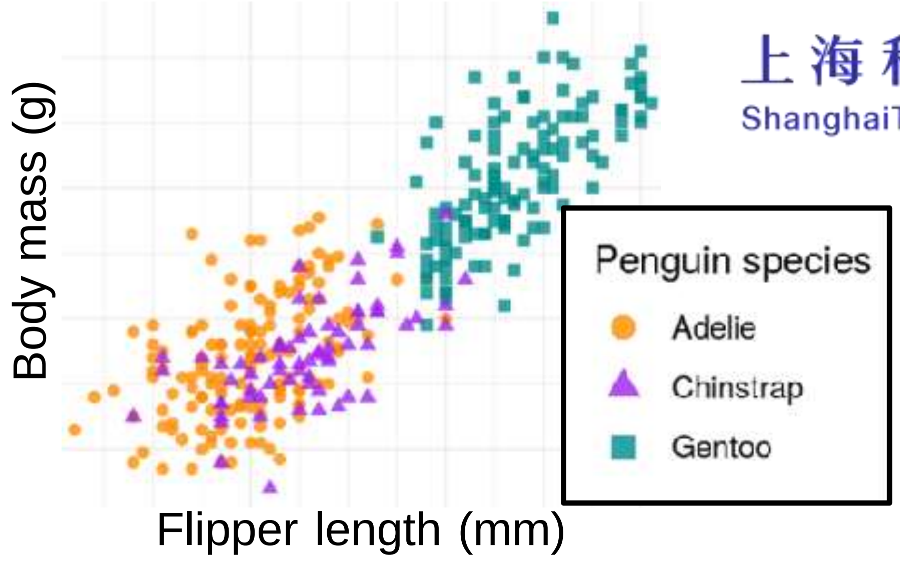
- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)



上海科技大学 ShanghaiTech University

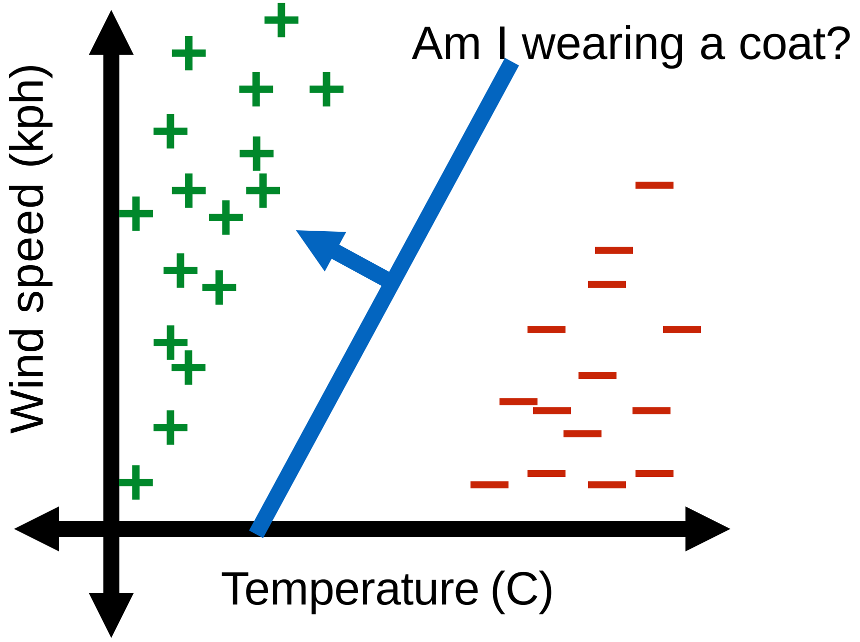
Recall

- Perceptron struggles with data that's not linearly separable



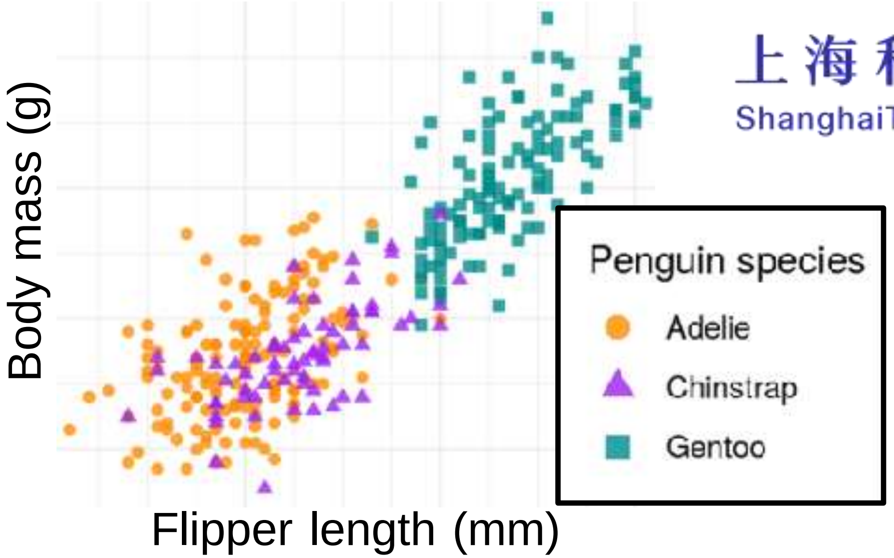
Notice

- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)



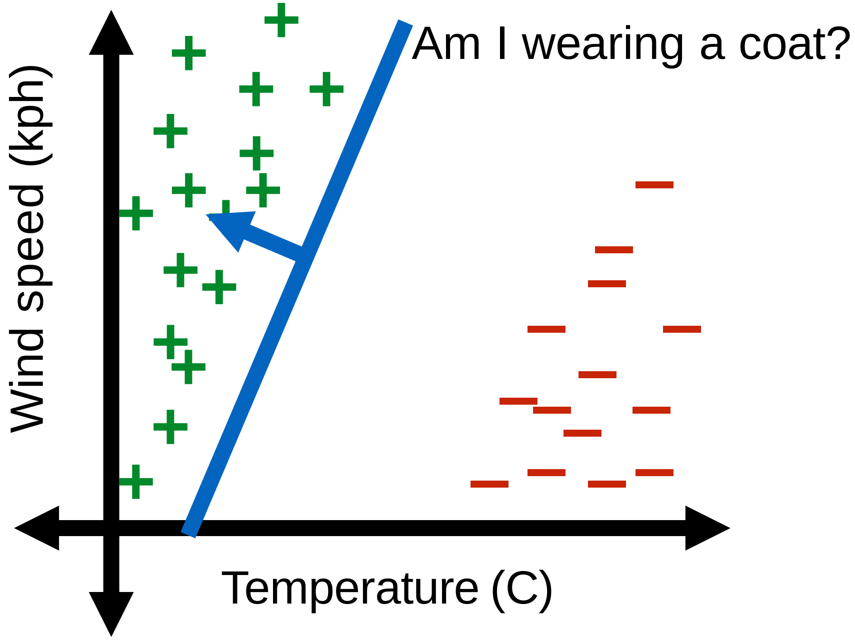
Recall

- Perceptron struggles with data that's not linearly separable



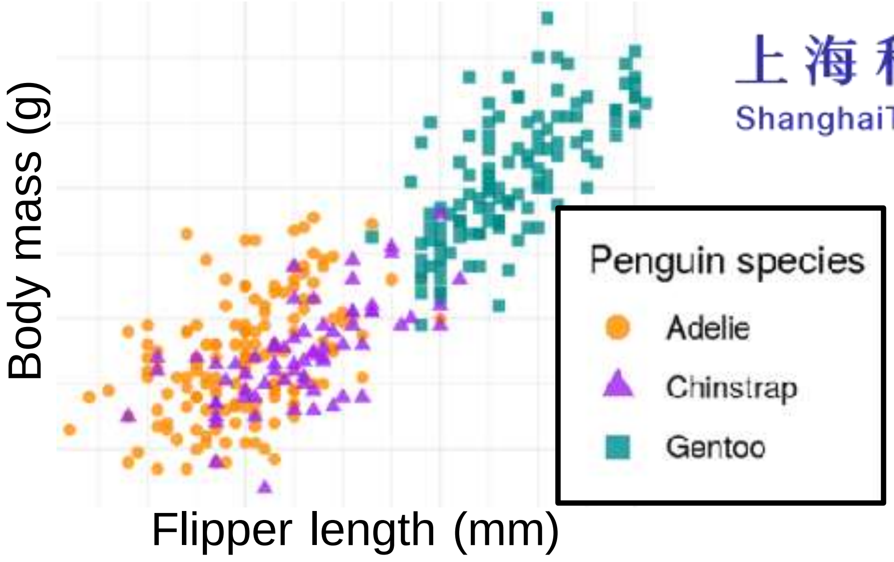
Notice

- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)



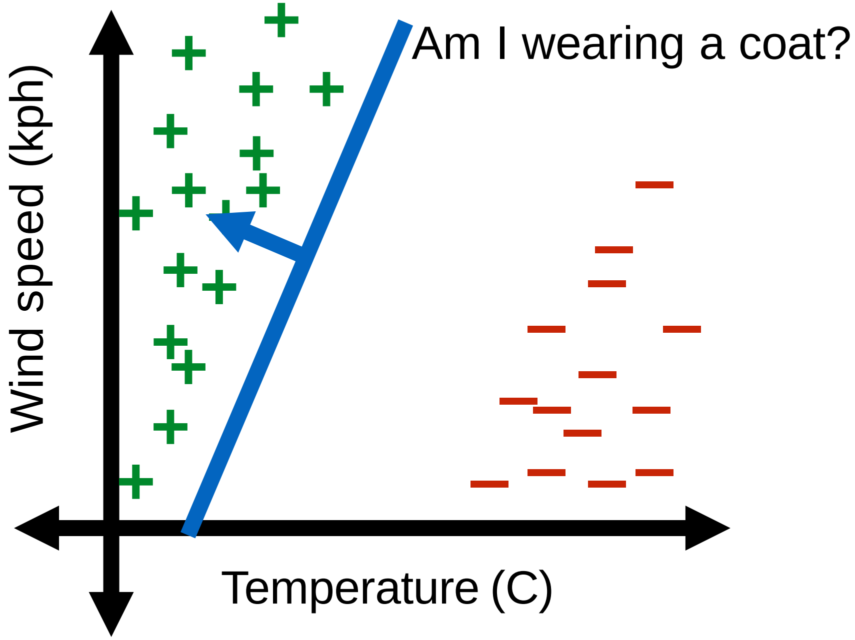
Recall

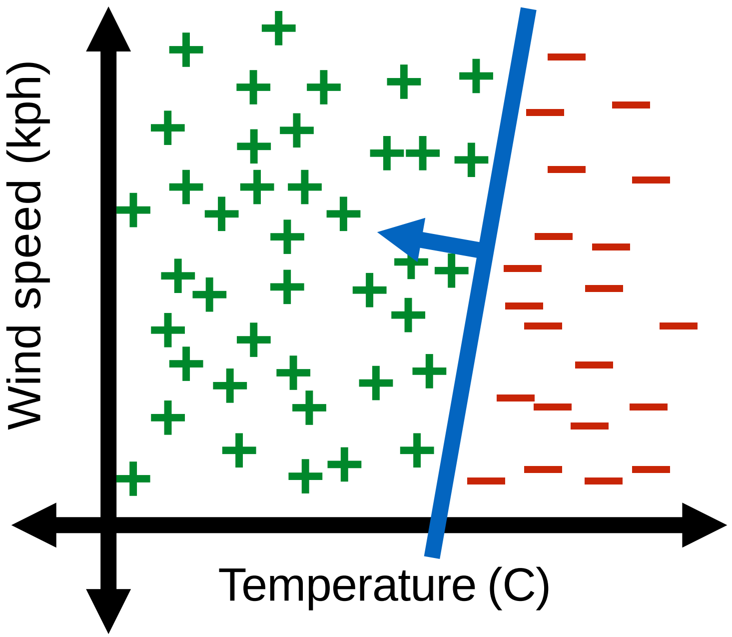
- Perceptron struggles with data that's not linearly separable



Notice

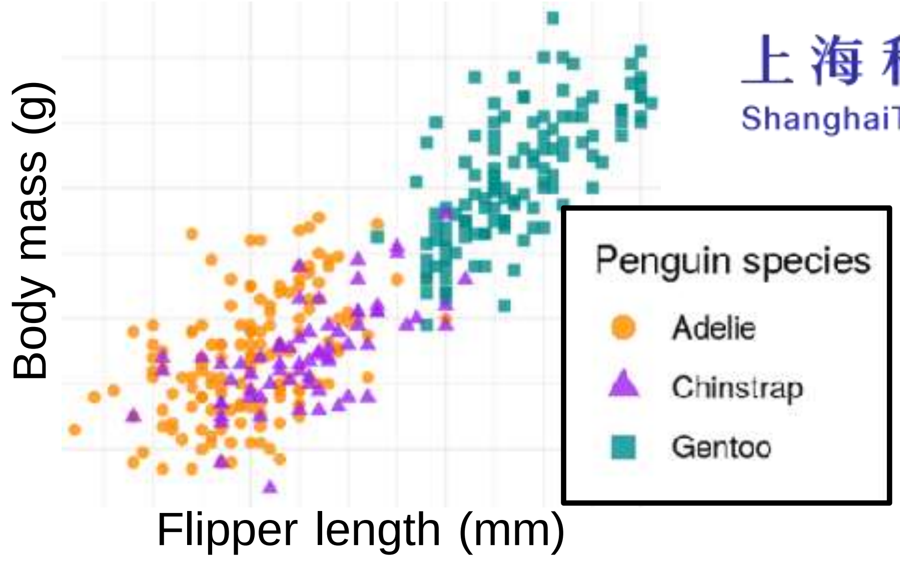
- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)





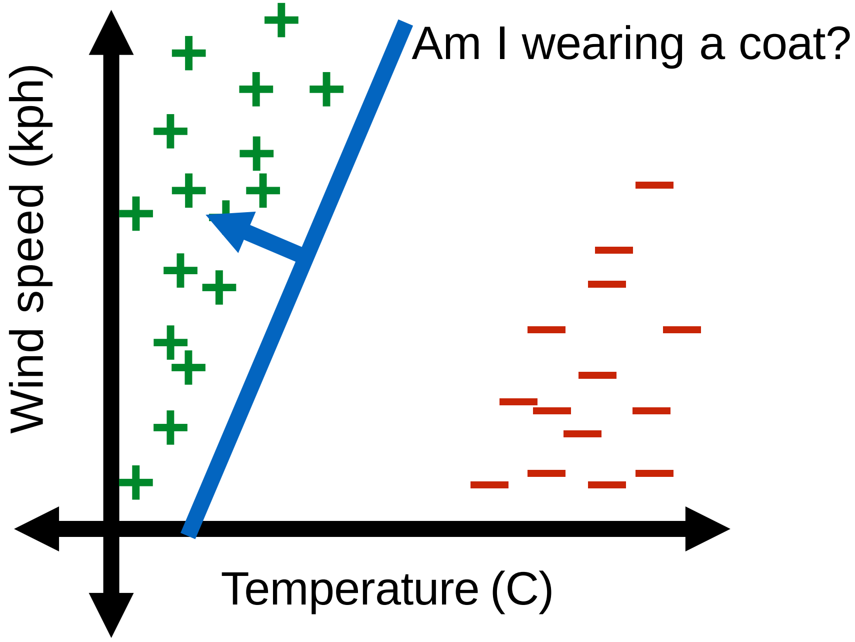
Recall

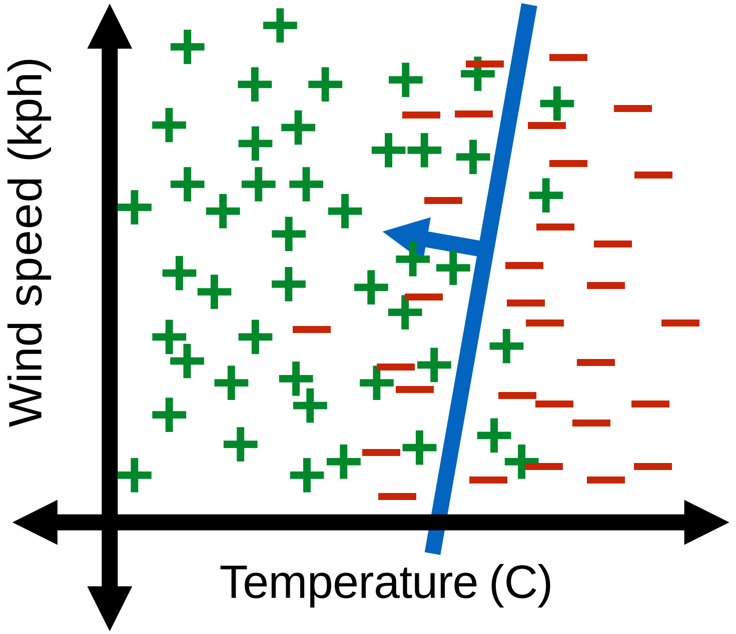
- Perceptron struggles with data that's not linearly separable



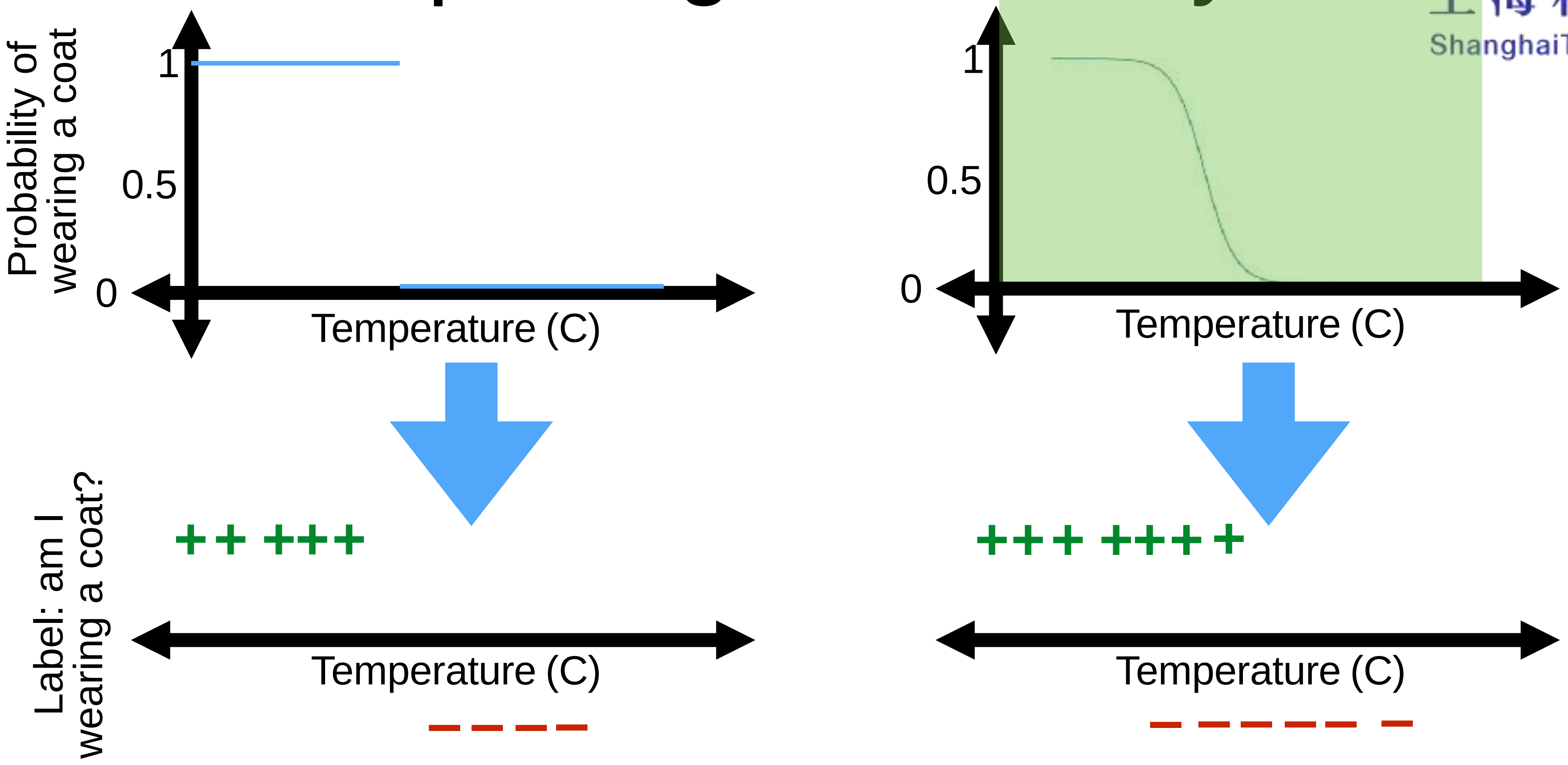
Notice

- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)



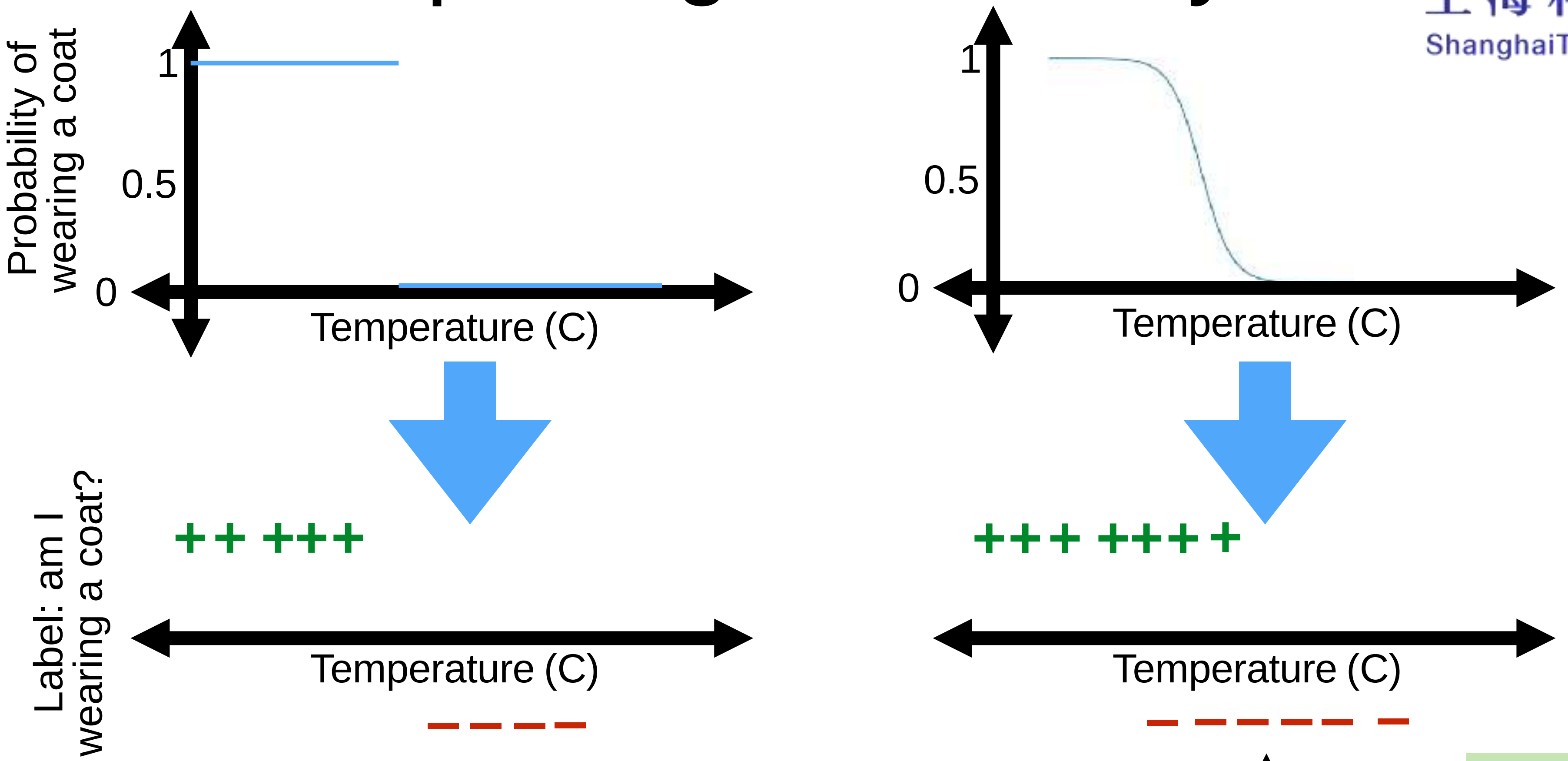


Capturing uncertainty



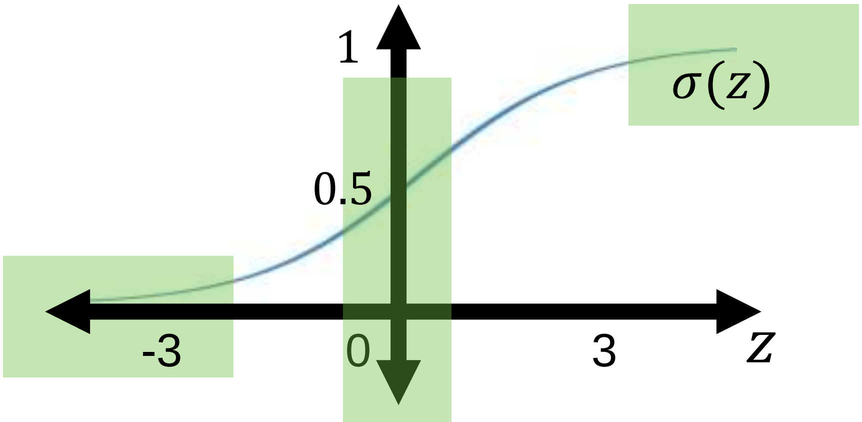
- How to make this shape?

Capturing uncertainty

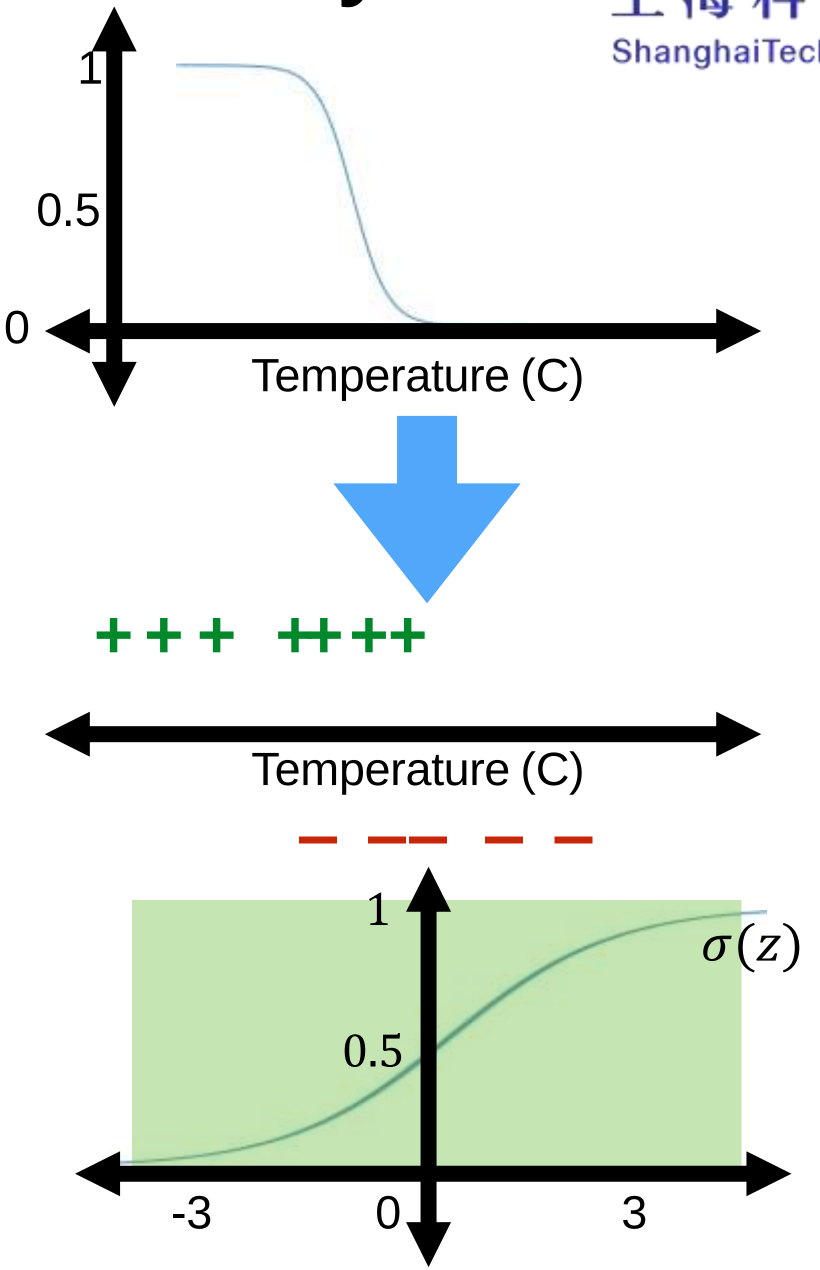


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



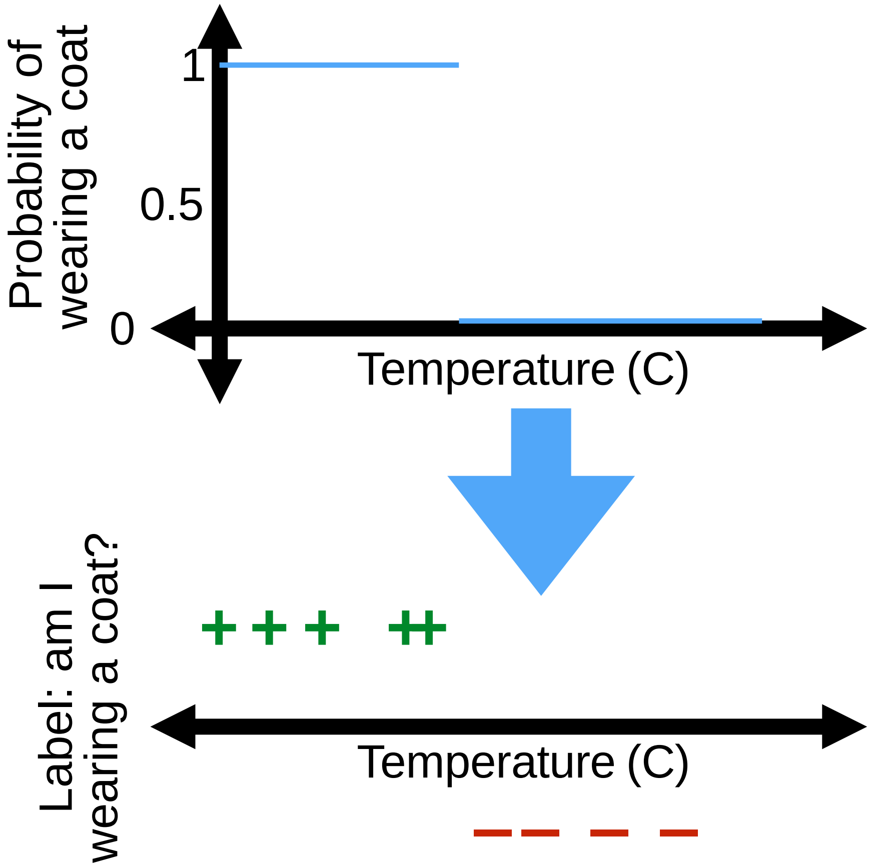
Capturing uncertainty

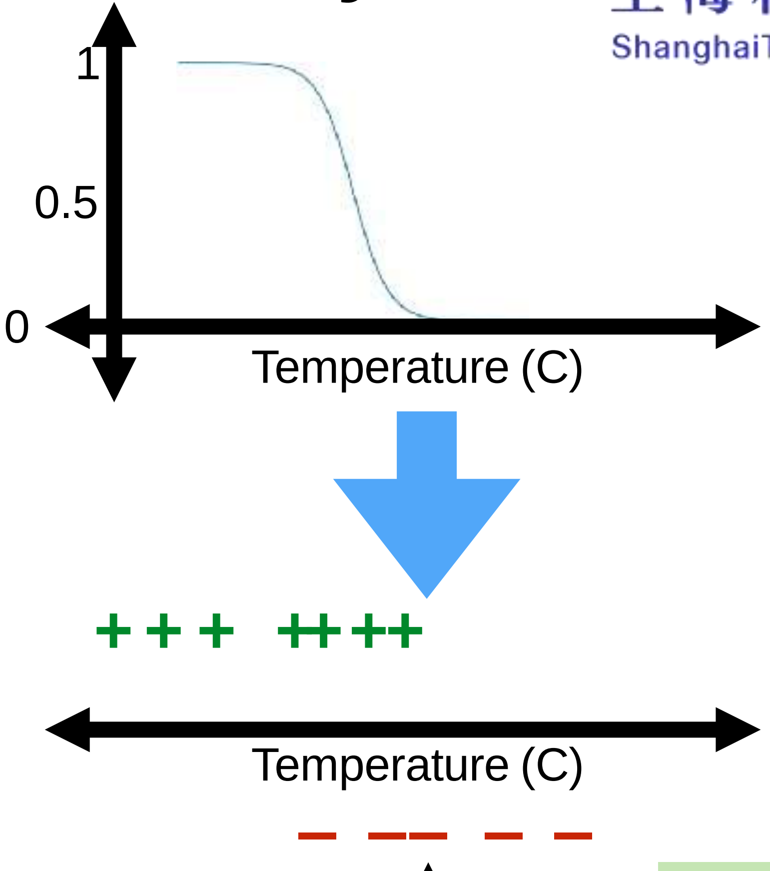


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

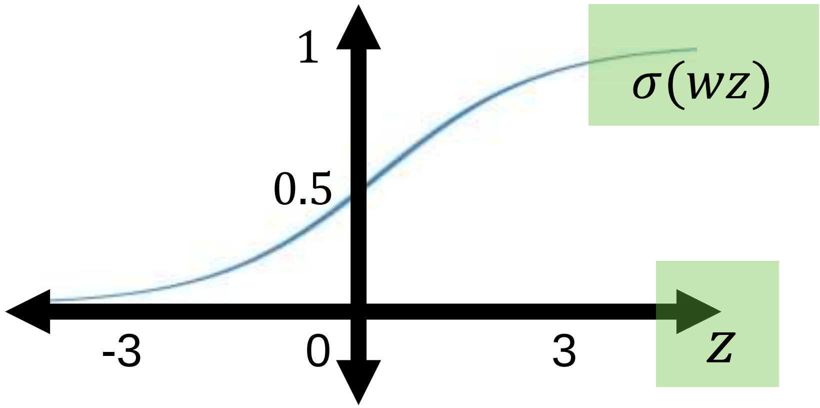
Capturing uncertainty



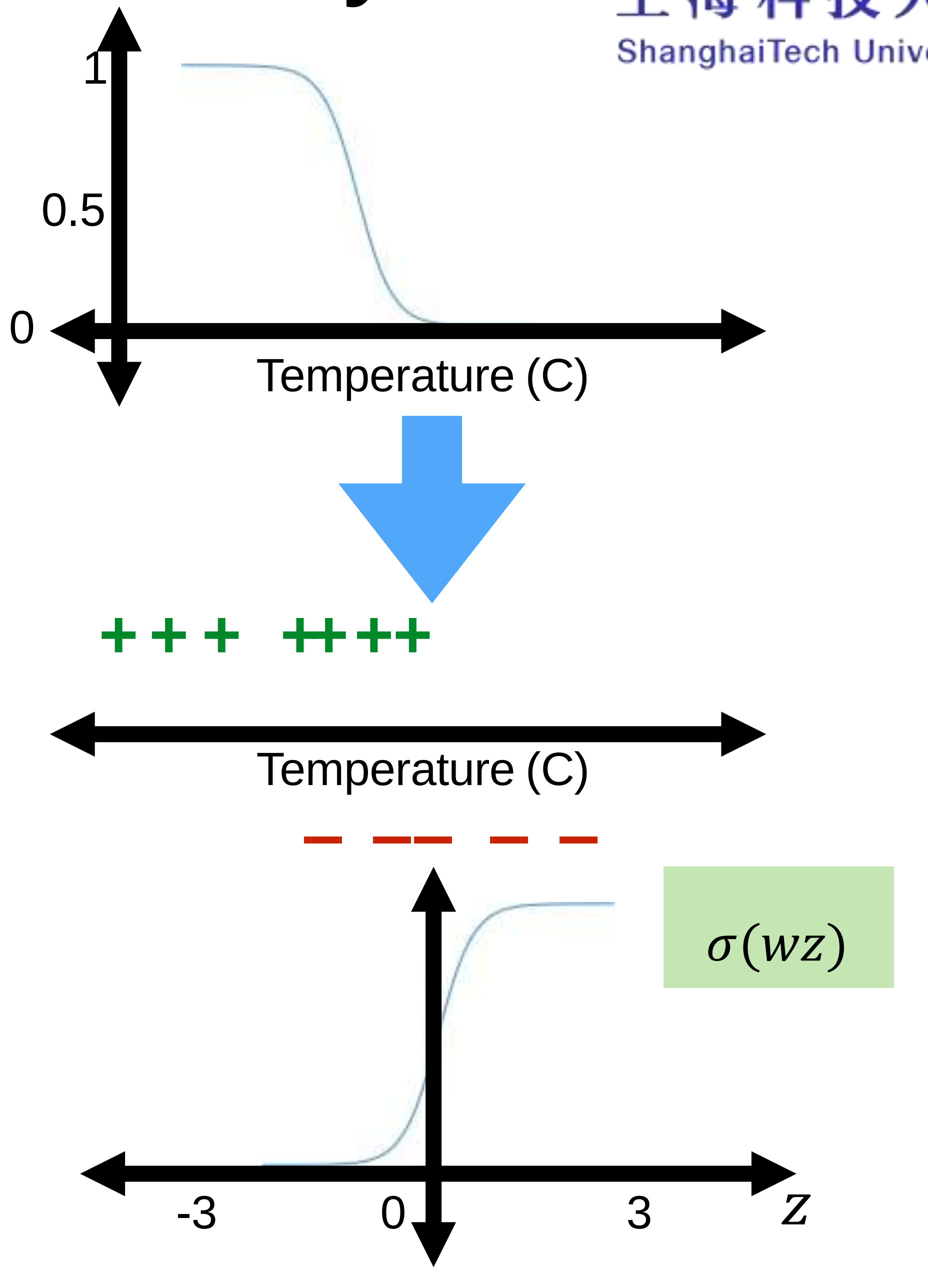


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



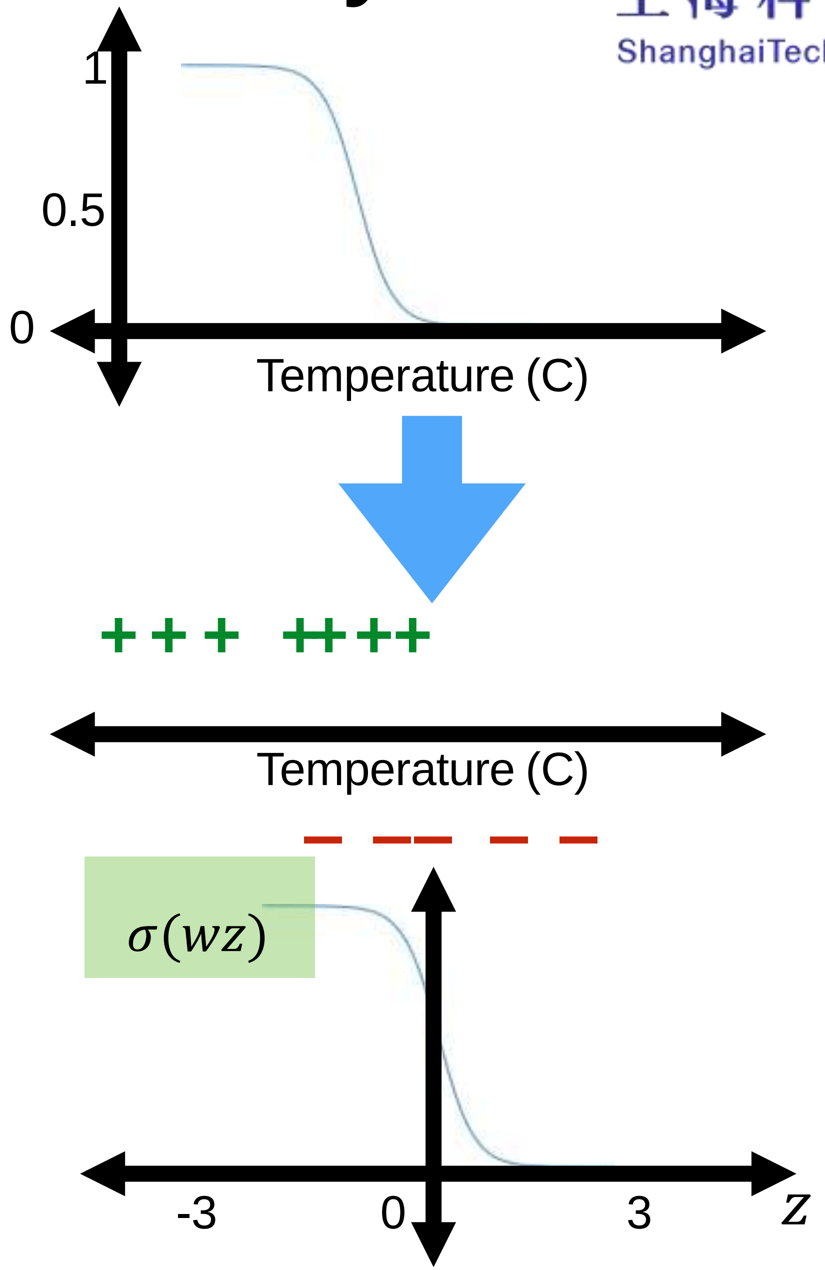
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

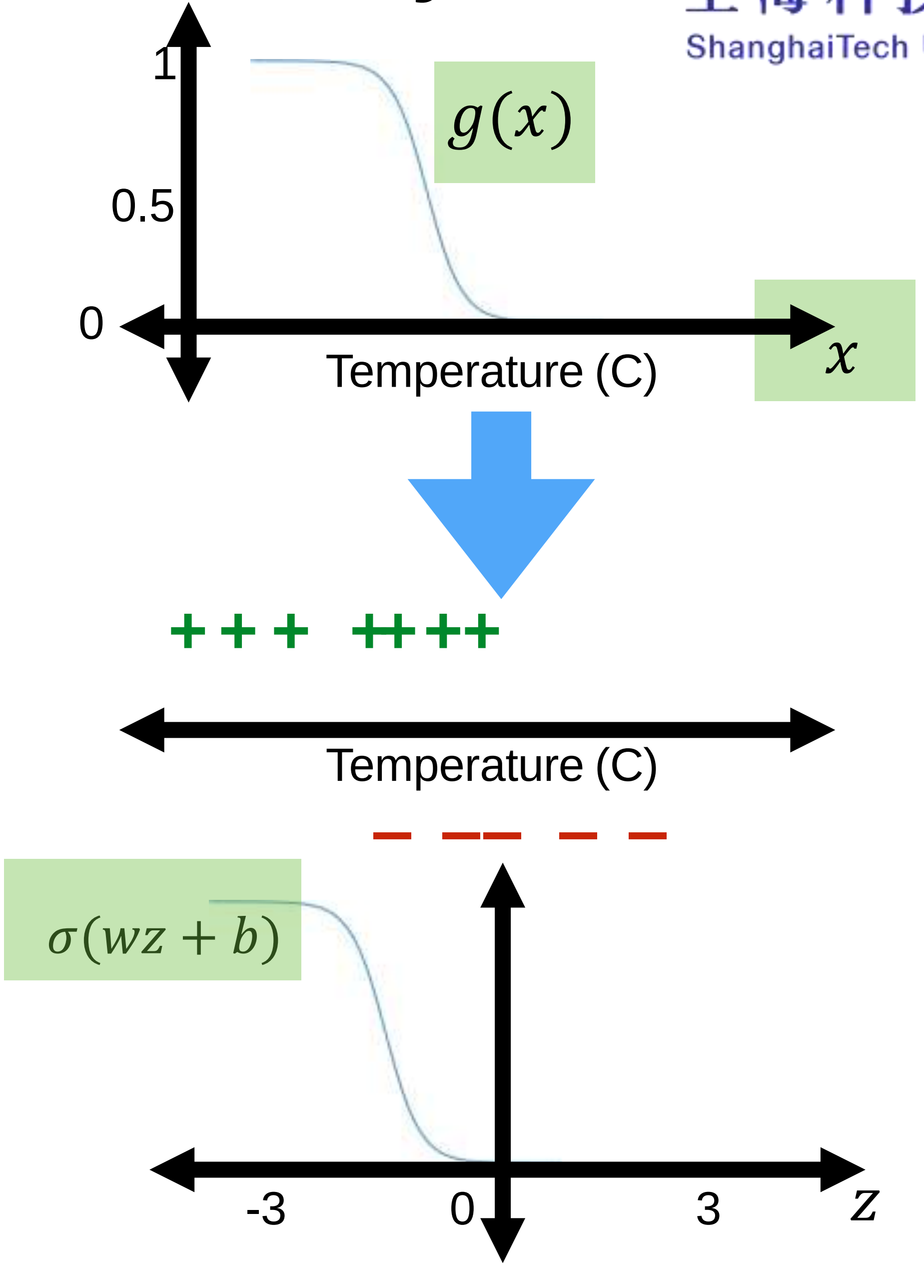


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

$$g(x) = \sigma(wx + b)$$
$$= \frac{1}{1 + \exp\{-(wx + b)\}}$$



- How to make this shape?
  - Sigmoid/logistic function

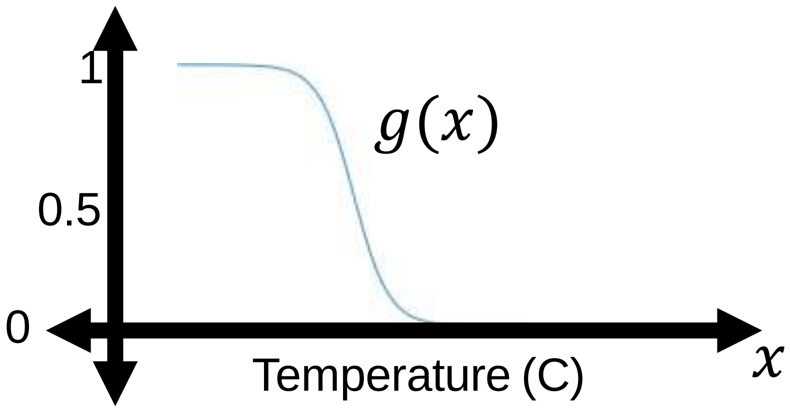
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

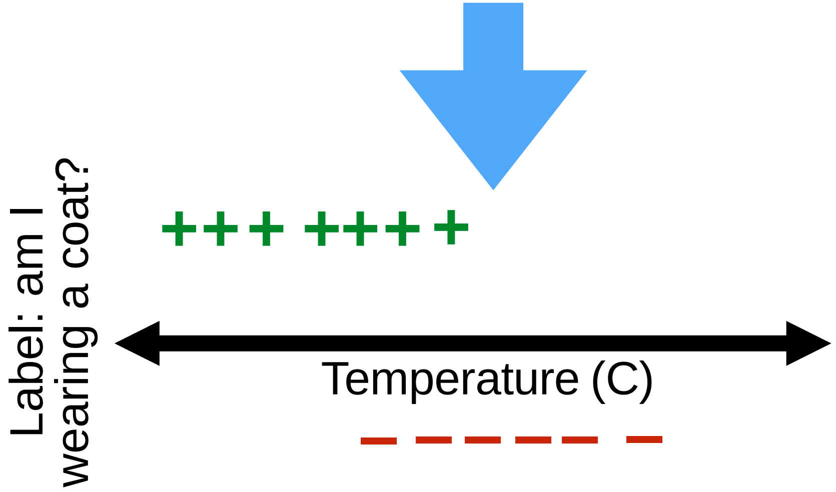
Capturing uncertainty

1 feature:

$$g(x) = \sigma(wx + b)$$

$$= \frac{1}{1 + \exp\{-(wx + b)\}}$$

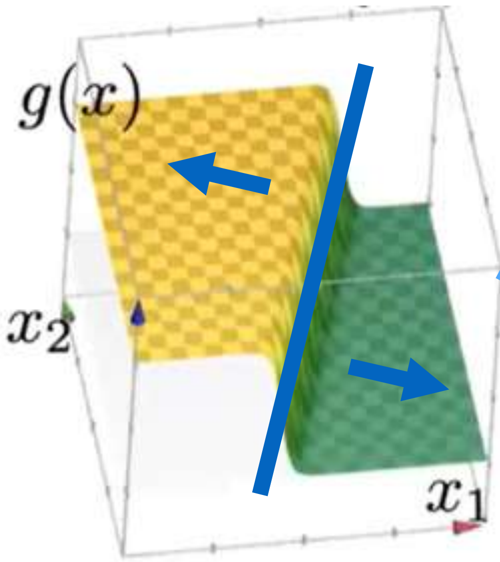


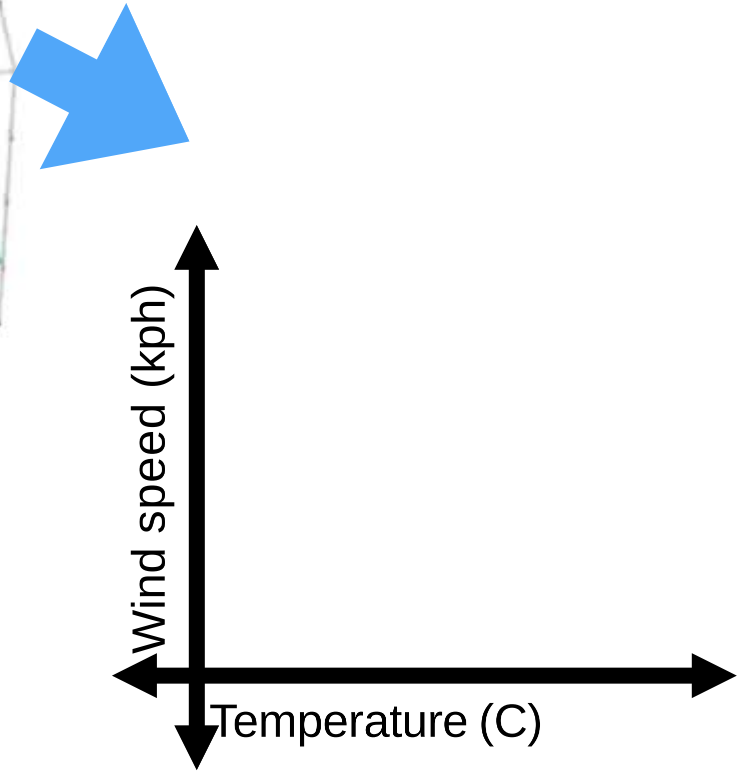


2 features:

$$g(x) = \sigma(w^T x + b)$$

$$= \frac{1}{1 + \exp\{-(w^T x + b)\}}$$

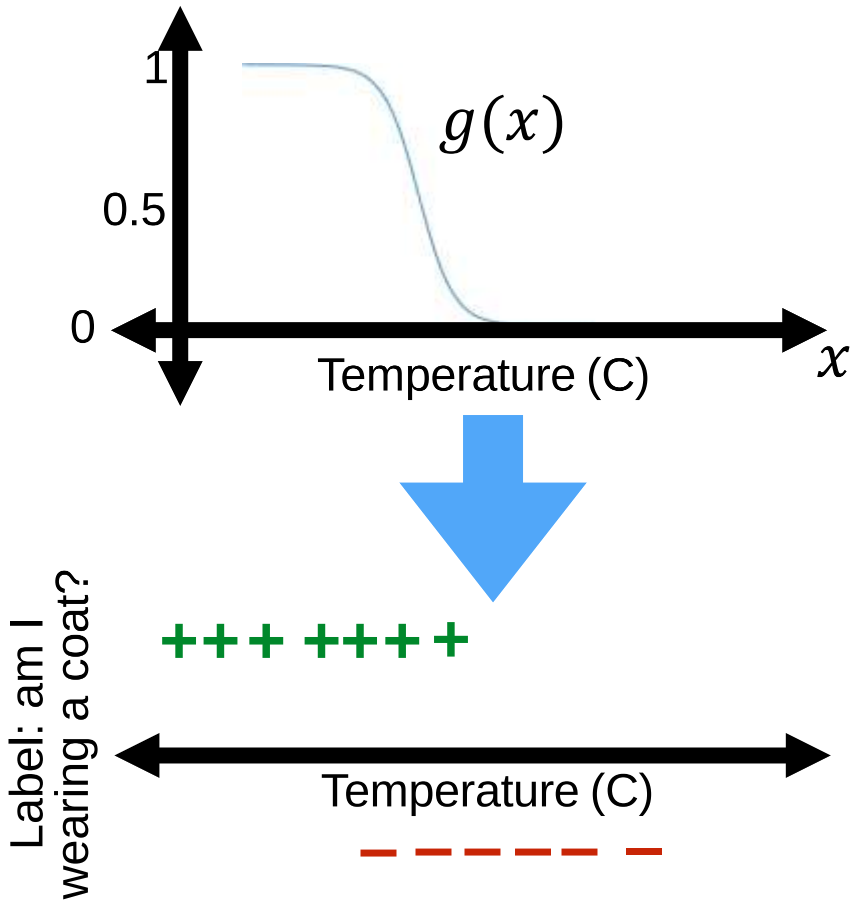




Capturing uncertainty

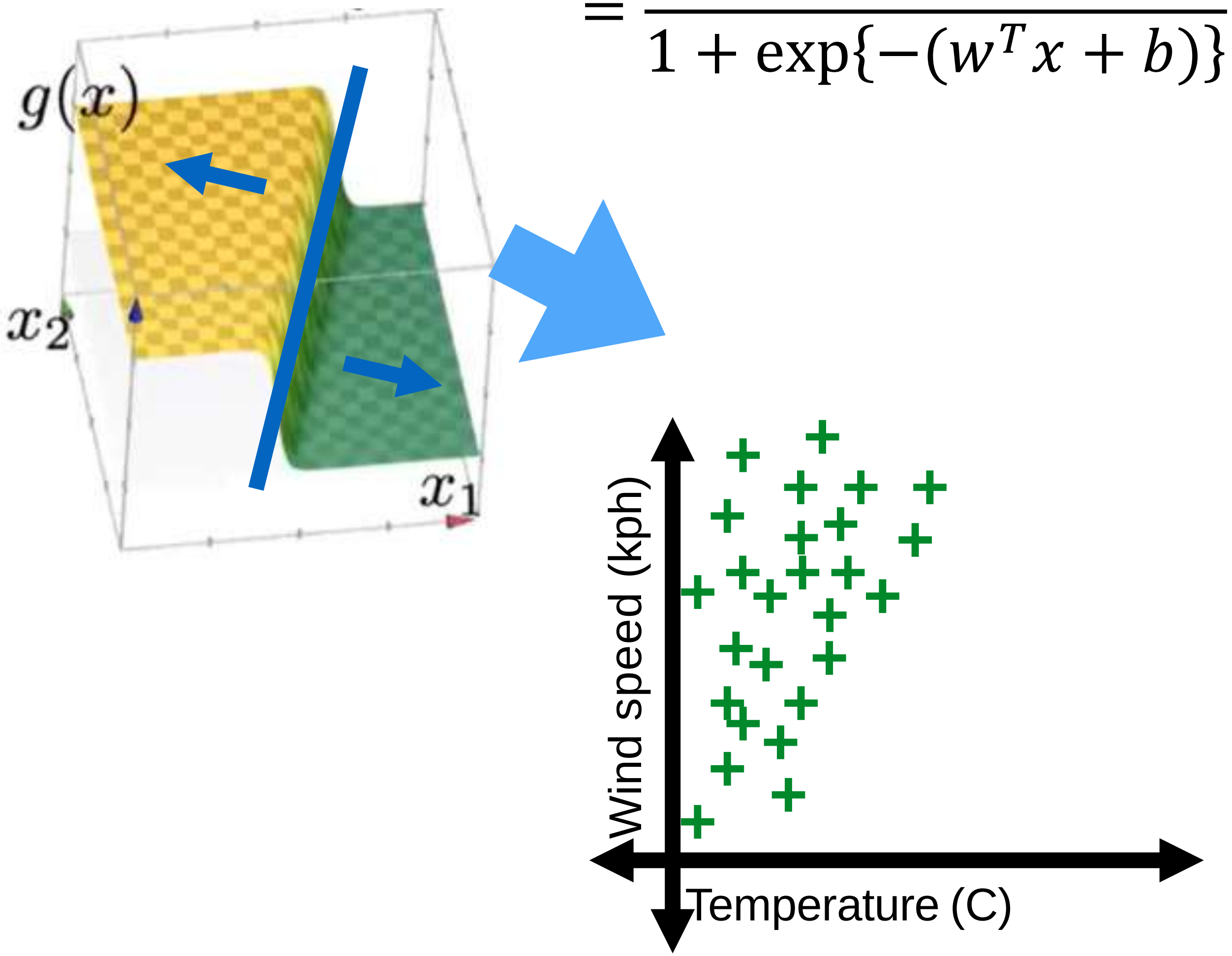
1 feature:

$$g(x) = \sigma(wx + b)$$
$$= \frac{1}{1 + \exp\{-(wx + b)\}}$$



2 features:

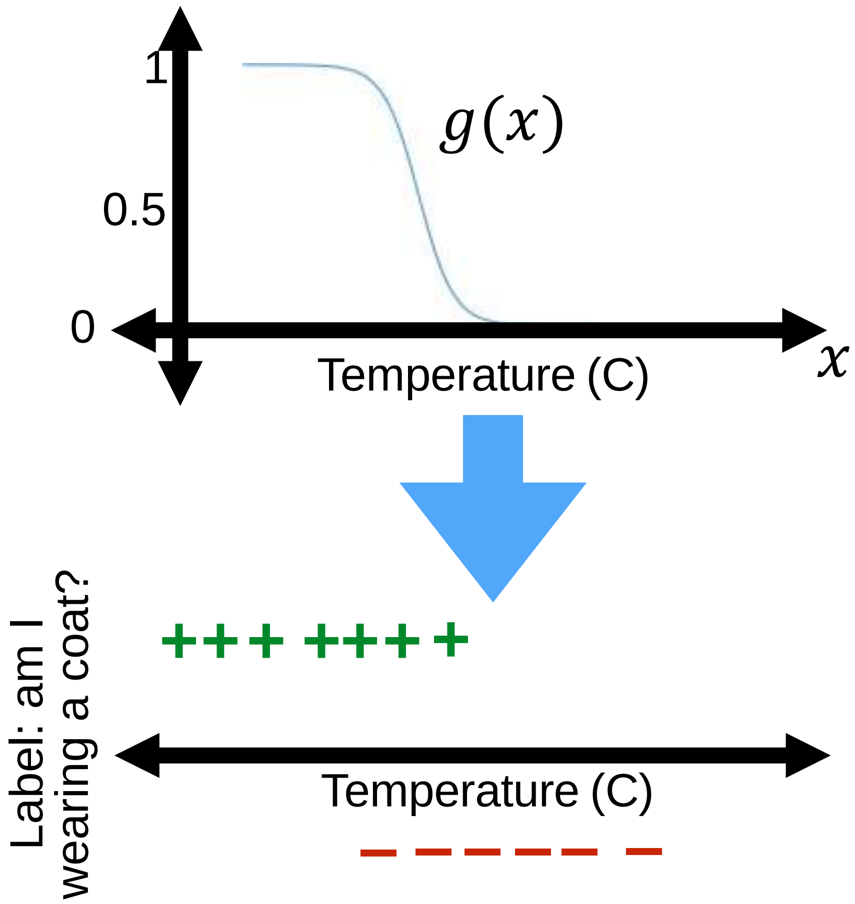
$$g(x) = \sigma(w^T x + b)$$
$$= \frac{1}{1 + \exp\{-(w^T x + b)\}}$$



Capturing uncertainty

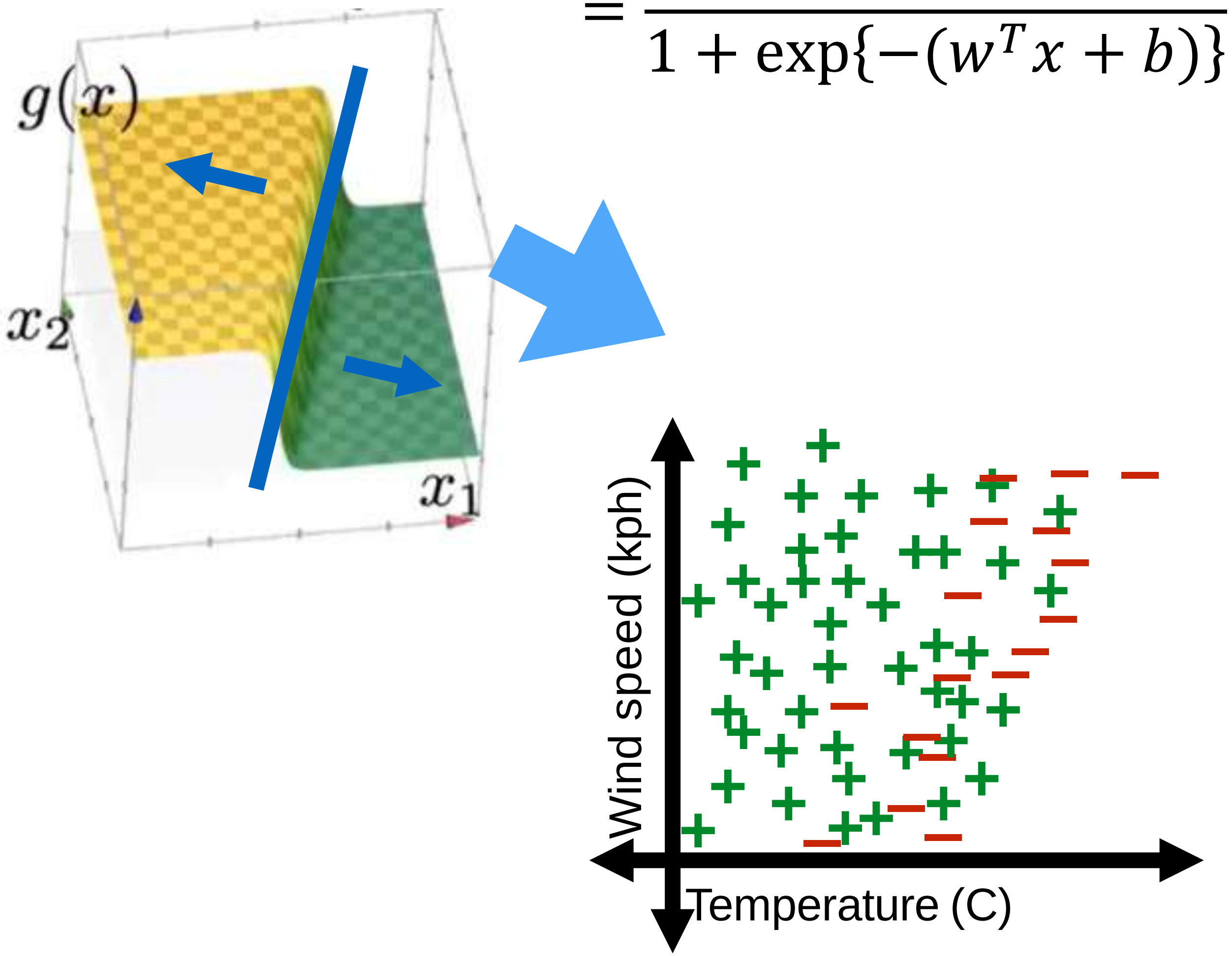
1 feature:

$$g(x) = \sigma(wx + b) = \frac{1}{1 + \exp\{-(wx + b)\}}$$



2 features:

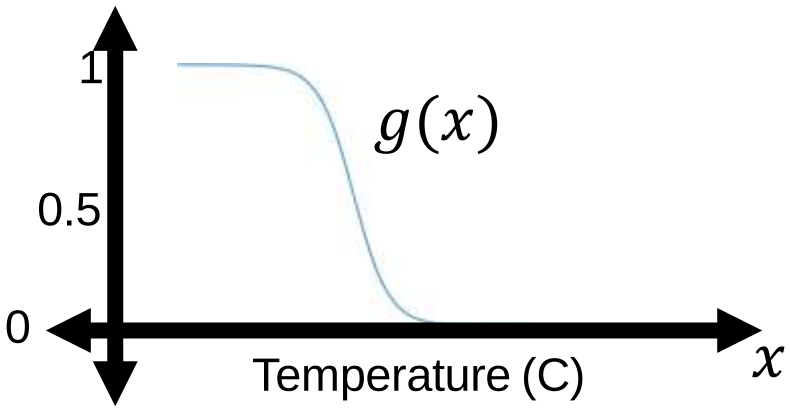
$$g(x) = \sigma(w^T x + b) = \frac{1}{1 + \exp\{-(w^T x + b)\}}$$

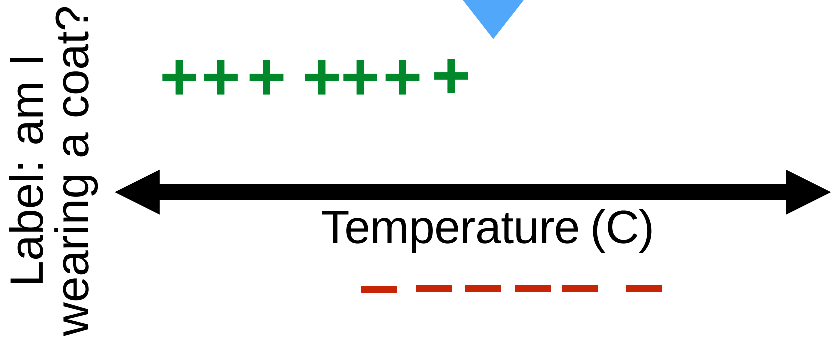


Capturing uncertainty

1 feature:

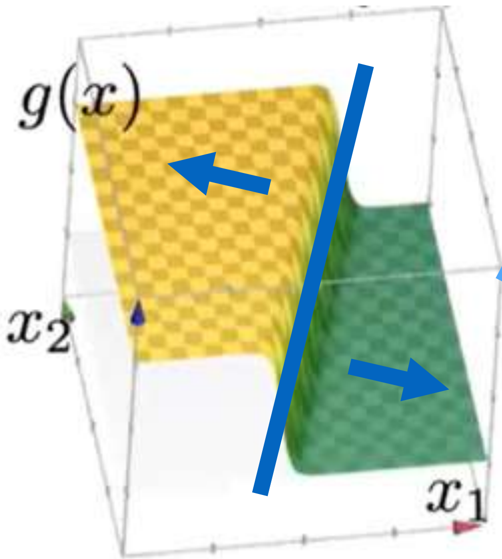
$$g(x) = \sigma(wx + b) = \frac{1}{1 + \exp\{-(wx + b)\}}$$

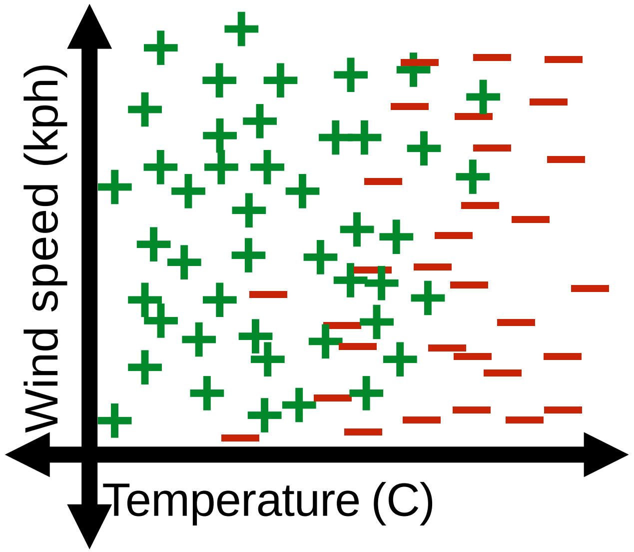




2 features:

$$g(x) = \sigma(w^T x + b) = \frac{1}{1 + \exp\{-(w^T x + b)\}}$$

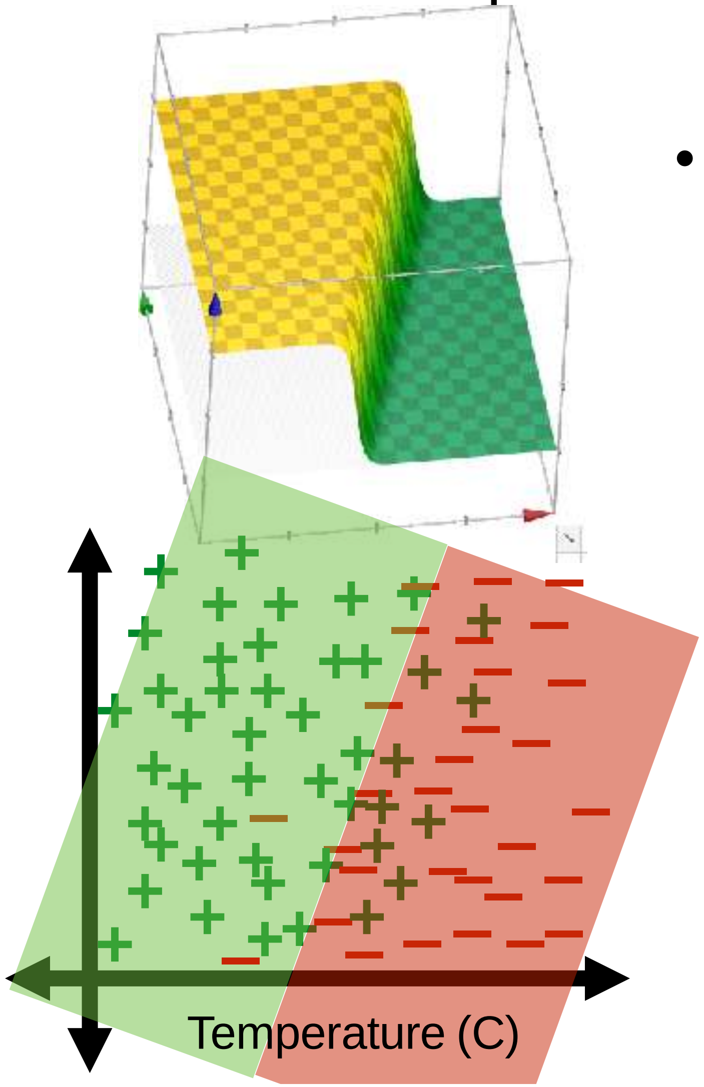




Linear Logistic Regression

aka logistic regression

- How do we learn a classifier (i.e. learn w, b)?
- How do we make predictions?



- Idea: predict +1 if probability >0.5

$$\sigma\left(w^T x + b\right) > 0.5$$

$$\frac{1}{1 + \exp\{-(w^T x + b)\}} > 0.5$$

$$\exp\left\{-\left(w^T x + b\right)\right\} < 1$$

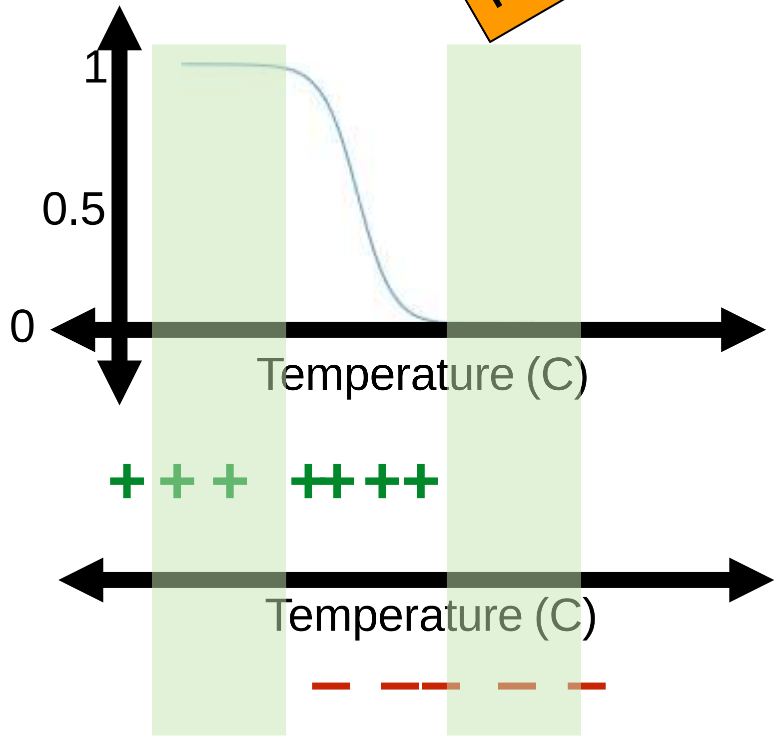
$$w^T x + b > 0$$

- Same hypothesis class as before! But we will get:
  - Uncertainties
  - Quality guarantees when data not linearly separable

Linear Logistic Regression

- How do we learn a classifier (i.e. learn w, b)?

aka logistic regression



Linear Logistic Regression

aka logistic regression

- How do we learn a classifier (i.e. learn w, b)?

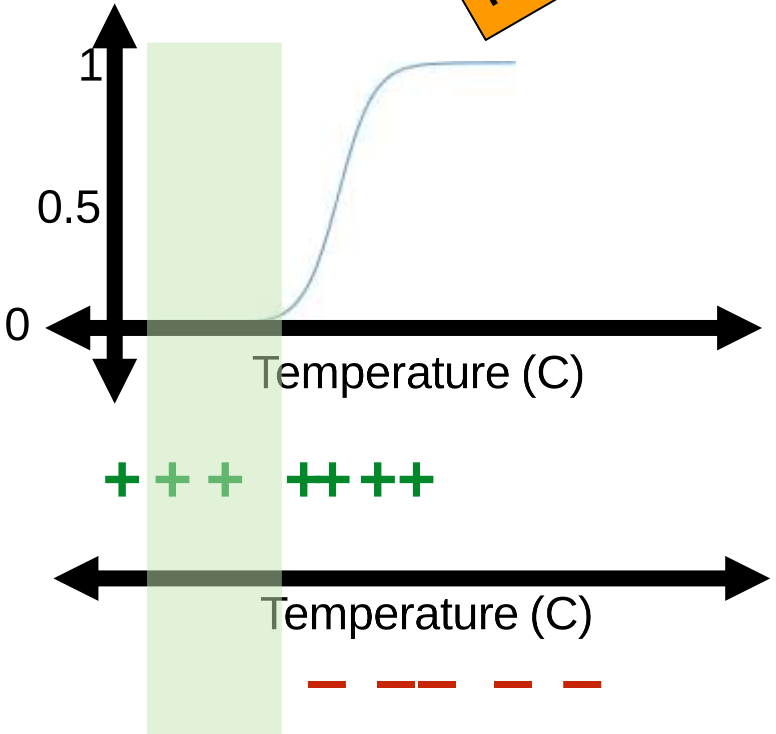
Probability (data)

= $\prod$ Probability (data point i)

[Let $g^{(i)} = \sigma(w^T x^{(i)} + b)$]

$$= \prod \begin{cases} g^{(i)} \; if \; y^{(i)} = +1 \\ \left(1 - g^{(i)}\right) \; else \end{cases}$$

$$= \prod \left\{ (g^{(i)})^{1\{y^{(i)}=+1\}} (1 - g^{(i)})^{1\{y^{(i)}\neq+1\}} \right.$$



Loss (data) = -(1/n)* -log probability (data)

$$= \frac{1}{n} \sum_{i=1}^{n} -\left(1\{y^{(i)} = +1\} \log g^{(i)} + 1\{y^{(i)} \neq +1\} \log(1 - g^{(i)})\right)$$

Negative log likelihood loss (g for guess, a for actual):

$$-L_{nll}(g, a) = 1\{a = +1\} \log g + 1\{a \neq +1\} \log(1 - g)$$

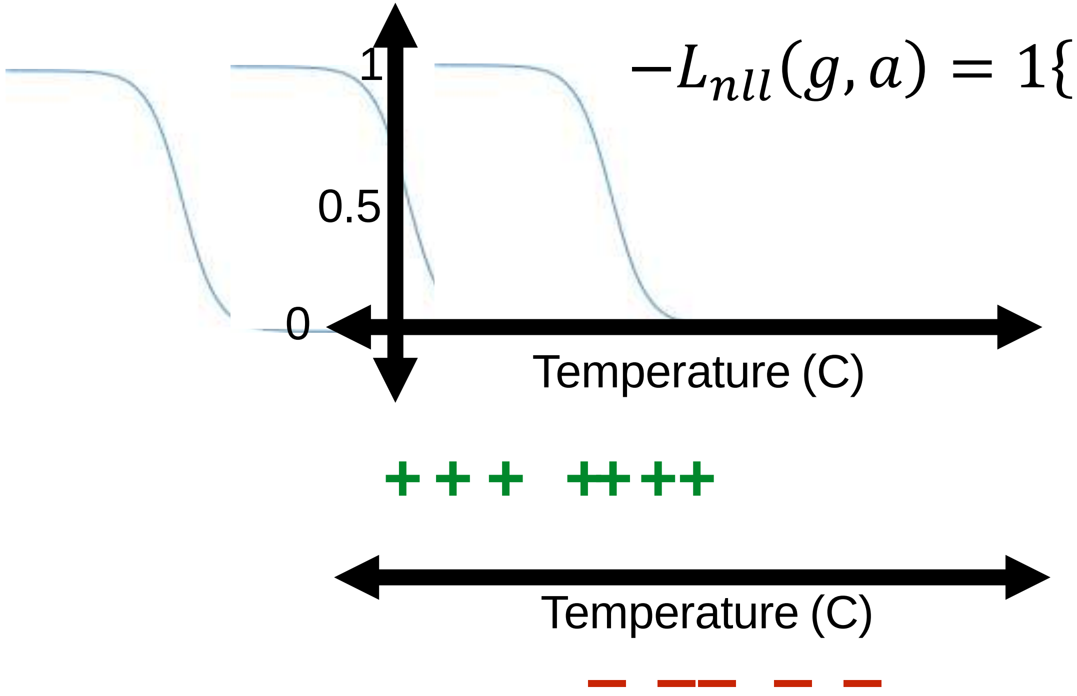
Linear Logistic Regression

aka logistic regression

- How do we learn a classifier (i.e. learn w, b)?
- Want to find parameter values to minimize average (negative loglikelihood) loss across the data

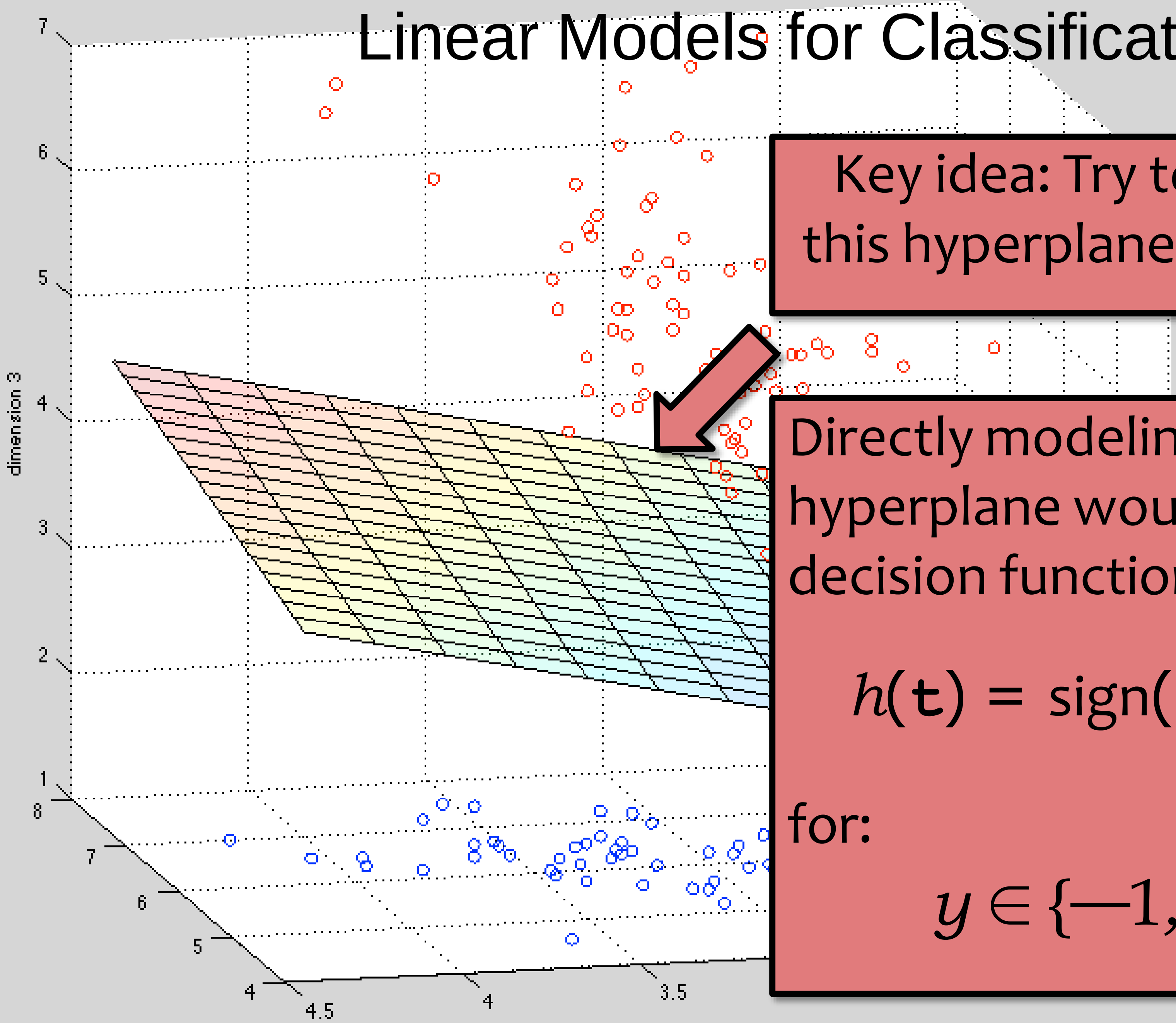
$$J(w,b) = \frac{1}{n}\sum_{i=1}^{n} L_{nll}\left(\sigma(w^T x^{(i)} + b), y^{(i)}\right)$$

$$-L_{nll}(g,a) = 1\{a = +1\}\log g + 1\{a \neq +1\}\log(1-g)$$



Linear Models for Classification

Recall…



Key idea: Try to learn this hyperplane directly

Directly modeling the hyperplane would use a decision function:

$$h(\mathbf{t}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{t})$$

for:

$$y \in \{-1, +1\}$$

Function Approximation

Previously, we assumed that our output was generated using a **deterministic target function:**

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$

$$y^{(i)} = c^*(\mathbf{x}^{(i)})$$

Our goal was to learn a hypothesis h(**x**) that best approximates c*(**x**)

Probabilistic Learning

Today, we assume that our output is **sampled** from a conditional **probability distribution:**

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$

$$y^{(i)} \sim p^*(\cdot|\mathbf{x}^{(i)})$$

Our goal is to learn a probability distribution p(y|**x**) that best approximates p*(y|**x**)

Likelihood Function

One R.V.

Given N **independent, identically distributed (iid)** samples D = $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ from a discrete **random variable** X with probability mass function (*pmf*) $p(x|\theta)$ …

- Case 1: The **likelihood** function

  $$L(\theta) = p(x^{(1)}|\theta)\, p(x^{(2)}|\theta) \ldots\, p(x^{(N)}|\theta)$$

The **likelihood** tells us how likely one sample is relative to another

- Case 2: The **log-likelihood** function is

  $$\ell(\theta) = \log p(x^{(1)}|\theta) + \ldots + \log p(x^{(N)}|\theta)$$

Likelihood Function

Two R.V.s

Given N **iid** samples $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ from a pair of **random variables** X, Y where Y is **discrete** with probability mass function (pmf) $p(y \mid x, \theta)$

- Case 3: The **conditional likelihood** function:

$$L(\theta) = p(y^{(1)} \mid x^{(1)}, \theta) \ldots p(y^{(N)} \mid x^{(N)}, \theta)$$

- Case 4: The **conditional log-likelihood** function is

$$\ell(\theta) = \log p(y^{(1)} \mid x^{(1)}, \theta) + \ldots + \log p(y^{(N)} \mid x^{(N)}, \theta)$$

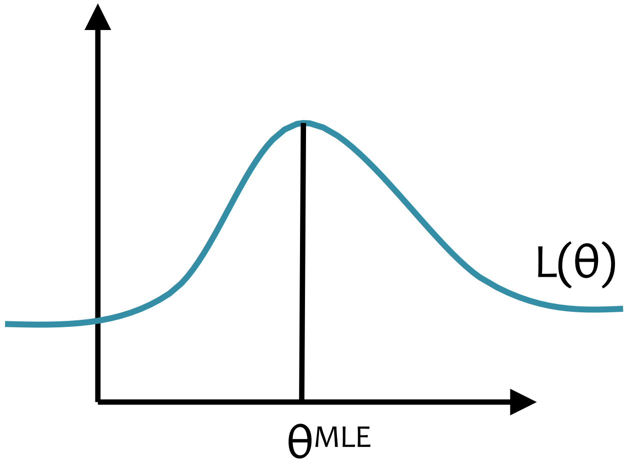
上海科技大学
ShanghaiTech University

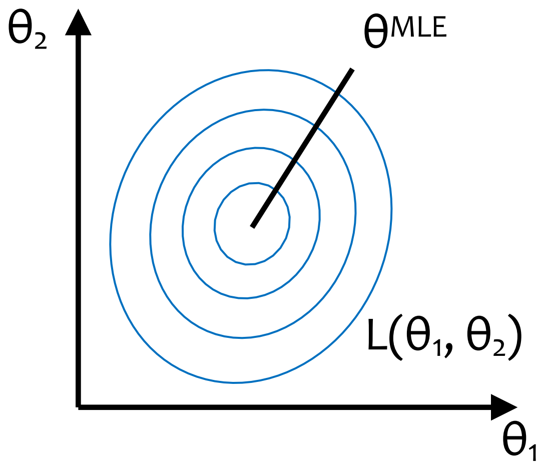
Suppose we have data $D = \{x^{(i)}\}_{i=1}^{N}$

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the likelihood of the data.

$$\boldsymbol{\theta}^{MLE} = arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(x^{(i)}|\boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)





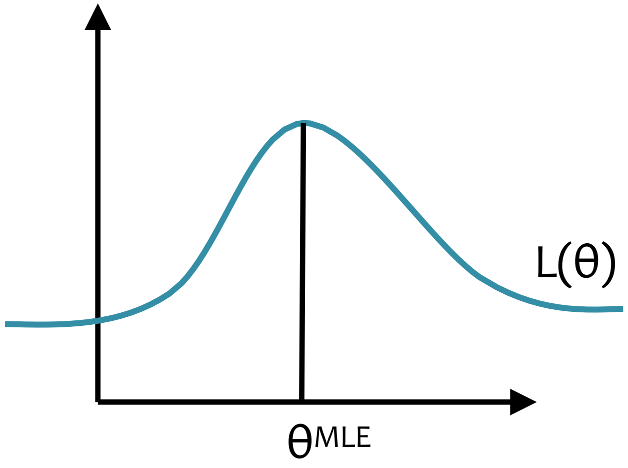
MLE

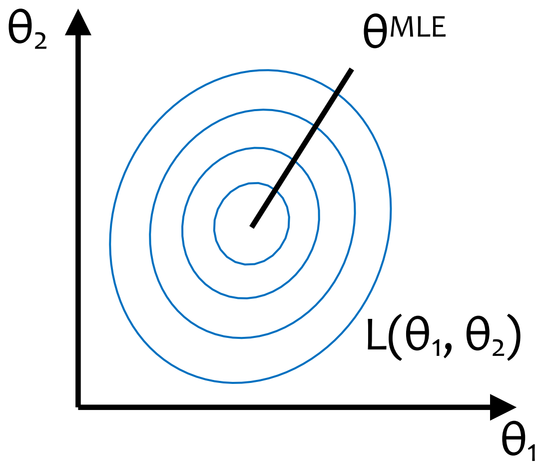
Suppose we have data $D = \{(y^{(i)}, x^{(i)})\}_{i=1}^{N}$

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the conditional likelihood of the data.

$$\boldsymbol{\theta}^{MLE} = arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(y^{(i)}|x^{(i)}, \boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)





上海科技大学 ShanghaiTech University

Suppose we have data $\mathrm{D} = \{(y^{(i)}, \mathrm{x}^{(i)})\}_{i=1}^{N}$

Principle of Maximum Likelihood Estimation:

Choose the parameters that maximize the conditional log-likelihood of the data.

$$\boldsymbol{\theta}^{MLE} = arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(y^{(i)}|x^{(i)}, \boldsymbol{\theta})$$

$$= arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}, \boldsymbol{\theta})$$

$$= arg\min_{\boldsymbol{\theta}} -\sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}, \boldsymbol{\theta})$$

What does maximizing likelihood accomplish?

- There is only a finite amount of probability mass (i.e. sum-to-one constraint)
- MLE tries to allocate **as much** probability mass **as possible** to the things we have observed…

  …**at the expense** of the things we have **not** observed

Data: Inputs are continuous vectors of length M. Outputs are discrete.

$$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \text{ where } \mathbf{x} \in \mathbb{R}^M \text{ and } y \in \{0, 1\}$$

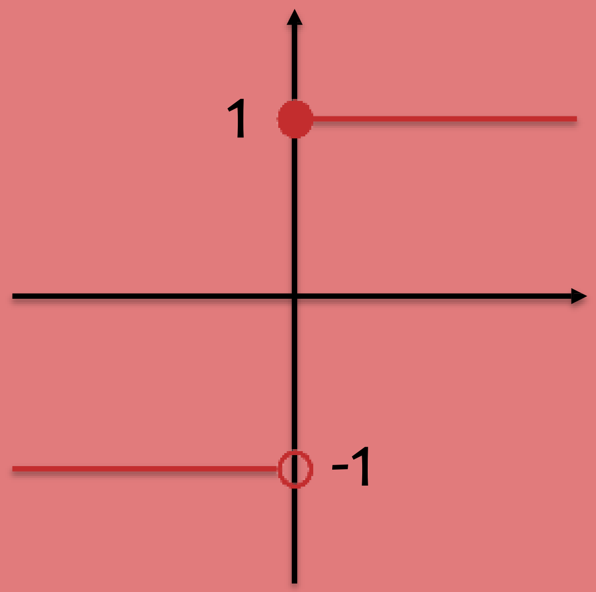
We are back to classification.

Despite the name logistic **regression.**

sign(·) vs. sigmoid(·)

Suppose we wanted to learn a linear classifier, but instead of predicting y ∈ {-1,+1} we wanted to predict y ∈ {0,1}

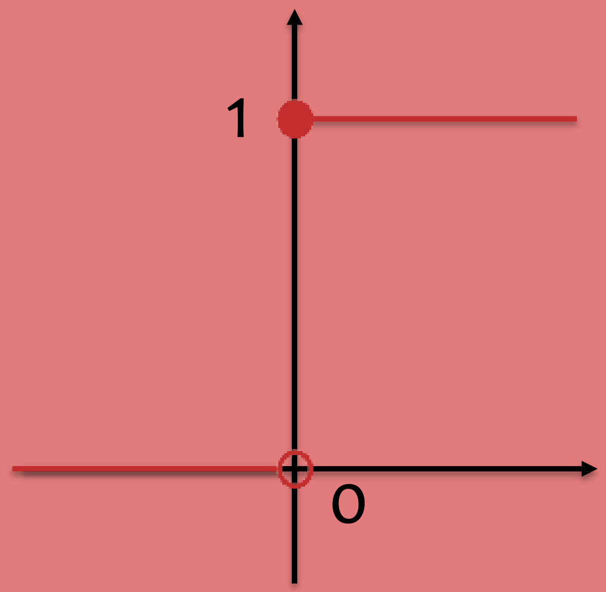
$$h(\mathrm{x}) = \mathrm{sign}(\boldsymbol{\theta}^T \mathbf{x})$$



sign(u)

Suppose we wanted to learn a linear classifier, but instead of predicting y ∈ {-1,+1} we wanted to predict y ∈ {0,1}

$$h(\mathbf{x}) = \text{“sign”}(\boldsymbol{\theta}^T \mathbf{x})$$



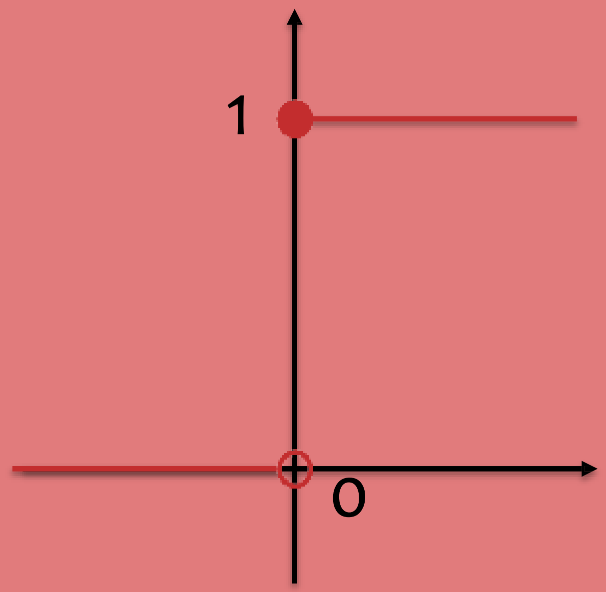
“sign”(u)

Goal: Learn a linear classifier with Gradient Descent

sign(·) vs. sigmoid(·)

But this decision function isn't differentiable…

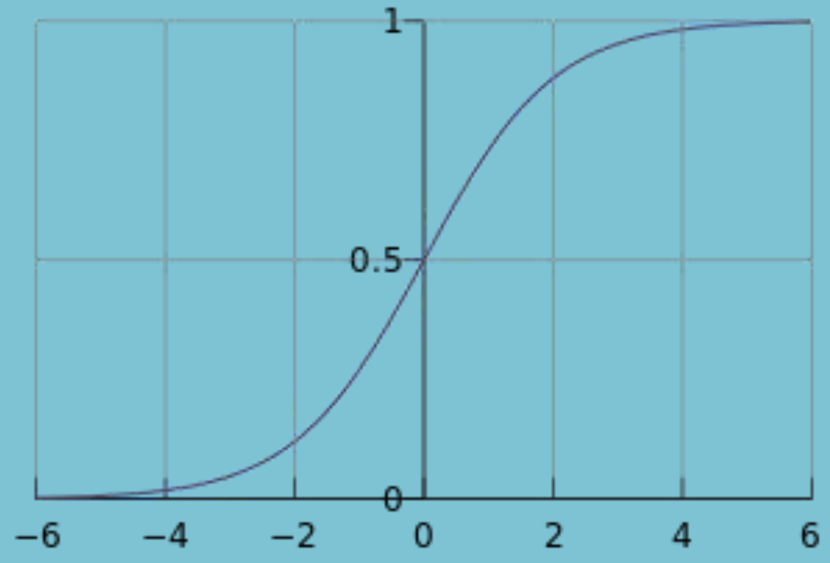
$$h(\mathbf{x}) = \text{“sign”}(\boldsymbol{\theta}^T \mathbf{x})$$



“sign”(u)

Use a differentiable function instead!

$$p_{\boldsymbol{\theta}}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \mathbf{x})}$$



$$\text{logistic}(u) \equiv \frac{1}{1 + e^{-u}}$$

The *logistic* function is also called the *sigmoid* function.

Data: Inputs are continuous vectors of length M. Outputs are discrete.

$$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \text{ where } \mathbf{x} \in \mathbb{R}^M \text{ and } y \in \{0, 1\}$$

Model: Logistic function applied to dot product of parameters with input vector.

$$p_{\boldsymbol{\theta}}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \mathbf{x})}$$

Learning: finds the parameters that minimize some objective function.

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\boldsymbol{\theta})$$

Prediction: Output is the most probable class.

$$\hat{y} = \underset{y \in \{0,1\}}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(y|\mathbf{t})$$

Logistic Regression

上海科技大学 ShanghaiTech University

- Suppose we have binary labels $y \in \{0,1\}$ and D-dimensional inputs $\boldsymbol{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$

1 prepended to $\boldsymbol{x}$

1. Model

$$
\begin{aligned}
g(x) &= \sigma(w^T x + b) \\
&= \frac{1}{1 + \exp\{-(w^T x + b)\}} \\
&= \frac{1}{1 + \exp\{-(\boldsymbol{\theta}^T \boldsymbol{x})\}}
\end{aligned}
$$

$$
p(y|x, \boldsymbol{\theta}) = \begin{cases} g(x) & \text{if } y = 1 \\ 1 - g(x) & \text{if } y = 0 \end{cases}
$$

Logistic Regression

上海科技大学
ShanghaiTech University

1. Model

$$g(x) = \sigma(w^T x + b)$$
$$= \frac{1}{1 + \exp\{-(w^T x + b)\}}$$
$$= \frac{1}{1 + \exp\{-(\boldsymbol{\theta}^T \boldsymbol{x})\}}$$

$$p(y|x, \boldsymbol{\theta}) = \begin{cases} g(x) & \text{if } y = 1 \\ 1 - g(x) & \text{if } y = 0 \end{cases}$$

$$P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})} = \frac{\exp(\boldsymbol{\theta}^T \boldsymbol{x})}{\exp(\boldsymbol{\theta}^T \boldsymbol{x}) + 1}$$

$$P(Y = 0|\boldsymbol{x}, \boldsymbol{\theta}) = 1 - P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{\exp(\boldsymbol{\theta}^T \boldsymbol{x}) + 1}$$

$$\Downarrow$$

$$\frac{P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta})}{P(Y = 0|\boldsymbol{x}, \boldsymbol{\theta})} = \exp(\boldsymbol{\theta}^T \boldsymbol{x})$$

$$\Downarrow$$

$$\log \frac{P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta})}{P(Y = 0|\boldsymbol{x}, \boldsymbol{\theta})} = \boldsymbol{\theta}^T \boldsymbol{x}$$

1. Model

$$g(x) = \sigma(w^T x + b)$$
$$= \frac{1}{1 + \exp\{-(w^T x + b)\}}$$
$$= \frac{1}{1 + \exp\{-(\boldsymbol{\theta}^T x)\}}$$

$$p(y|x, \boldsymbol{\theta}) = \begin{cases} g(x) & \text{if } y = 1 \\ 1 - g(x) & \text{if } y = 0 \end{cases}$$

2. Objective

$$J(w, b) = \frac{1}{n} \sum_{i=1}^{n} L_{nll}\left(\sigma(w^T x^{(i)} + b), y^{(i)}\right)$$

$$J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L_{nll}\left(\sigma(\boldsymbol{\theta}^T x^{(i)}), y^{(i)}\right)$$

$$J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} -\log p(y^{(i)}|x^{(i)}, \boldsymbol{\theta})$$

$$-L_{nll}(g, a) = 1\{a = +1\} \log g + 1\{a \neq +1\} \log(1 - g)$$

2. Objective

- Find $\boldsymbol{\theta}$ that minimizes

$$\ell(\boldsymbol{\theta}) = -\frac{1}{N}\log P\left(y^{(1)},\dots,y^{(N)}\middle|\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)},\boldsymbol{\theta}\right)$$

$$= -\frac{1}{N}\log\prod_{n=1}^{N} P\left(y^{(n)}\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)$$

$$= -\frac{1}{N}\log\prod_{n=1}^{N} P\left(Y=1\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)^{y^{(n)}}\left(P\left(Y=0\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)\right)^{1-y^{(n)}}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\log P\left(Y=1\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)+\left(1-y^{(n)}\right)\log P\left(Y=0\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\log\frac{P\left(Y=1\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)}{P\left(Y=0\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)}+\log P\left(Y=0\middle|\boldsymbol{x}^{(n)},\boldsymbol{\theta}\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}-\log\left(1+\exp\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}\right)\right)$$

3. Gradients

Minimizing the Negative Conditional (log-)Likelihood

上海科技大学 ShanghaiTech University

$$J(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\boldsymbol{\theta}^T\boldsymbol{x}^{(n)} - \log\left(1 + \exp\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}\right)\right)$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\nabla_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}\right) - \nabla_{\boldsymbol{\theta}}\log\left(1 + \exp\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}\right)\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\boldsymbol{x}^{(n)} - \frac{\exp\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}\right)}{1 + \exp\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)}\right)}\boldsymbol{x}^{(n)}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{\theta}\right) - y^{(n)}\right)$$

Learning Logistic Regression

上海科技大学
ShanghaiTech University

Learning: Four approaches to solving $\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\boldsymbol{\theta})$

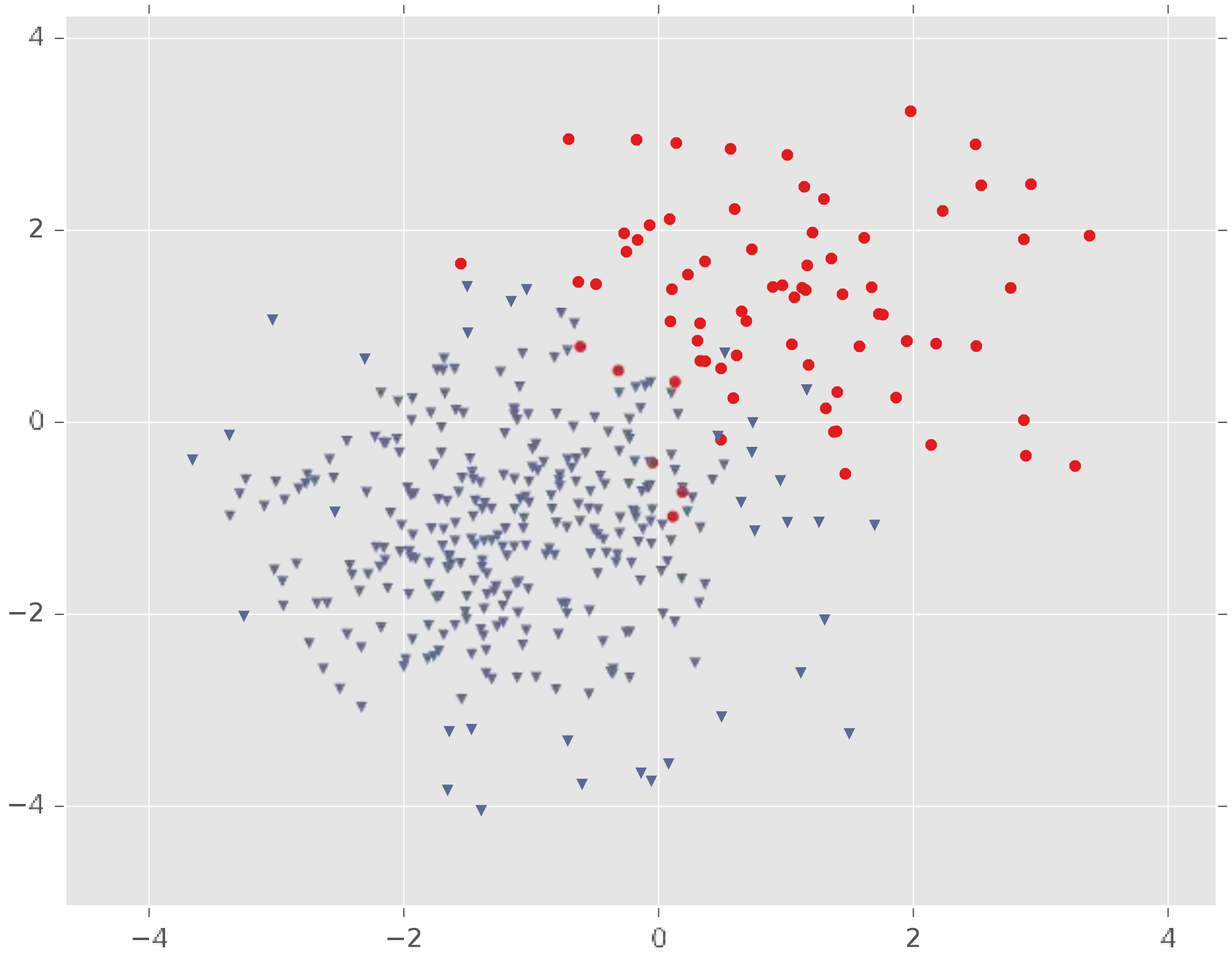
Approach 0: Random Search
(horridly slow because it lacks gradient information)

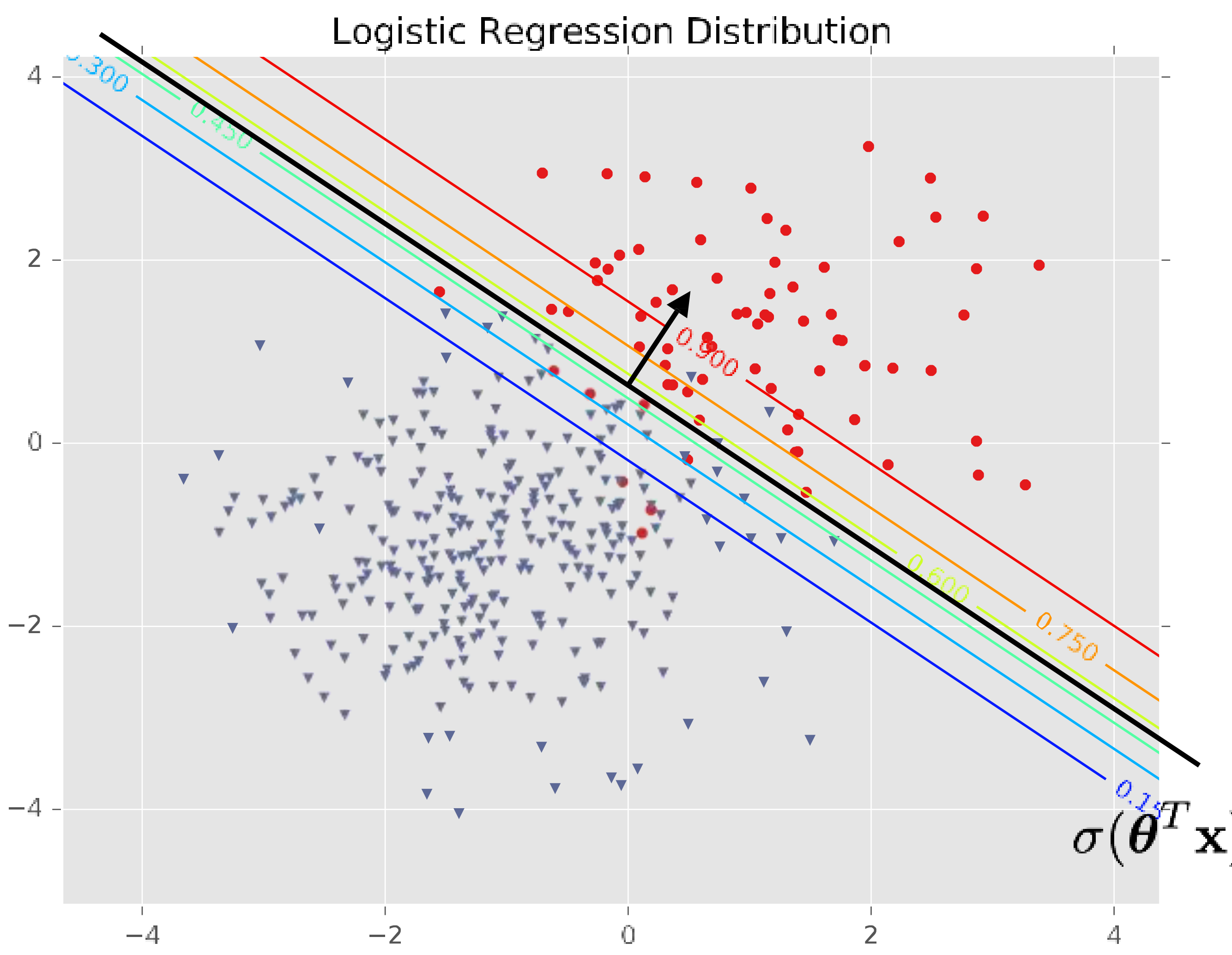
Approach 1: Gradient Descent
(take large confident steps opposite the gradient)

Approach 2: Stochastic Gradient Descent (SGD)
(take many small steps roughly opposite the gradient)

~~**Approach 3:** Closed Form~~
~~(set derivatives equal to zero and solve for parameters)~~

Logistic Regression does not have a closed form solution for MLE parameters.


4
2
0
−2
−4
−4
−2
0
2
4



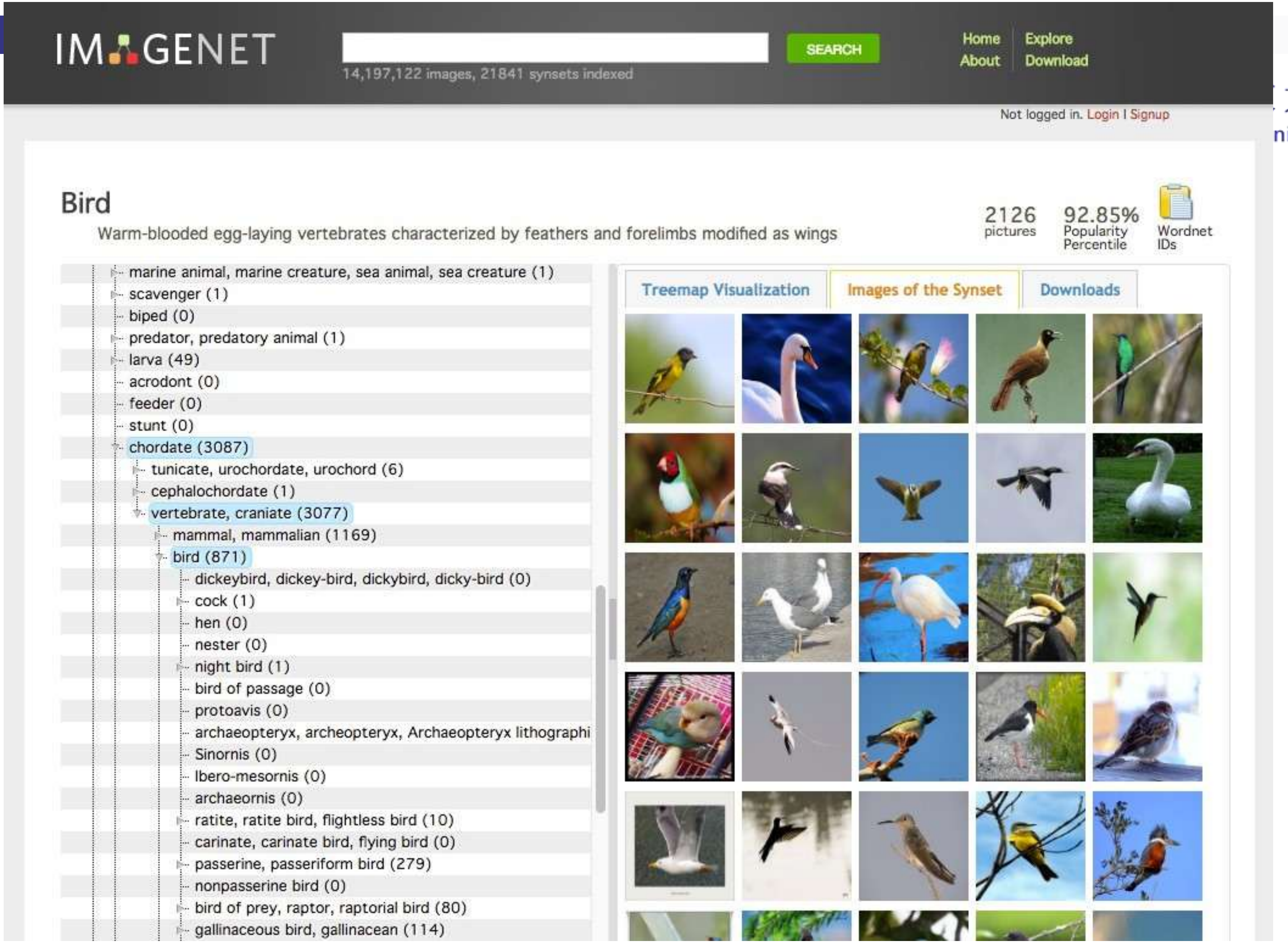
$$\sigma(\boldsymbol{\theta}^T \mathbf{x}) = 0.5$$

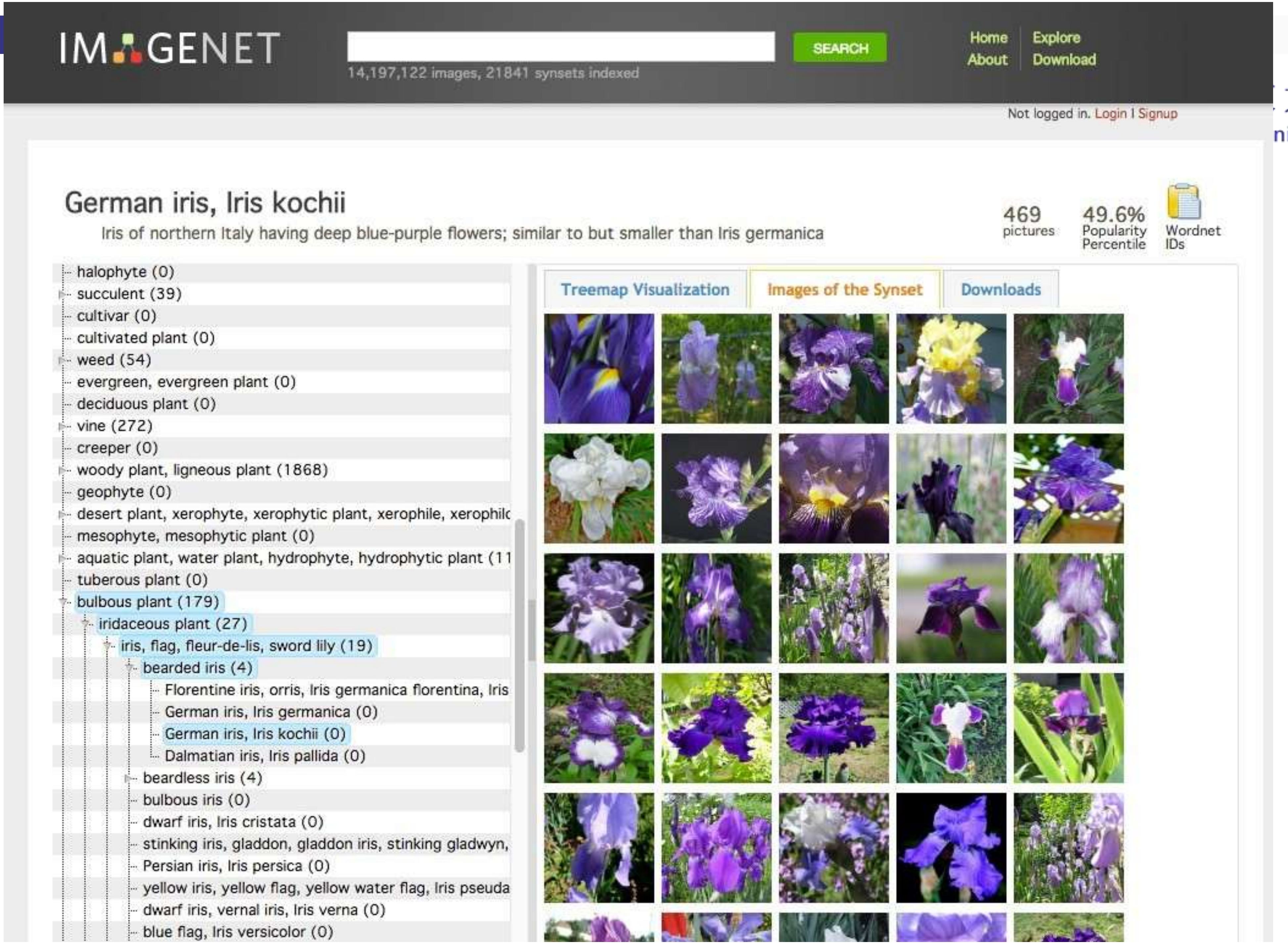

Classification with Logistic Regression
4
2
0
−2
−4
−4
−2
0
2
4

Example: Image Classification

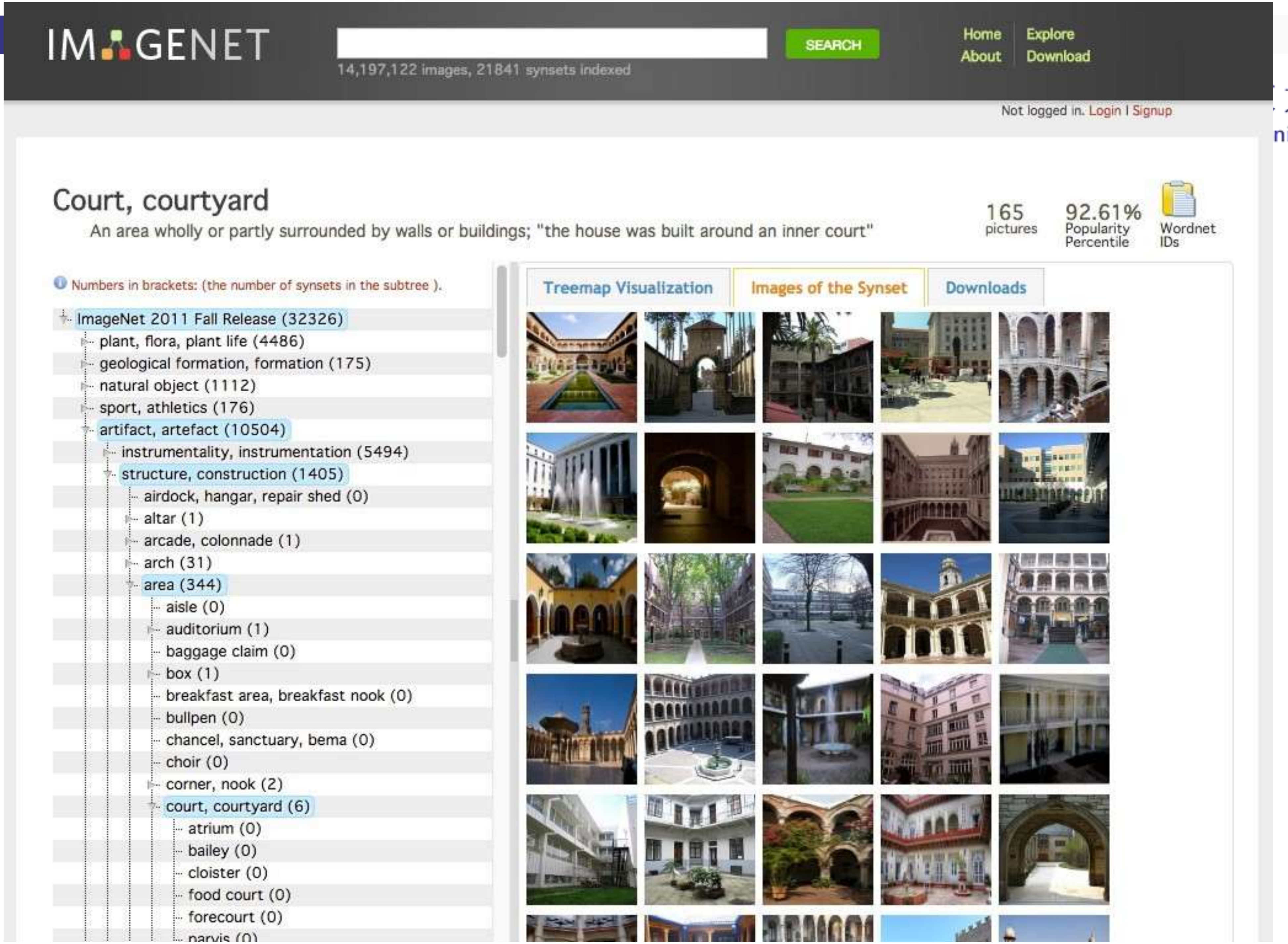
上海科技大学
ShanghaiTech University

- ImageNet LSVRC-2010 contest:
  - **Dataset:** 1.2 million labeled images, 1000 classes
  - **Task:** Given a new image, label it with the correct class
  - **Multiclass** classification problem
- Examples from http://image-net.org/


IMAGENET
SEARCH
14,197,122 images, 21841 synsets indexed
Home
About
Explore
Download
Not logged in. Login I Signup
Bird
Warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings
2126 pictures
92.85% Popularity Percentile
Wordnet IDs
marine animal, marine creature, sea animal, sea creature (1)
scavenger (1)
biped (0)
predator, predatory animal (1)
larva (49)
acrodont (0)
feeder (0)
stunt (0)
chordate (3087)
tunicate, urochordate, urochord (6)
cephalochordate (1)
vertebrate, craniate (3077)
mammal, mammalian (1169)
bird (871)
dickeybird, dickey-bird, dickybird, dicky-bird (0)
cock (1)
hen (0)
nester (0)
night bird (1)
bird of passage (0)
protoavis (0)
archaeopteryx, archeopteryx, Archaeopteryx lithographi
Sinornis (0)
Ibero-mesornis (0)
archaeornis (0)
ratite, ratite bird, flightless bird (10)
carinate, carinate bird, flying bird (0)
passerine, passeriform bird (279)
nonpasserine bird (0)
bird of prey, raptor, raptorial bird (80)
gallinaceous bird, gallinacean (114)
Treemap Visualization
Images of the Synset
Downloads



大学
niversity


IMAGENET
SEARCH
14,197,122 images, 21841 synsets indexed
Home
About
Explore
Download
Not logged in. Login I Signup
Court, courtyard
An area wholly or partly surrounded by walls or buildings; "the house was built around an inner court"
165
pictures
92.61%
Popularity Percentile
Wordnet IDs
Numbers in brackets: (the number of synsets in the subtree).
ImageNet 2011 Fall Release (32326)
plant, flora, plant life (4486)
geological formation, formation (175)
natural object (1112)
sport, athletics (176)
artifact, artefact (10504)
instrumentality, instrumentation (5494)
structure, construction (1405)
airdock, hangar, repair shed (0)
altar (1)
arcade, colonnade (1)
arch (31)
area (344)
aisle (0)
auditorium (1)
baggage claim (0)
box (1)
breakfast area, breakfast nook (0)
bullpen (0)
chancel, sanctuary, bema (0)
choir (0)
corner, nook (2)
court, courtyard (6)
atrium (0)
bailey (0)
cloister (0)
food court (0)
forecourt (0)
parvis (0)
Treemap Visualization
Images of the Synset
Downloads

Example: Image Classification

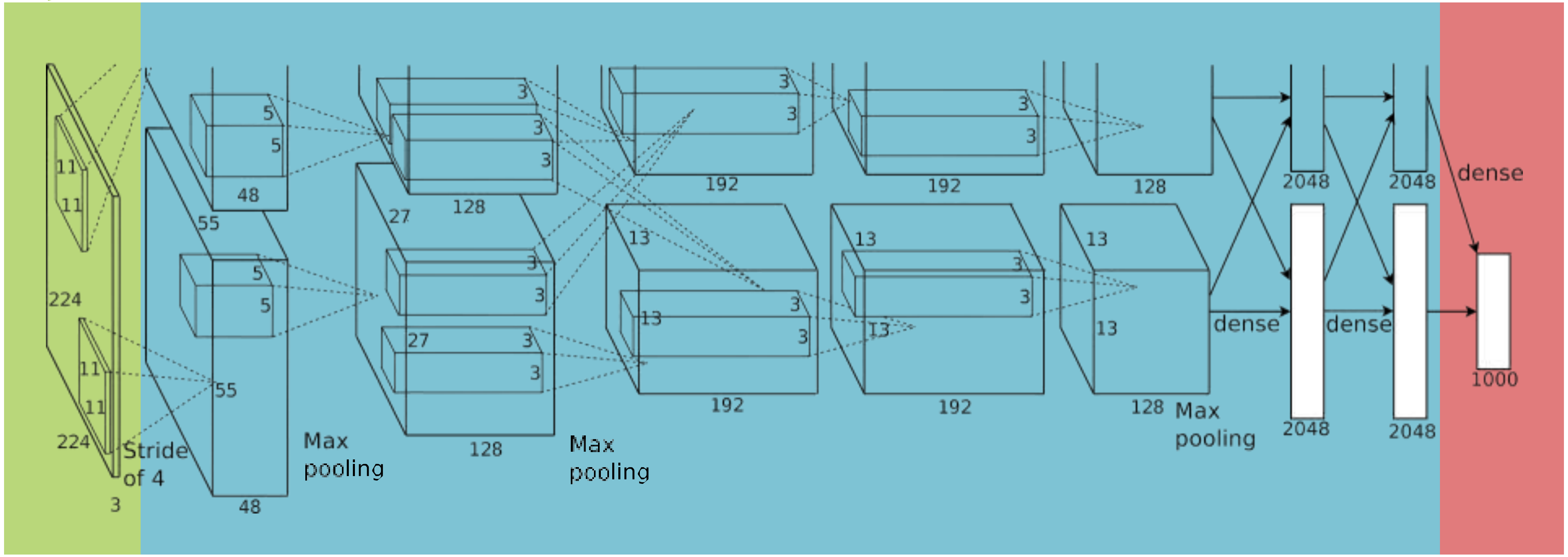
上海科技大学
ShanghaiTech University

CNN for Image Classification
(Krizhevsky, Sutskever & Hinton, 2011)
17.5% error on ImageNet LSVRC-2010 contest

Input image (pixels)

- Five convolutional layers (w/max-pooling)
- Three fully connected layers

1000-way softmax



Example: Image Classification

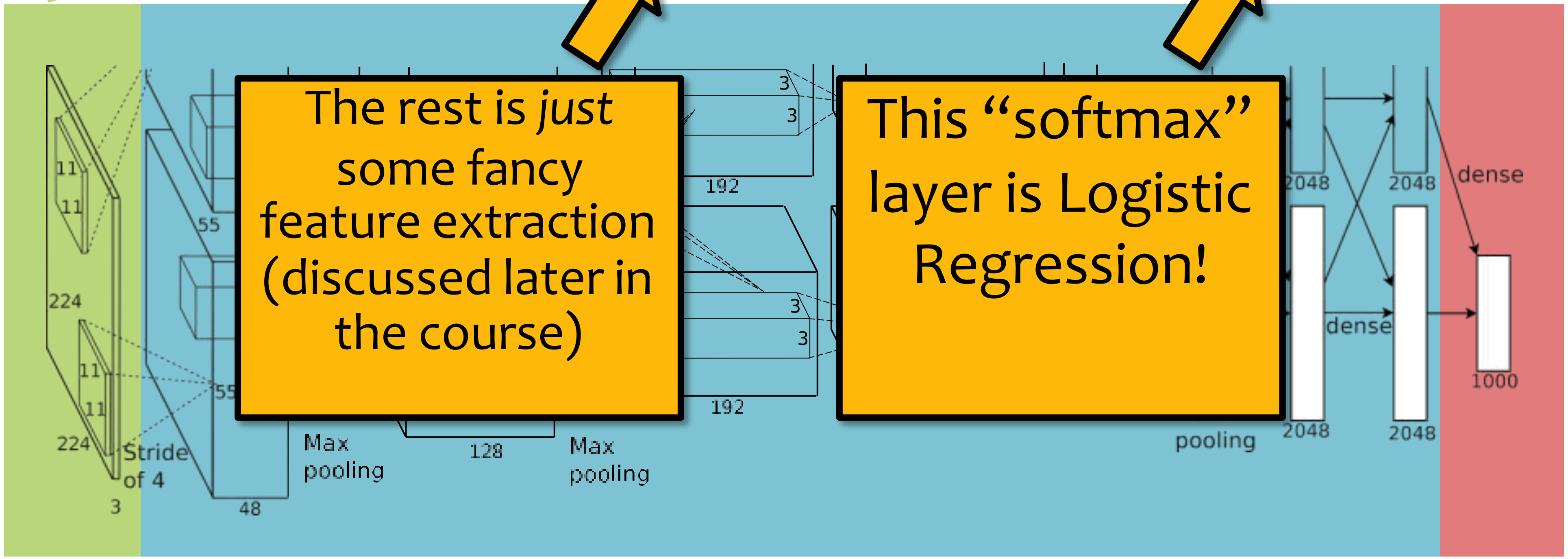
上海科技大学
ShanghaiTech University

CNN for Image Classification
(Krizhevsky, Sutskever & Hinton, 2011)
17.5% error on ImageNet LSVRC-2010 contest

Input image (pixels)

- Five convolutional layers (w/max-pooling)
- Three fully connected layers

1000-way softmax



The rest is *just* some fancy feature extraction (discussed later in the course)

This "softmax" layer is Logistic Regression!

Linear Models

Matching Game

上海科技大学
ShanghaiTech University

Poll Question:

Match the Algorithm to its Update Rule

1. SGD for Logistic Regression $h_\theta(\mathrm{x}) = p(y = 1 \mid \mathrm{x})$	**4.** $\theta_k \leftarrow \theta_k + (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})$
2. Least Mean Squares $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}$	**5.** $\theta_k \leftarrow \theta_k + \dfrac{1}{1 + \exp \lambda(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})}$
3. Perceptron $h_{\boldsymbol{\theta}}(\mathbf{x}) = \mathrm{sign}(\boldsymbol{\theta}^T \mathbf{x})$	**6.** $\theta_k \leftarrow \theta_k + \lambda(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})x_k^{(i)}$

Answer:

A. 1=5, 2=4, 3=6
B. 1=5, 2=6, 3=4
C. 1=6, 2=4, 3=4
D. 1=5, 2=6, 3=6
E. 1=6, 2=6, 3=6
F. 1=6, 2=5, 3=5
G. 1=5, 2=5, 3=5
H. 1=4, 2=5, 3=6
I. None of the above

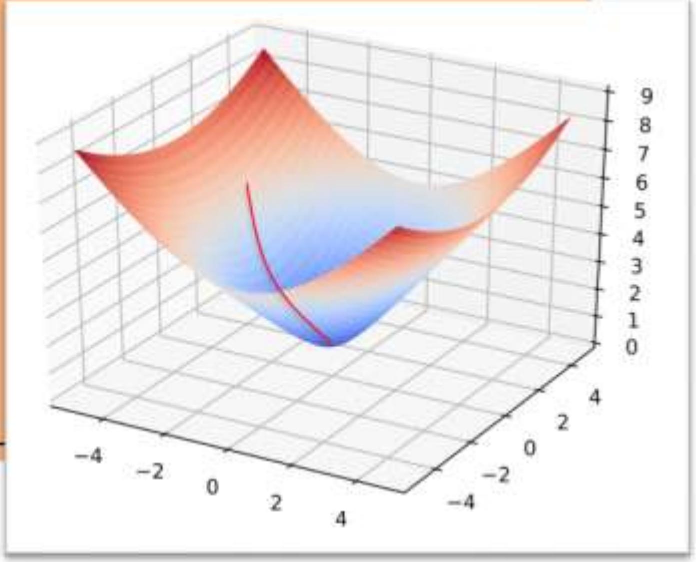
Gradient Descent

上海科技大学 ShanghaiTech University

Recall...

Algorithm 1 Gradient Descent

1: **procedure** GD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
2: $\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
3: $\quad$**while** not converged **do**
4: $\quad\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla J(\boldsymbol{\theta})$
5: $\quad$**return** $\boldsymbol{\theta}$



per-example objective:

$$J^{(i)}(\boldsymbol{\theta})$$

original objective:

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \begin{bmatrix} \frac{d}{d\theta_1} J(\boldsymbol{\theta}) \\ \frac{d}{d\theta_2} J(\boldsymbol{\theta}) \\ \vdots \\ \frac{d}{d\theta_M} J(\boldsymbol{\theta}) \end{bmatrix}$$

Gradient Descent

- Input: training dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}^{(i)} \left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)} \right)$$

   a. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}^{(t)}\right)$
   b. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

上海科技大学
ShanghaiTech University

Poll Question:

What is the computational cost of one iteration of gradient descent for logistic regression?

A. $O(1)$ **(TOXIC)** B. $O(N)$ C. $O(D)$ D. $O(ND)$

- Input: training dataset $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}^{(t)}\right) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}^{(i)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)}\right)$$

   a. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}^{(t)}\right)$
   b. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Gradient Descent

- Input: training dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$O(ND) \left\{ \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}^{(i)} \left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)} \right) \right.$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

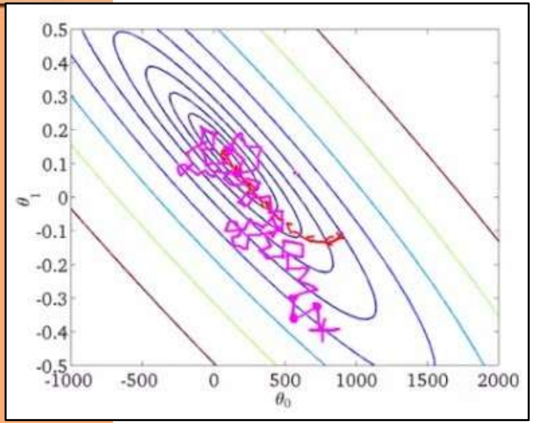
Stochastic Gradient Descent (SGD)

上海科技大学 ShanghaiTech University

Recall...

Algorithm 1 Stochastic Gradient Descent (SGD)

> **procedure** SGD($\mathcal{D}, \boldsymbol{\theta}^{(0)}$)
> $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
>   **while** not converged **do**
>   **for** $i \in$ shuffle($\{1, 2, \ldots, N\}$) **do**
> $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
>   **return** $\boldsymbol{\theta}$



We can also apply SGD to solve the MCLE problem for Logistic Regression.

We need a per-example objective:

$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} J^{(i)}(\boldsymbol{\theta}) \qquad\qquad J^{(i)}(\boldsymbol{\theta}) = -\log p(y^{(i)}|x^{(i)}, \boldsymbol{\theta})$$

Stochastic Gradient Descent (SGD)

- Input: training dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Randomly sample a data point from $\mathcal{D}$, $(\boldsymbol{x}^{(i)}, y^{(i)})$
   b. Compute the pointwise gradient:
   $$\nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta}^{(t)}) = \boldsymbol{x}^{(i)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)}\right)$$
   c. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta}^{(t)})$
   d. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

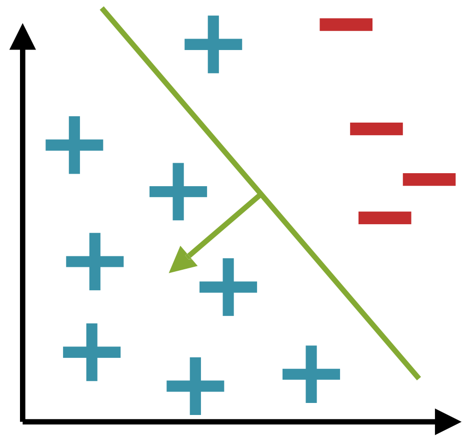
Logistic Regression vs. Perceptron

上海科技大学 ShanghaiTech University

Poll Question:

True or False: Just like Perceptron, **one step** (i.e. iteration) **of SGD for Logistic Regression** will result in a change to the parameters **only** if the current example is **incorrectly** classified.

Answer:



Bayes Optimal Classifier

上海科技大学
ShanghaiTech University

Functio

Previous
was ge
functio

Our goa
best ap

Suppose you knew the distribution p*(y | **x**) or had a good approximation to it.

Question:

How would you design a function y = h(**x**) to predict a single label?

Answer:

You'd use the *Bayes optimal classifier*!

Probabilistic Learning

Today, we assume that our output is **sampled** from a conditional **probability distribution:**

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$

$$y^{(i)} \sim p^*(\cdot|\mathbf{x}^{(i)})$$

Our goal is to learn a probability distribution p(y|**x**) that best approximates p*(y|**x**)

Bayes Optimal Classifier

上海科技大学 ShanghaiTech University

Bayes Optimal classifier:

The best possible classifier for a given data distribution $P^*(y|x)$

$$\hat{y} = h(x) = \begin{cases} 1 & if\ P^*(y=1|x) \geq \alpha \\ 0 & otherwise \end{cases}$$

- For 0/1 loss:

$$l(y, \hat{y}) = \mathbb{I}(y \neq \hat{y})$$

$$\alpha = 0.5$$

Probabilistic Learning

Today, we assume that our output is **sampled** from a conditional **probability distribution:**

$$\mathbf{x}^{(i)} \sim p^*(\cdot)$$

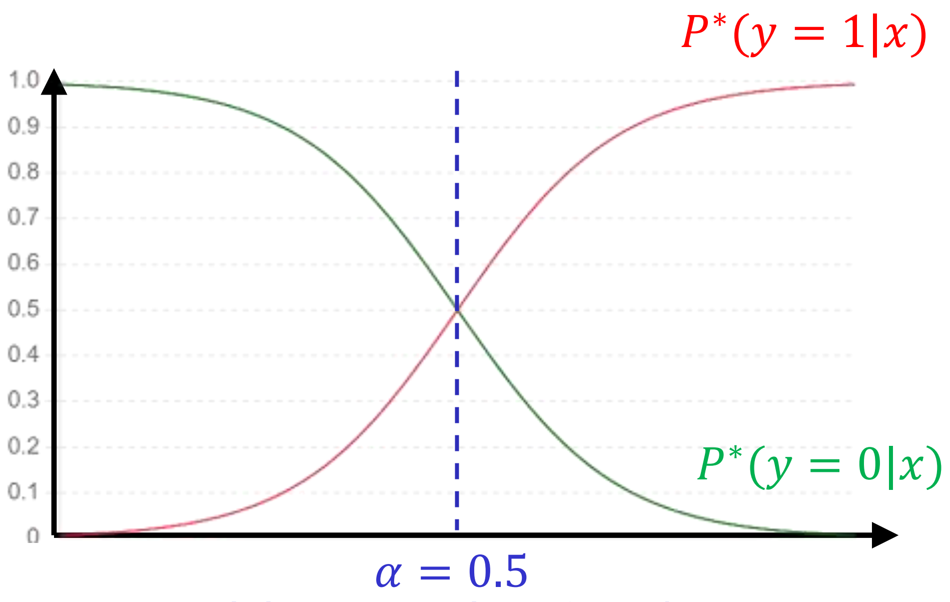
$$y^{(i)} \sim p^*(\cdot|\mathbf{x}^{(i)})$$

Our goal is to learn a probability distribution p(y|**x**) that best approximates p*(y|**x**)

Bayes Optimal Classifier

上海科技大学 ShanghaiTech University

Suppose you have an **oracle** that knows the data generating distribution, p*(y|x).
Q: What is the optimal classifier in this setting?
A: The Bayes optimal classifier! This is the best classifier for the distribution p* and the loss function.



Decision Boundary for 0/1 loss

- Bayes Decision Rule:

$$\hat{y} = h(x) = \begin{cases} 1 & if\ P^*(y=1|x) \geq \alpha \\ 0 & otherwise \end{cases}$$

- For 0/1 loss, $l(y,\hat{y}) = \mathbb{I}(y \neq \hat{y})$ $\quad \alpha = 0.5$
- For asymmetric loss:

$$l(y,\hat{y}) = \begin{cases} 1{,}000{,}000 & if\, y \neq \hat{y}\ and\ y = 1 \text{ (False Negative)} \\ 1{,}000 & if\, y \neq \hat{y}\ and\ y = 0 \text{ (False Positive)} \\ 0 & otherwise \end{cases}$$

$$\alpha = 0.0001$$

Definition: The **reducible error** is the expected loss of a hypothesis h(x) that could be reduced if we knew p*(y|x) and picked the optimal h(x) for that p*.

Definition: The **irreducible error** is the expected loss of a hypothesis h(x) that could **not** be reduced if we knew p*(y|x) and picked the optimal h(x) for that p*.

Comparison Between Different Losses

上海科技大学
ShanghaiTech University

Aspect	0/1 Loss	Asymmetric Loss	Log-Loss (Cross-Entropy)
Purpose	Measures classification accuracy	Adjusts decision boundary penalties	Optimizes logistic regression
Mathematically Differentiable?	❌ No	❌ No	✅ Yes
Used for?	**Evaluation**	**Adjusting decision boundary**	**Optimization**
Training Logistic Regression?	❌ No	❌ No	✅ Yes
Alternative to Incorporate in Logistic Regression?	Adjust decision threshold α	Use weighted log-loss	Directly used

Model Performance Metrics

上海科技大学 ShanghaiTech University

Model performance metrics are measurements used to evaluate the effective and efficiency of a predictive model or machine learning algorithm.

- ✓Accuracy
- ✓Precision
- ✓Recall (Sensitivity)
- ✓F1-Score
- ✓Confusion Matrix
- ✓ROC Curve and AUC

TP TN FP FN

Confusion Matrix

	Actual	
Predicted	**Positive**	**Negative**
Positive	**True Positive (TP)**	**False Positive (FP)**
Negative	**False Negative (FN)**	**True Negative (TN)**

Accuracy = (TP + TN)/(TP + TN + FP + FN)

Accuracy is useful for evaluating classification model when classes are balanced (binary or multi-class classification).

When classes in the dataset are highly imbalanced, meaning there is a significant disparity in the number of instances between classes, accuracy can be misleading.

A model may achieve high accuracy by simply predicting the majority class for every instance, ignoring the minority class entirely.

TP TN FP FN

Confusion Matrix

	Actual		
Predicted		**Positive**	**Negative**
	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

Precision = TP / (TP + FP)

Precision is particularly useful in scenarios where the cost of false positives is high.

The importance of precision is in music or video recommendation systems, etc., where wrong results could lead to customer churn, and this could be harmful to the business.

TP TN FP FN

上海科技大学
ShanghaiTech University

Confusion Matrix

		Actual	
Predicted		**Positive**	**Negative**
	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

Recall = TP / (TP + FN)

Recall is particularly useful in scenarios where capturing all positive instances is crucial, even if it means accepting a higher rate of false positives.

In medical diagnosis, missing a positive instance (false negative) can have severe consequences for the patent's health or even lead to loss of life. High recall ensures that the model identifies as many positive cases as possible, reducing the likelihood of missing critical diagnoses.

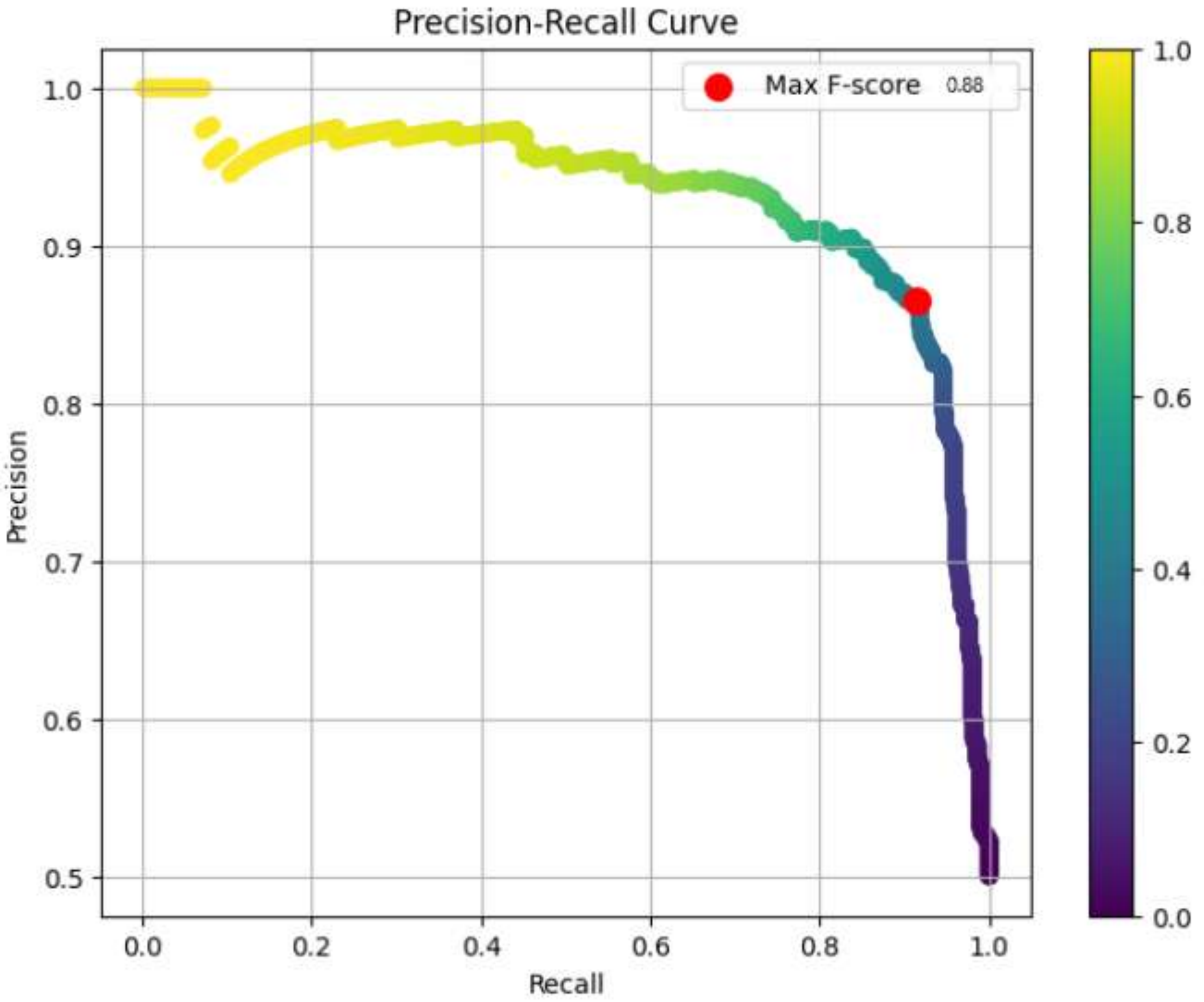
TP TN FP FN

上海科技大学 ShanghaiTech University

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 = 2 / (1/Precision + 1/Recall)



Mini-Batch SGD

- **Gradient Descent:**
  Compute true gradient exactly from all N examples
- **Stochastic Gradient Descent (SGD):**
  Approximate true gradient by the gradient of one randomly chosen example
- **Mini-Batch SGD:**
  Approximate true gradient by the average gradient of K randomly chosen examples

while not converged: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \mathbf{g}$

Three variants of first-order optimization:

Gradient Descent: $\mathbf{g} = \nabla J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \nabla J^{(i)}(\boldsymbol{\theta})$

SGD: $\mathbf{g} = \nabla J^{(i)}(\boldsymbol{\theta})$ where i sampled uniformly

Mini-batch SGD: $\mathbf{g} = \frac{1}{S} \sum_{s=1}^{S} \nabla J^{(i_s)}(\boldsymbol{\theta})$ where i_s sampled uniformly $\forall s$

Logistic Regression Learning Objectives

You should be able to…

- Apply the principle of maximum likelihood estimation (MLE) to learn the parameters of a probabilistic model
- Given a discriminative probabilistic model, derive the conditional log-likelihood, its gradient, and the corresponding Bayes Classifier
- Explain the practical reasons why we work with the log of the likelihood
- Implement logistic regression for binary (and multiclass) classification
- Prove that the decision boundary of binary logistic regression is linear