Regularization & Multiclass Classification

Alexander G. Ororbia II

Introduction to Machine Learning

CSCI-335

3/23/2026

Special Thanks: Sargur N. Srihari, Andrew Ng, Carlos Guestrin

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- **Regularization**
- Multi-class classification

Ways to Address Overfitting:

Options:

1. Reduce number of features:
   - Manually select which features to keep
   - Model selection algorithm
2. Regularization:
   - Keep all the features, but reduce magnitude/values of parameters θ_j
   - Works well when we have a lot of features, each of which contributes a bit to predicting y

In regularized linear regression, we choose θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (perhaps too large for our problem, say $\lambda = 10^{10}$)?

- Algorithm works fine; setting λ to be very large cannot hurt it
- Algorithm fails to eliminate overfitting.
- Algorithm results in underfitting. (Fails to fit even training data well).
- Gradient descent will fail to converge.

In regularized linear regression, we choose θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

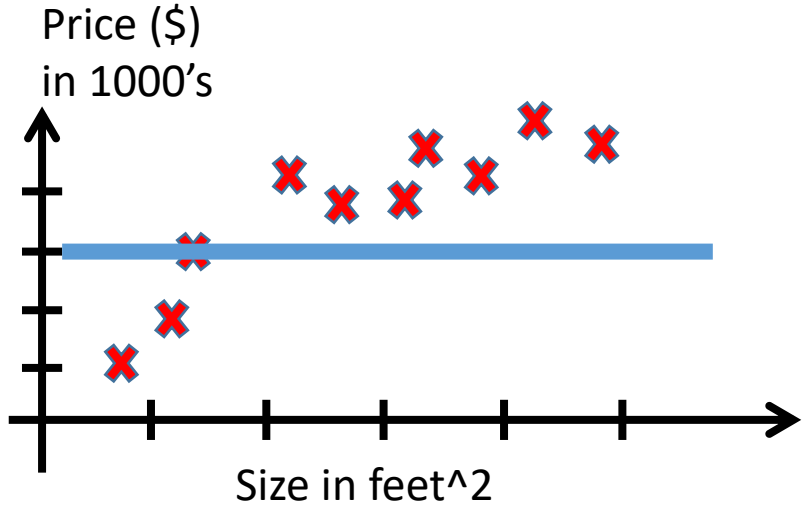
What if λ is set to an extremely large value (perhaps too large for our problem, say $\lambda = 10^{10}$)?

- Algorithm works fine; setting λ to be very large cannot hurt it
- Algorithm fails to eliminate overfitting.
- Algorithm results in underfitting. (Fails to fit even training data well).
- Gradient descent will fail to converge.

Answer:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (say $\lambda = 10^{10}$)?

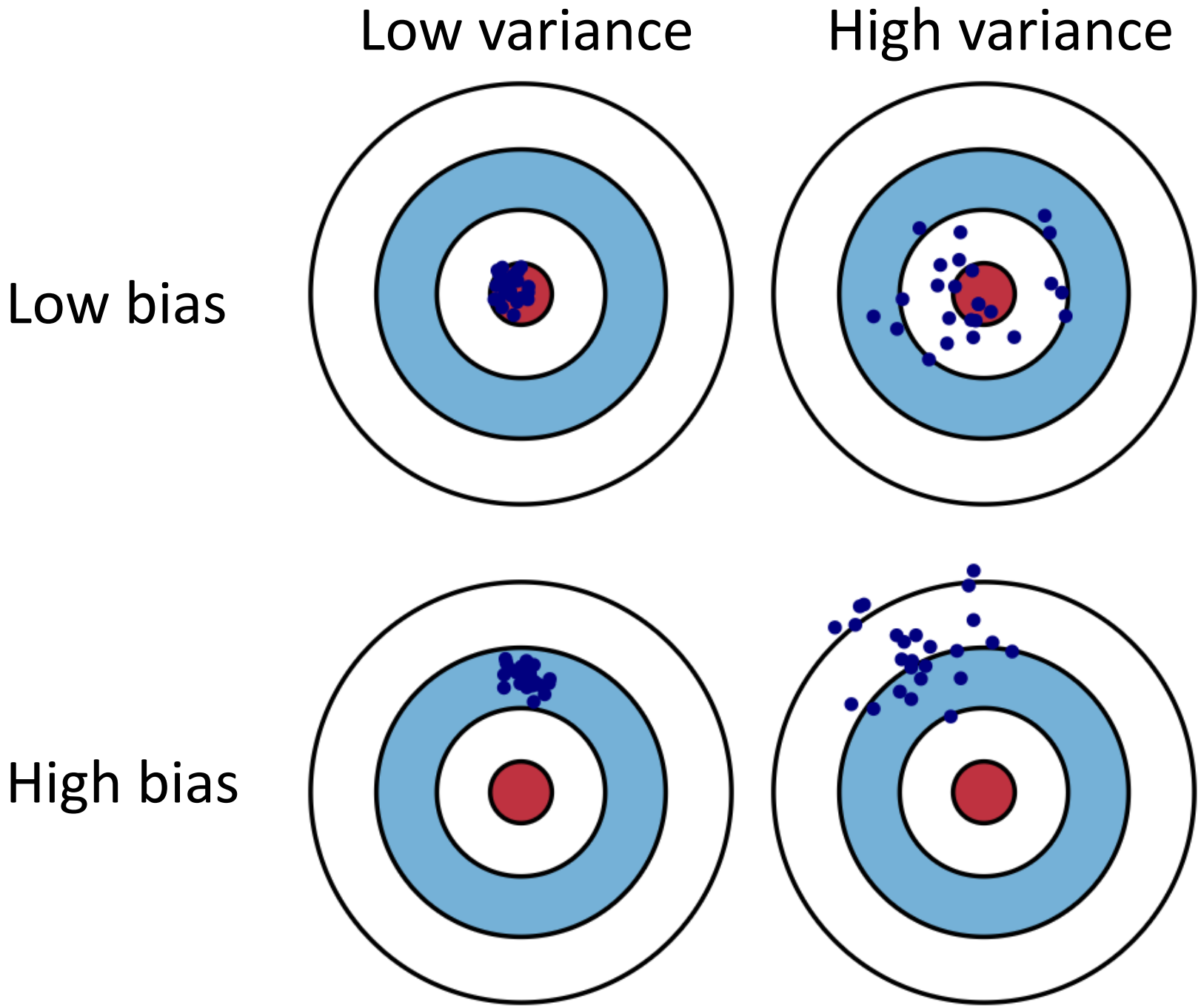


$$h_\theta(x) = \theta_0 + \cancel{\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n} = \theta^\top x$$

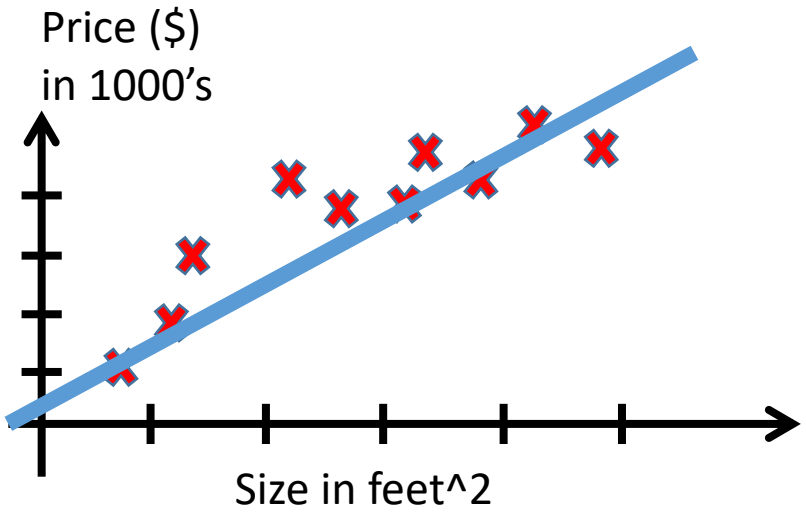
Bias-Variance Tradeoff (Again!)

- **Bias:** difference between
  what you expect to learn and **truth**
  - Measures how well you expect to represent true solution
  - Decreases with more complex model

- **Variance:** difference between
  what you expect to learn and
  what you learn from a particular dataset
  - Measures how *sensitive* learner is to specific dataset
  - Increases with more complex model


Low variance
High variance
Low bias
High bias

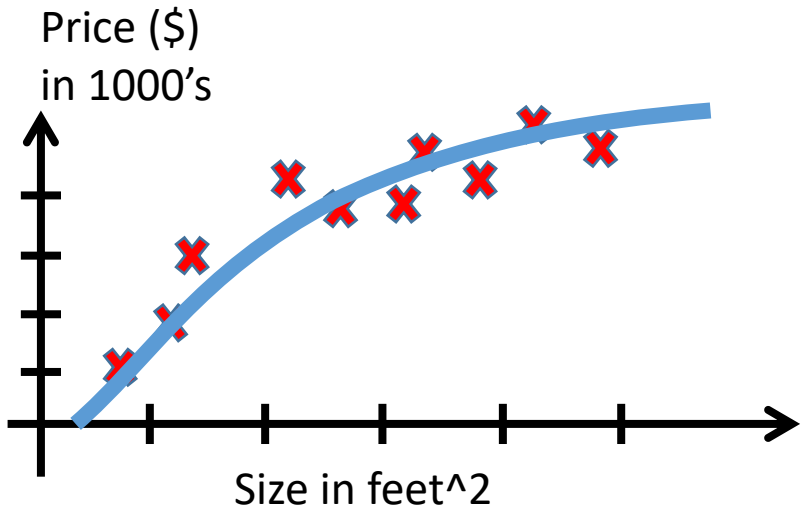
Example: Polynomial Regression (Overfitting)



$h_\theta(x) = \theta_0 + \theta_1 x$

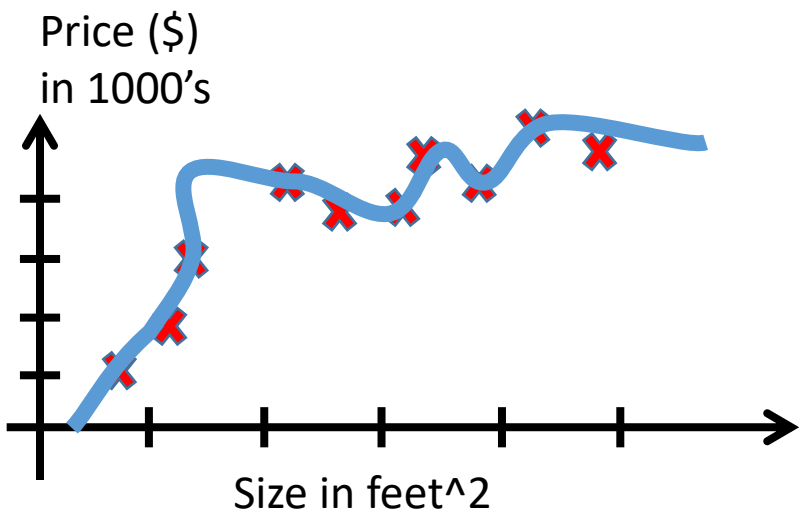
Underfitting

High bias



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$

Just right (the "sweet spot")

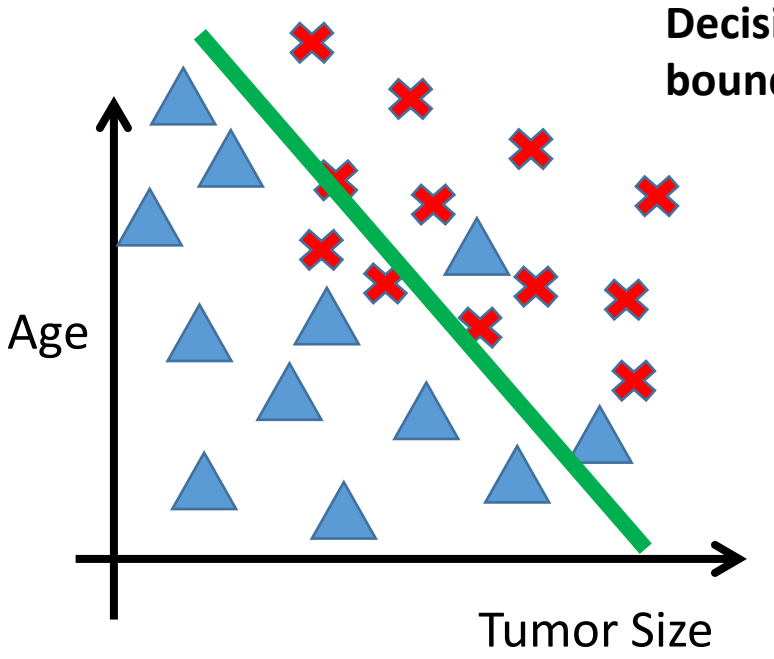


$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \cdots$

Overfitting

High variance

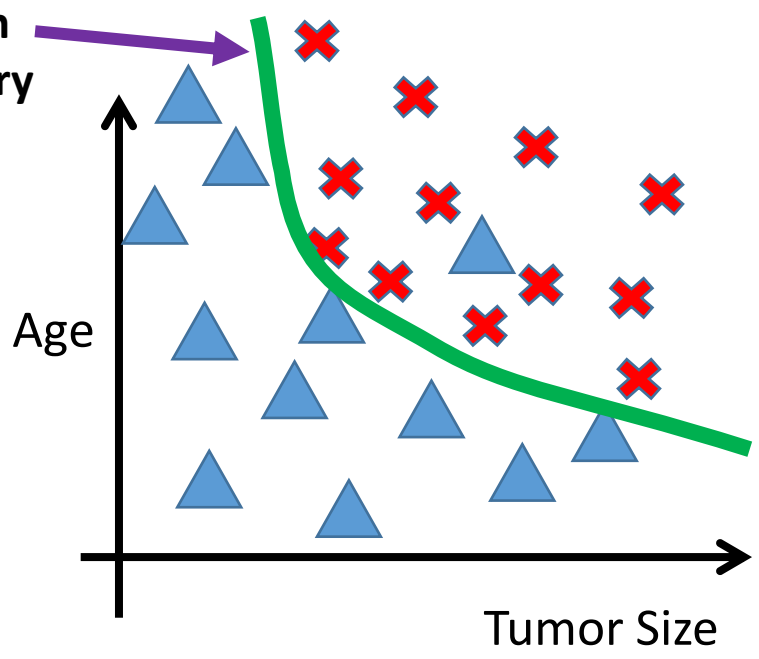
Example: Logistic Regression (Overfitting)



$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2)$

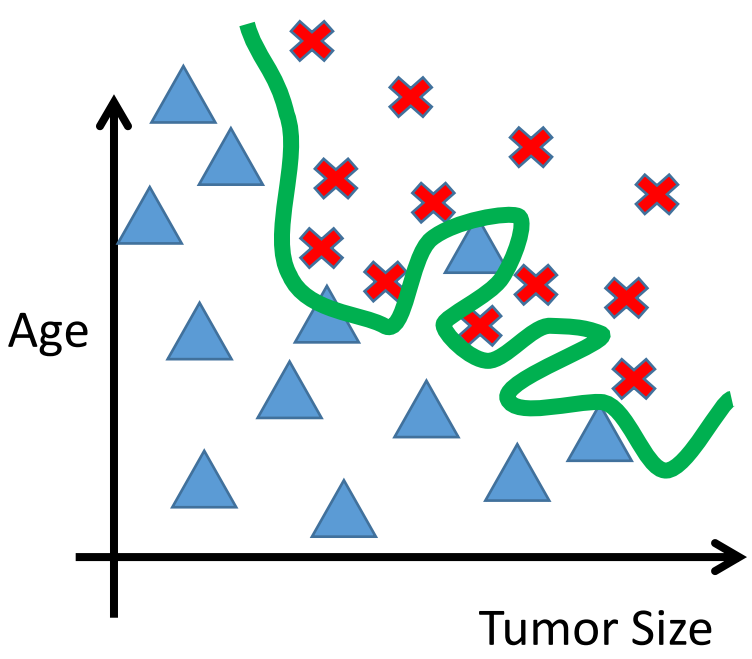
Underfitting

High bias



$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

Just right
(the "sweet spot")



$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \theta_7 x_1 x_2^3 + \cdots)$

Overfitting

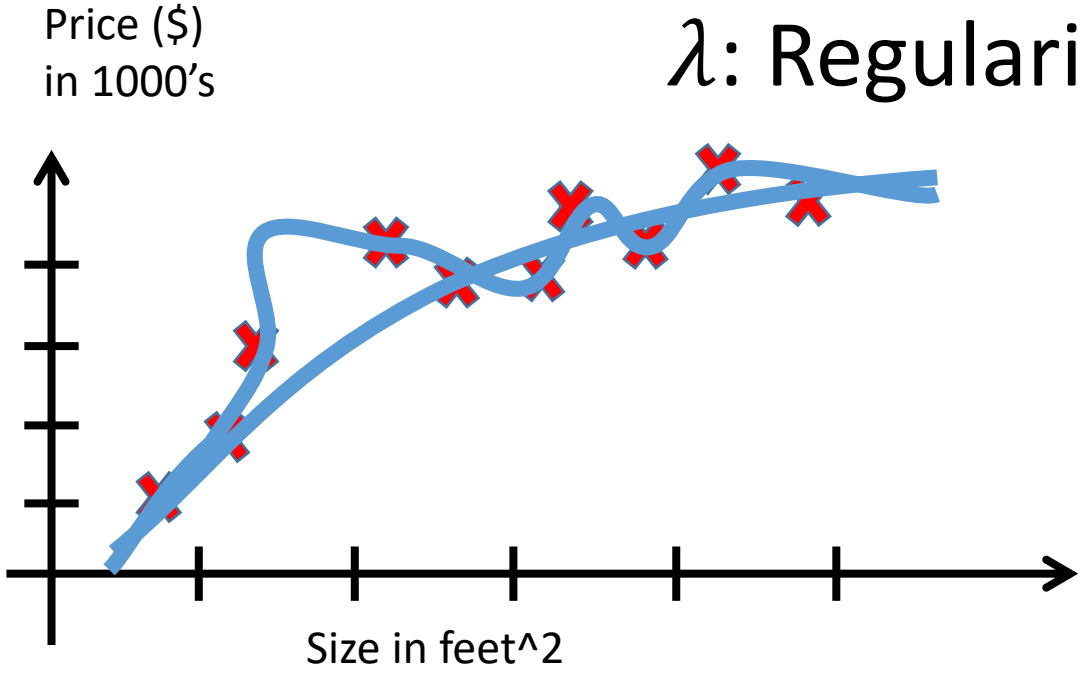
High variance

Regularization

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n}\theta_j^2\right]$$

λ: Regularization parameter

$\min_\theta J(\theta)$



Regularized Linear Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

n: Number of features

θ_0 (bias) is not penalized

Gradient Descent (Previously)

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \qquad (j = 0)$$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)} \right] \qquad (j = 1, 2, 3, \cdots, n)$$

}

Gradient Descent (Regularized)

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)$$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)} + \lambda \theta_j \right]$$

}

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)} \right]$$

Works like weight or parameter "decay"

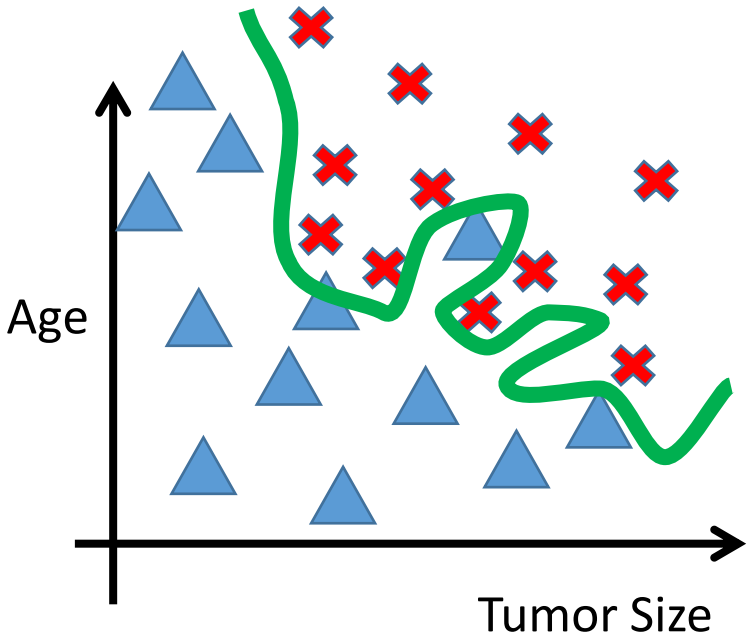
$|\theta|_1$: Lasso Regularization

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}|\theta_j|\right]$$

LASSO: Least Absolute Shrinkage and Selection Operator

Also known as L1 Penalty – remember distance functions / L-p norms?

Regularized Logistic Regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \theta_7 x_1 x_2^3 + \cdots)$$

- Cost function:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right) + \frac{\lambda}{2}\sum_{j=1}^{n} \theta_j^2\right]$$

Gradient Descent (Regularized)

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)$$

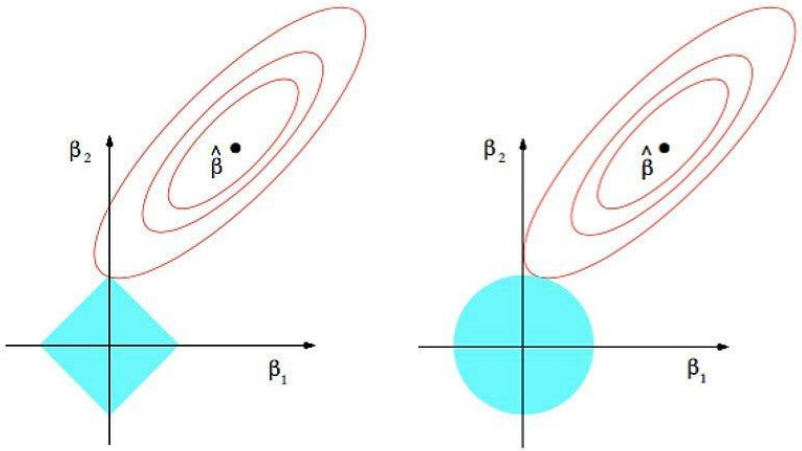
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}$$

$$\theta_j := \theta_j - \alpha \underbrace{\left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)} - \lambda \theta_j \right]}_{\frac{\partial}{\partial \theta_j} J(\theta)}$$

}

General Regularization Terminology

Regularization function	Name	Solver
$\|\theta\|_2^2 = \sum_{j=1}^{n} \theta_j^2$	Tikhonov regularization Ridge regression	Closed form
$\big\|\|\theta\|\big\|_1 = \sum_{j=1}^{n} \|\theta_j\|$	LASSO regression	Proximal gradient descent, least angle regression
$\alpha\big\|\|\theta\|\big\|_1 + (1-\alpha)\ \|\theta\|_2^2$	Elastic net regularization	Proximal gradient descent



How about MAP?

- **Maximum (conditional) likelihood estimate (MCLE)**

$$\theta_{\mathrm{MCLE}} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{m} P_\theta\left(y^{(i)}|x^{(i)}\right)$$

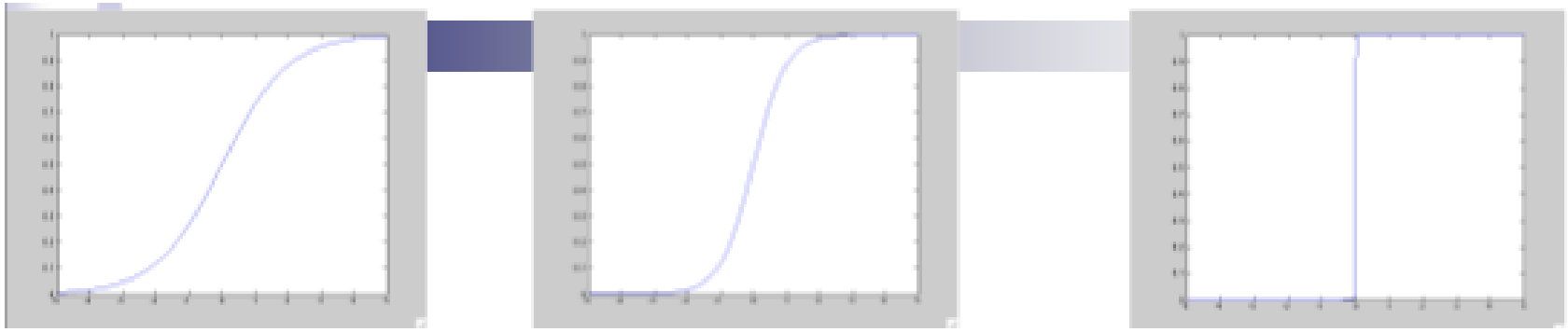
- **Maximum (conditional) a posterior estimate (MCAP)**

$$\theta_{\mathrm{MCAP}} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{m} P_\theta\left(y^{(i)}|x^{(i)}\right)P(\theta)$$

Prior $P(\theta)$

- Common choice of $P(\theta)$:
  - Normal distribution, zero mean, identity covariance
  - "Pushes" parameters towards zeros
- Corresponds to **Regularization**
  - Helps avoid very large weights and overfitting

Notice the "saturation" effect on logistic link1



$$\frac{1}{1+e^{-x}} \qquad \frac{1}{1+e^{-2x}} \qquad \frac{1}{1+e^{-100x}}$$

MLE versus MAP

- **Maximum (conditional) likelihood estimate (MCLE)**

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)}$$

- **Maximum (conditional) a posterior estimate (MCAP)**

$$\theta_j := \theta_j - \alpha\lambda\theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)}$$

MLE vs. MAP

- **Maximum (conditional) likelihood estimate (MCLE)**

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)}$$

- **Maximum (conditional) a posterior estimate (MCAP)**

$$\theta_j := \theta_j - \alpha\lambda\theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)}$$

Logistic Regression

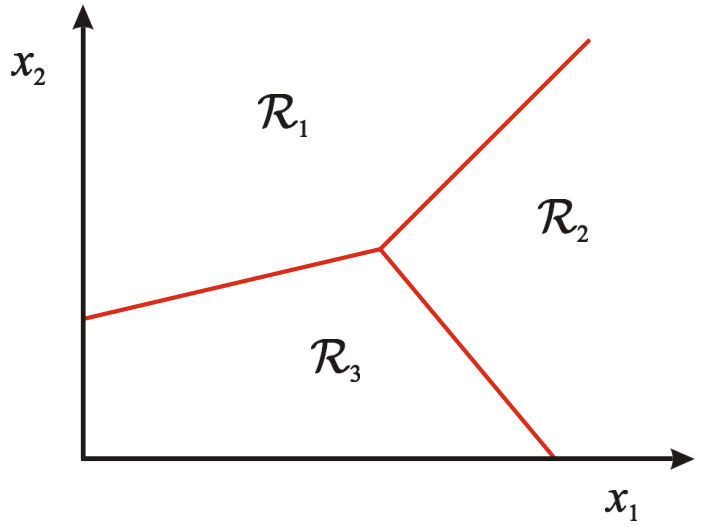
- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- Regularization
- **Multi-class classification**

Multi-Class Classification

- ***What if we have more than two classes/categories/conditions?***
- Example scenarios:
  - Email foldering/tagging: Work, Friends, Family, Hobby
  - Medical diagrams: Not ill, Cold, Flu
  - Weather: Sunny, Cloudy, Rain, Snow

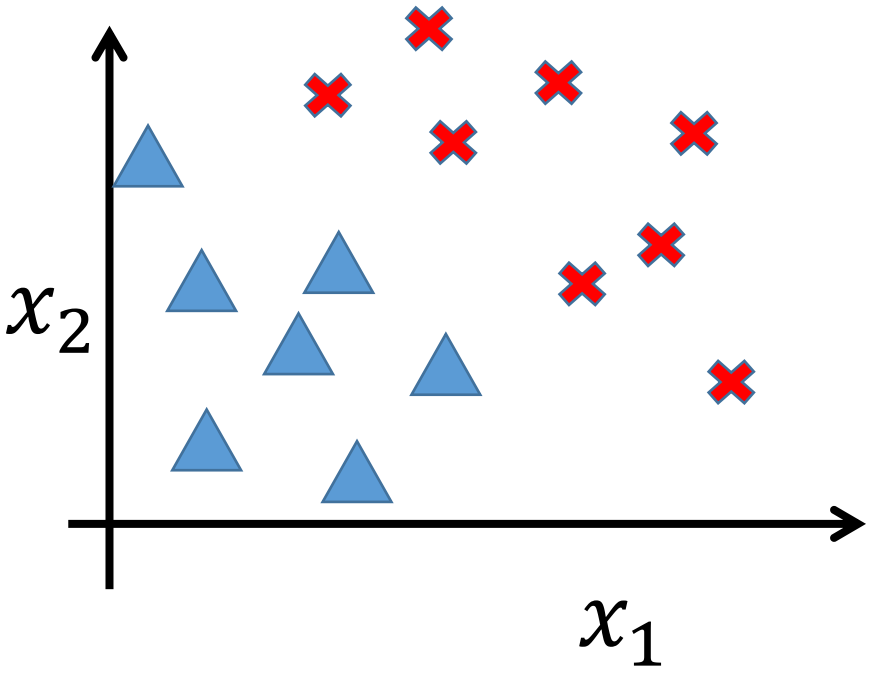
Classification / Decision-Making

- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*

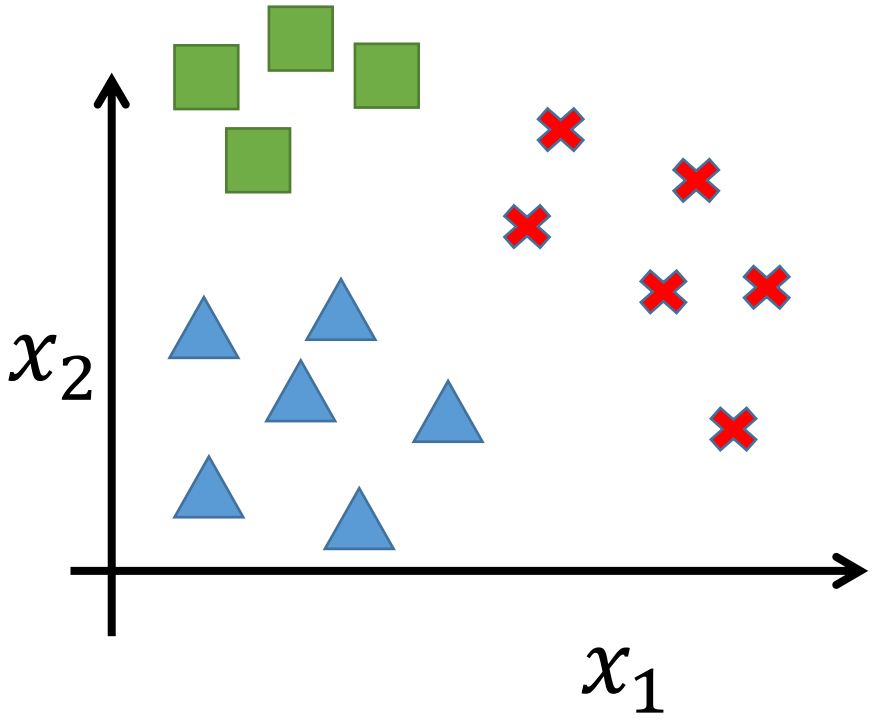


Slide credit: L. Lazebnik

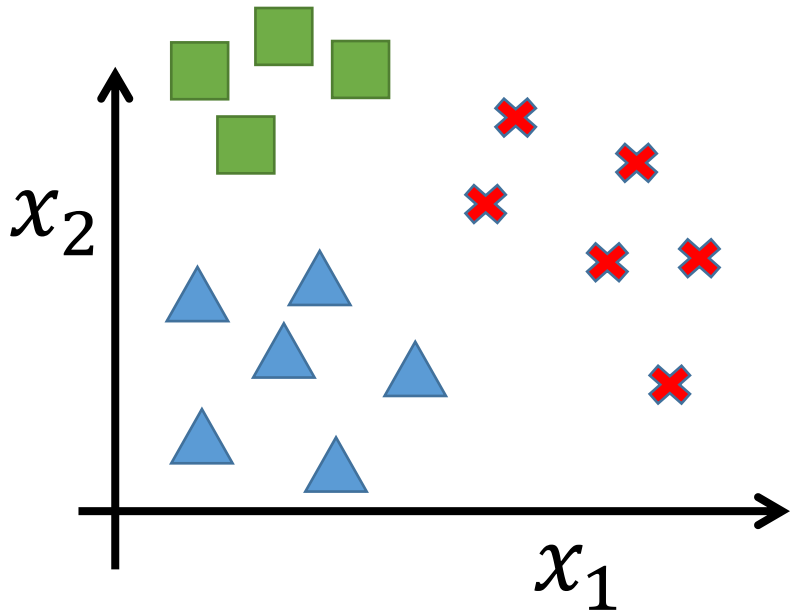
Binary classification



Multiclass classification



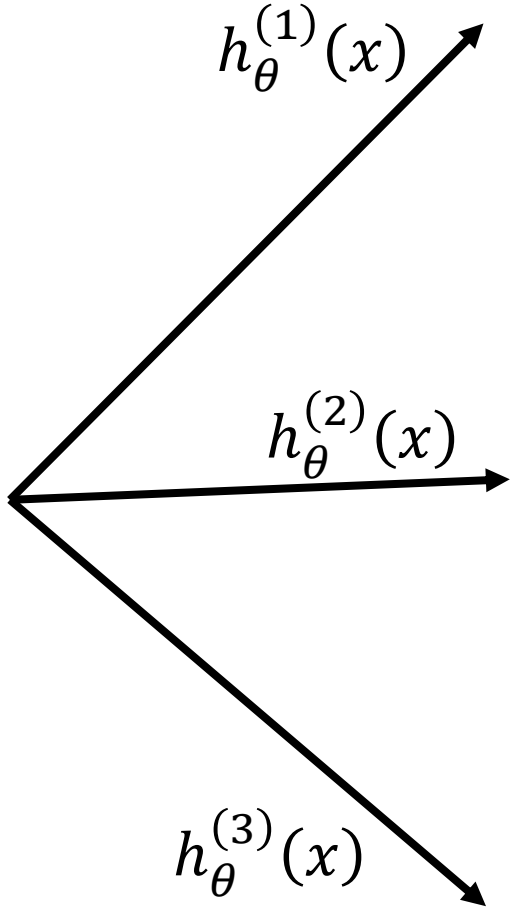
One-vs-All (One-vs-Rest)

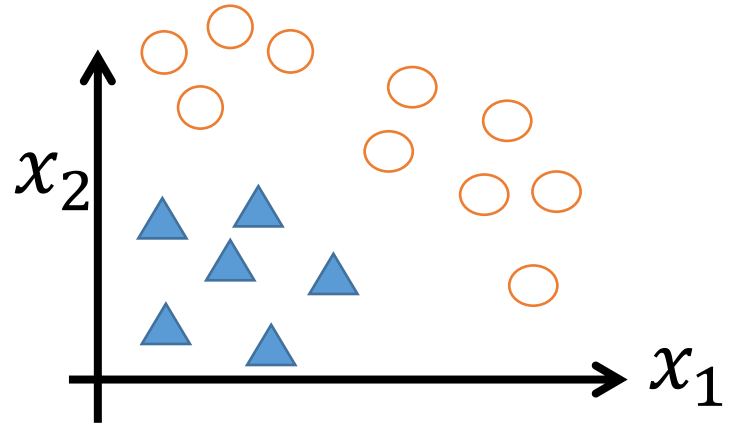


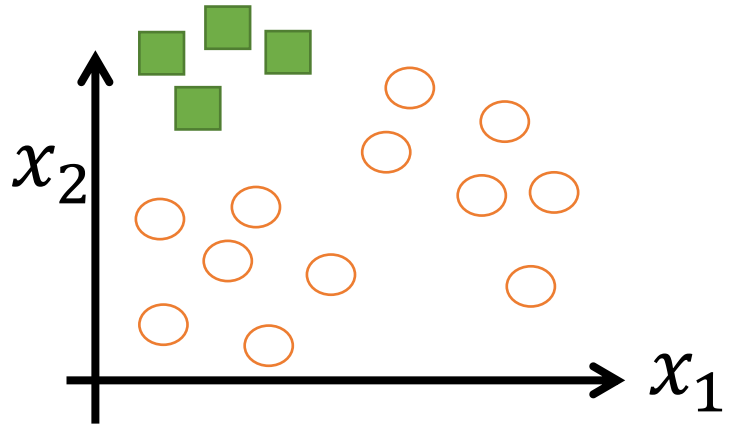
Class 1:

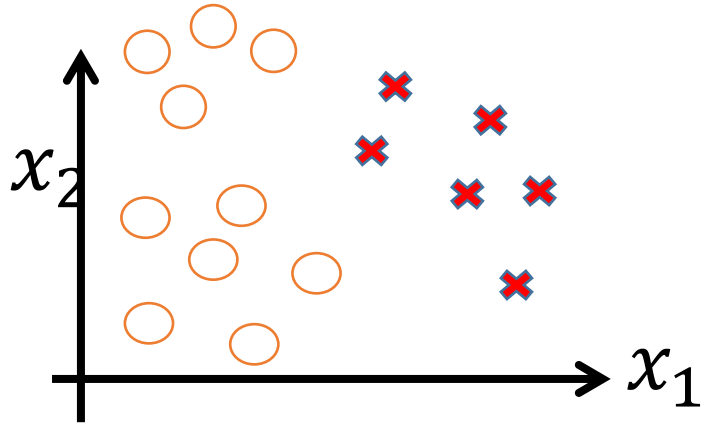
Class 2:

Class 3:









$$h_\theta^{(i)}(x) = P(y = i|x;\theta) \quad (i = 1,2,3)$$

One-vs-All

- Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$

- Given a new input x, pick the class i that maximizes

$$\max_{\mathrm{i}} h_\theta^{(i)}(x)$$

One-vs-All

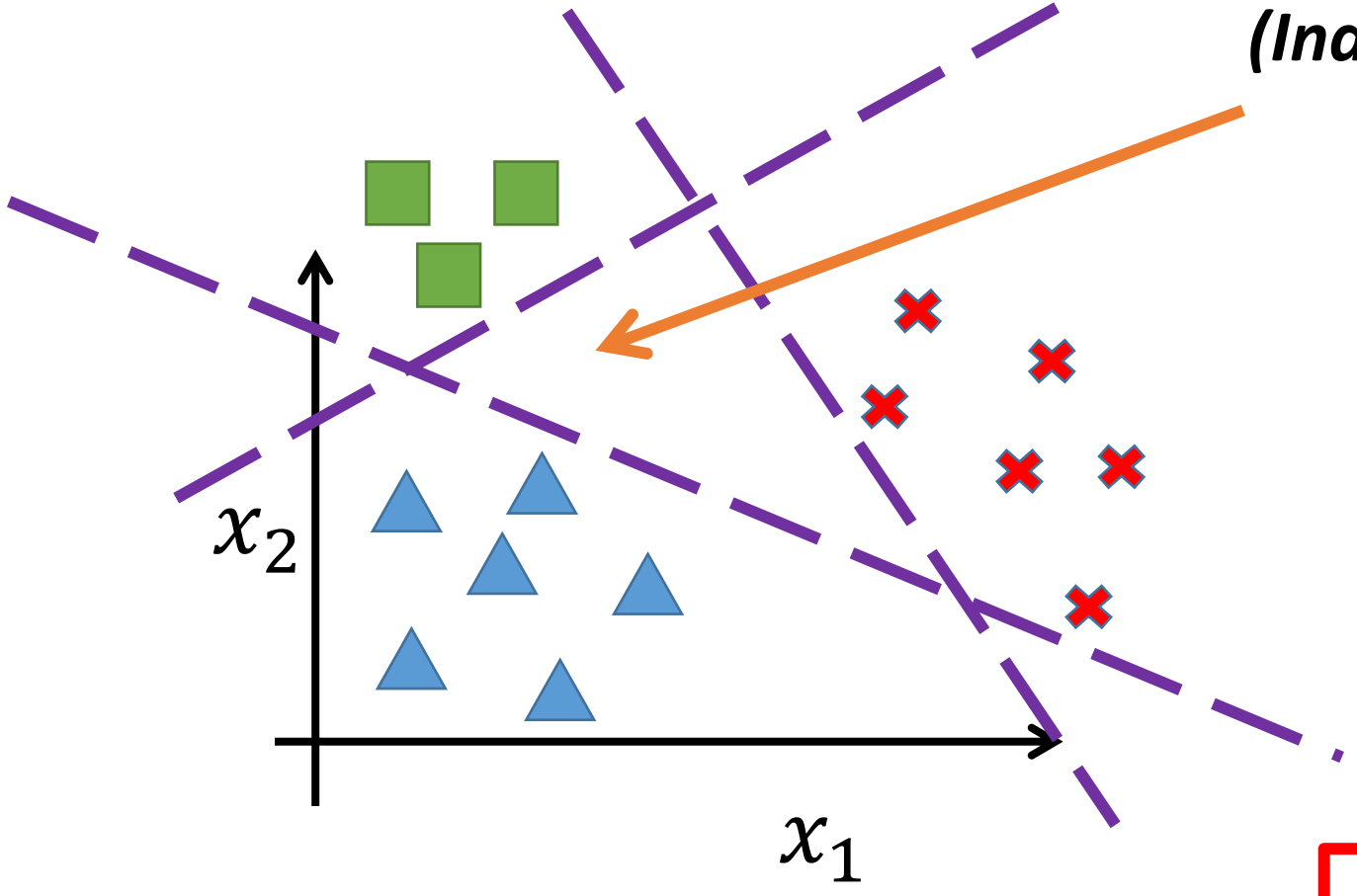
- Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$

- Given a new input x, pick the class i that maximizes

$$\max_{\mathrm{i}} h_\theta^{(i)}(x)$$

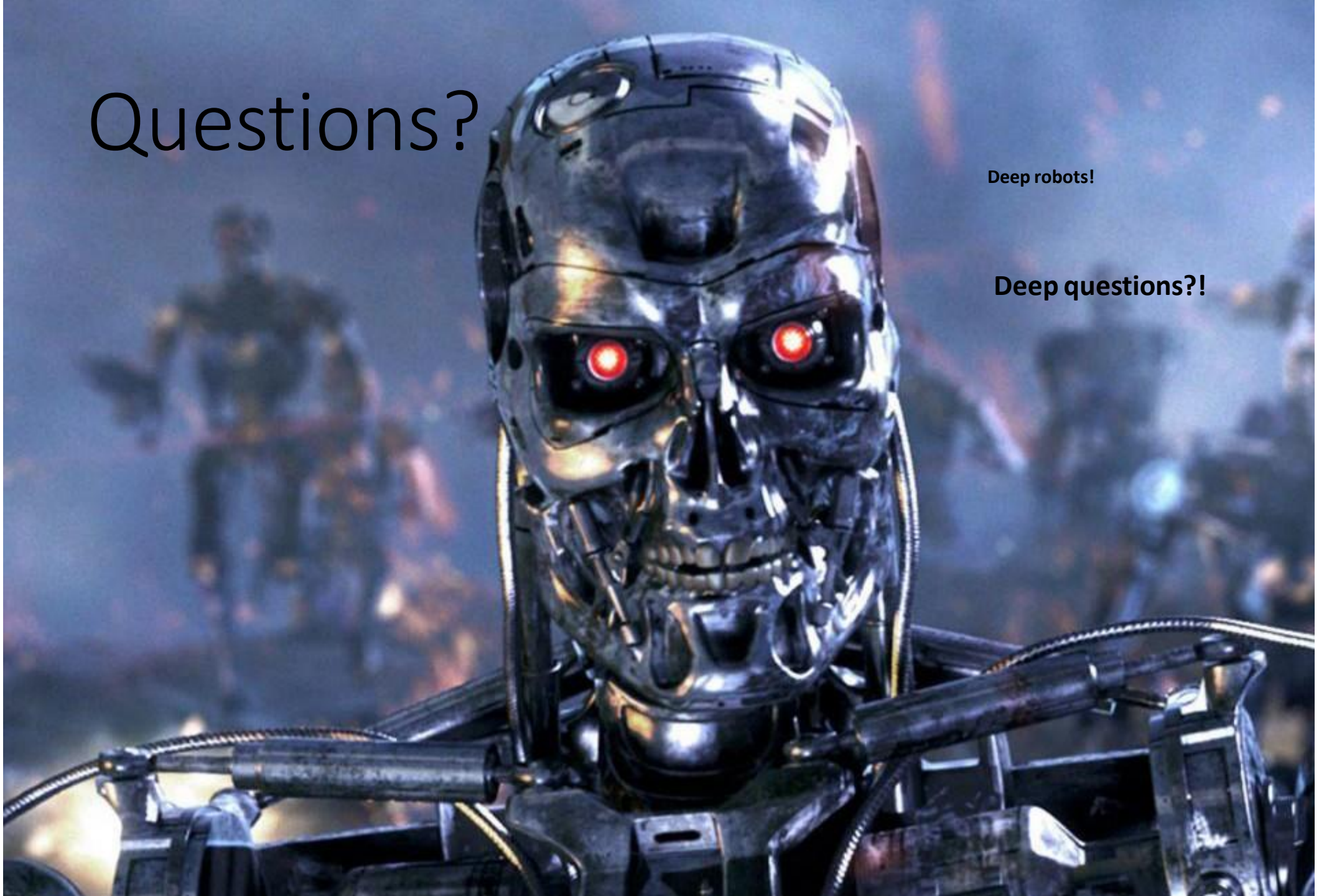
Problem:
Indeterminacy

What class does this zone belong to? (Indeterminate region)



Problem:
Indeterminacy

Questions?



Deep robots!

Deep questions?!