

AI504: Programming for Artificial Intelligence

Week 2: Basic Machine Learning

Edward Choi

Grad School of AI

edwardchoi@kaist.ac.kr

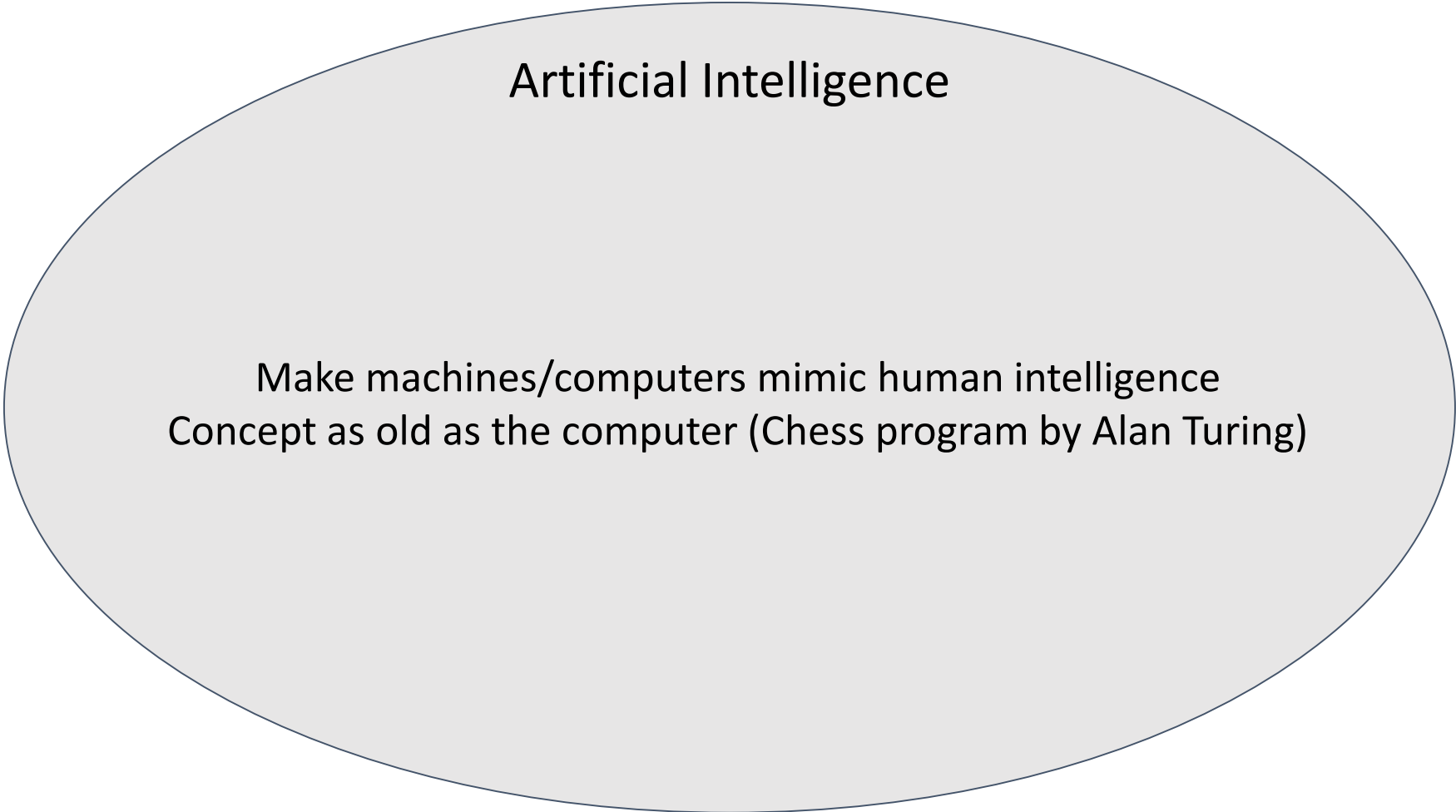
WARNING!!!

- There are 600 students in this class
 - **300 will probably FAIL**
- No exceptions will be made
 - I will lose my scholarship
 - I won't be able to graduate
 - I was busy with my paper
 - It's just a one-line mistake
 - I forgot to change the file name
 - ...
- It is your **RESPONSIBILITY** to
 - Check emails and KLMS
 - Follow the project spec

AI Teaching Assistant

- Can LLMs handle course-related Q&A as effectively as human TAs?
 - Continued effort from last year
- We will collect the following:
 - Student profiles
 - Conversation logs
 - Use the AI TA for this course only
 - Satisfaction surveys
 - Interviews
- AI TA will launch today
- **Mandatory Survey** will start today → **Due date Sep. 12th 23:59pm**
- Failure to complete the survey → **Automatic F**

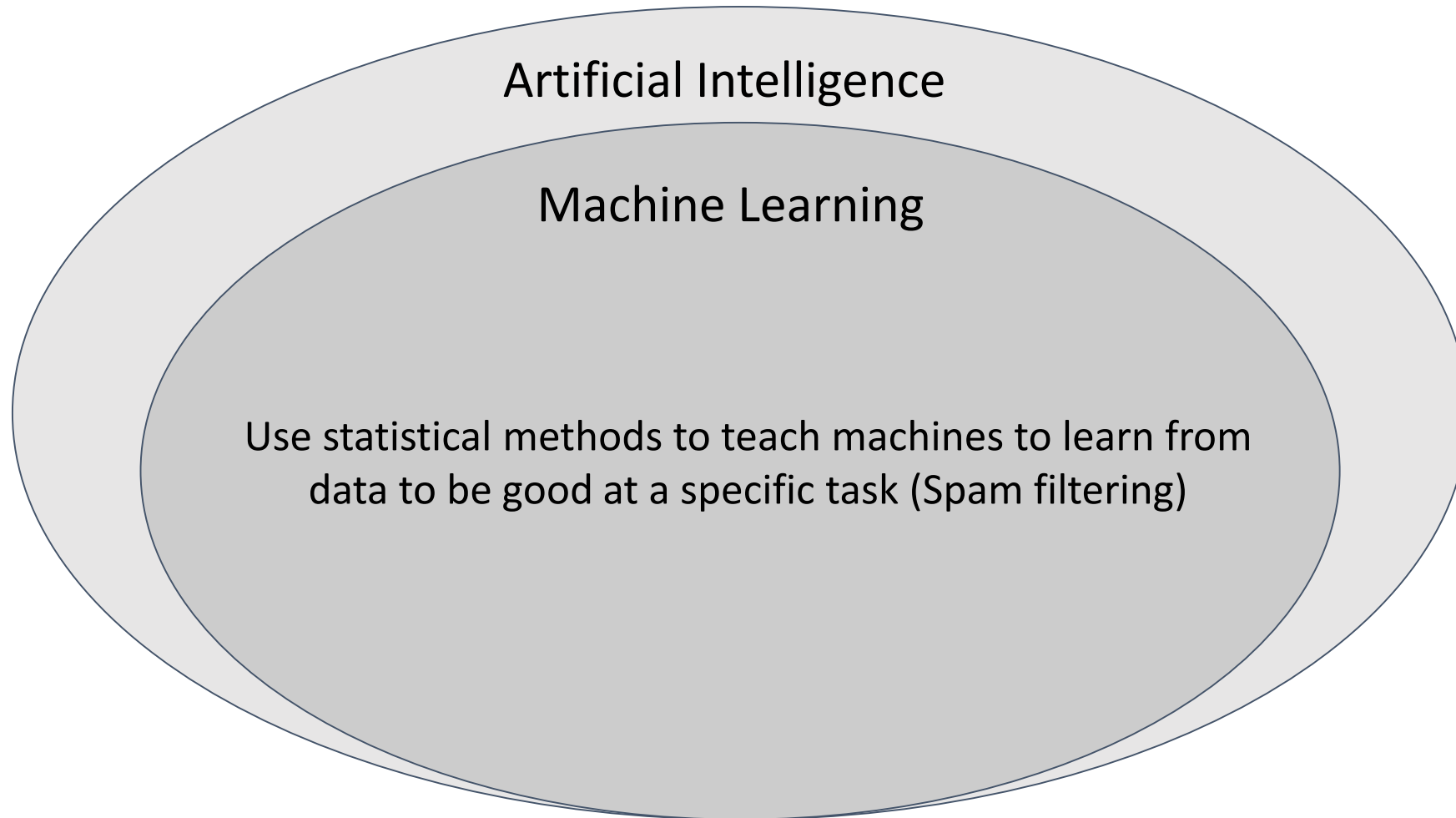
What is AI?



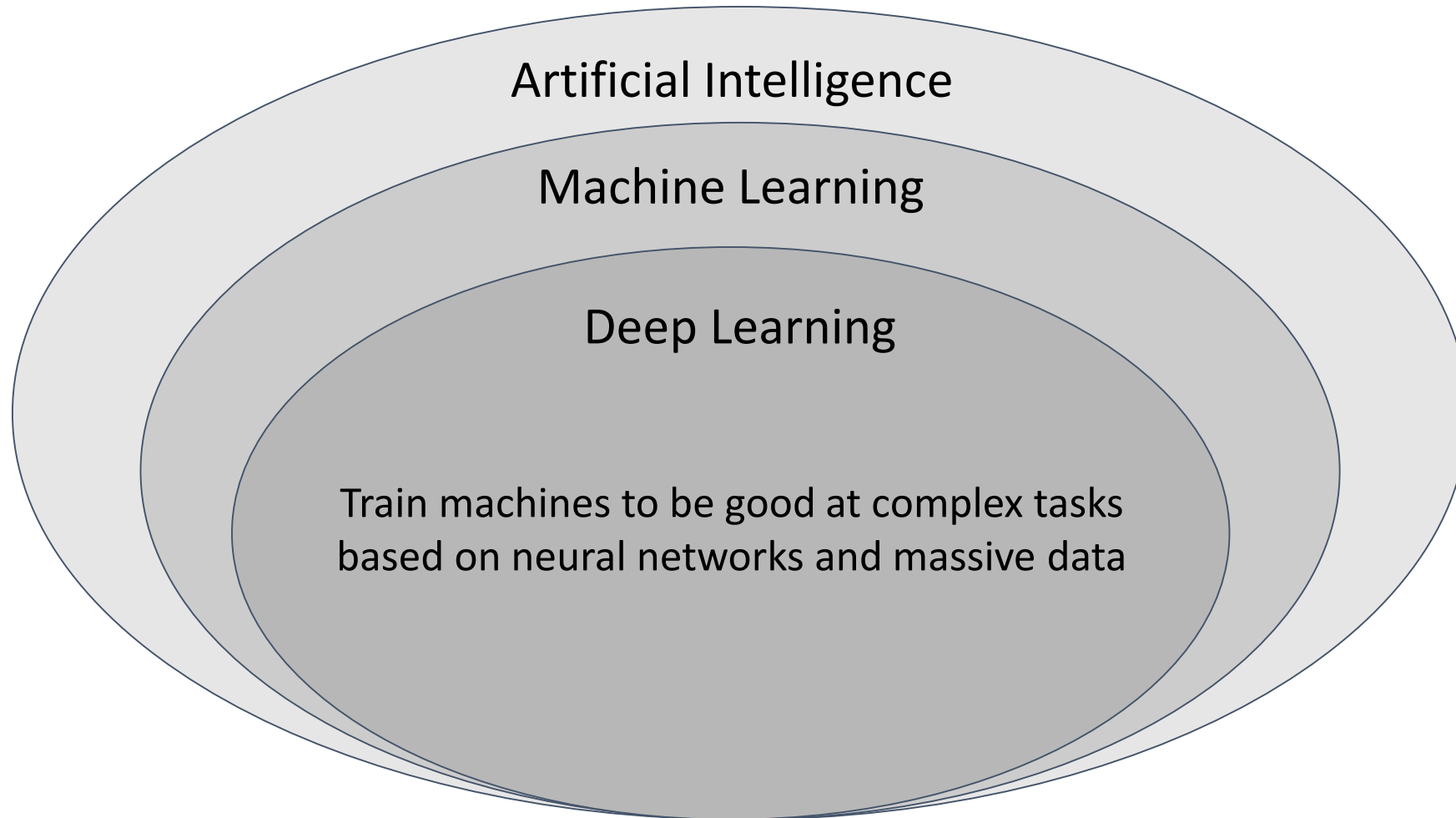
Artificial Intelligence

Make machines/computers mimic human intelligence
Concept as old as the computer (Chess program by Alan Turing)

What is Machine Learning?



What is Deep Learning?



Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
 - Learn a function that maps an input $\mathbf{x}$ to an output $\mathbf{y}$
 - Examples?
- Unsupervised Learning
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
 - Learn a function that maps an input $\mathbf{x}$ to an output $\mathbf{y}$
 - Examples
 - Image classification
 - French-English translation
 - Image captioning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
 - Learn a distribution/manifold function of data $\mathbf{X}$ (no label $\mathbf{y}$)
 - Examples?
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
 - Learn a distribution/manifold function of data $\mathbf{X}$ (no label $\mathbf{y}$)
 - Examples
 - Clustering
 - Low-rank matrix factorization
 - Kernel density estimation
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
 - Given an environment **E** and a set of actions **A**, learn a function that maximizes the long-term reward R .
 - Examples?

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
 - Given an environment **E** and a set of actions **A**, learn a function that maximizes the long-term reward **R**.
 - Examples
 - Go
 - Atari
 - Self-driving car

Machine Learning Categories

- Supervised Learning
 - Unsupervised Learning
 - Reinforcement Learning
- } Lines are a bit blurry with generative models and self-supervised learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

All practices in this course are either SL or UL

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called **statistical machine learning**!

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
 - You need to have a goal
 - Otherwise how are you going to train your model?
 - We call this goal “objective function”

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
 - You need to have a goal
 - Otherwise how are you going to train your model?
 - We call this goal “objective function”
 - e.g. Dog classification



Objective: Correctly classify Dog/No Dog

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
 - You need to have a goal
 - Otherwise how are you going to train your model?
 - We call this goal “objective function”
 - e.g. Dog classification



Objective: Minimize $loss(y, y')$

y : true class, y' : predicted class ($y' = f(x; \theta)$)

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
 - You need to have a goal
 - Otherwise how are you going to train your model?
 - We call this goal “objective function”
 - e.g. Dog classification



Objective: Minimize $-(y * \log(y') + (1-y) * \log(1-y'))$
 y : true class, y' : predicted class ($y' = f(x; \theta)$)

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
 - You need to have a goal
 - Otherwise how are you going to train your model?
 - We call this goal “objective function”
- You need to find θ that minimizes $loss(y, f(x; \theta))$
 - That's why it's called **optimization!**

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
 - Your machine needs to learn from data
 - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
 - You need to have a goal
 - Otherwise how are you going to train your model?
 - We call this goal “objective function”
- You need to find θ that minimizes $loss(y, f(x; \theta))$
 - There are other loss functions
 - Regression: Mean Squared Error (MSE) or Mean Absolute Error (MAE)

How to Optimize Your Model

- Find θ that minimizes $f(\theta) = \theta^2 - 2\theta + 1$
 - $\theta = 1 \rightarrow$ One global minimum
 - Why one global minimum?

How to Optimize Your Model

- Find θ that minimizes $f(\theta) = \theta^2 - 2\theta + 1$
 - $\theta = 1 \rightarrow$ One global minimum
 - Why one global minimum?
- Find θ that minimizes $loss(y, f(x; \theta))$
 - $loss(y, f(x; \theta))$ is a complex (not meaning imaginary), non-convex function
 - Many local minima, no analytical solution

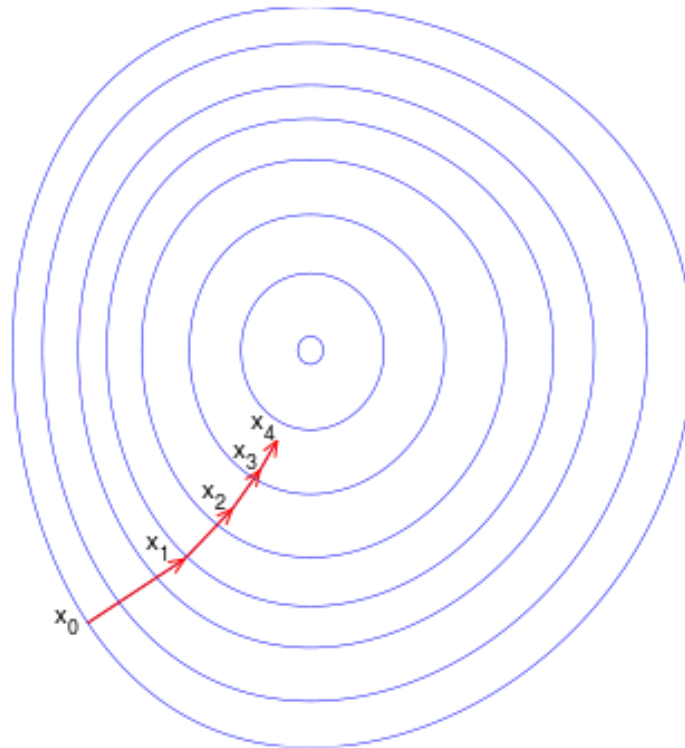
How to Optimize Your Model

- Find θ that minimizes $f(\theta) = \theta^2 - 2\theta + 1$
 - $\theta = 1 \rightarrow$ One global minimum
 - Why one global minimum?
- Find θ that minimizes $loss(y, f(x; \theta))$
 - $loss(y, f(x; \theta))$ is a complex (not meaning imaginary), non-convex function
 - Many local minima, no analytical solution
- Numerical method
 - Iteratively find a better θ until you are satisfied.

Gradient Descent

- Updating your parameters θ based on your training data X, Y

$$\theta_{k+1} = \theta_k - \eta \nabla_{\theta} \mathcal{L}(Y, f(X; \theta_k))$$



Stochastic Gradient Descent

- Updating your parameters θ based on a **subset** of your training data

$$\theta_{k+1} = \theta_k - \eta \nabla_{\theta} \mathcal{L}(\boxed{Y}, f(\boxed{X}; \theta_k))$$

Not your entire data.

A subset (i.e. minibatch) of the entire data.

- Why use SGD?
 - Your training data is too big
 - Cannot fit in your memory, takes too long to calculate the gradients
 - Sometimes smaller batch-size helps avoid suboptimal minimum
 - If your minibatches are I.I.D, SGD $\rightarrow$ GD
- Modern deep learning is founded on SGD

Evaluation

- How do you know when to stop SGD?
 - You can't optimize your model forever
- Using the value of your loss function
 - It's not very intuitive
 - Hard to tell when to stop? ($1e-3$? $1e-4$?)

Evaluation

- Popular Evaluation Metrics
 - Accuracy
 - Used for multi-class classification
 - Area under the ROC (AUROC)
 - Used for binary classification
 - Precision & Recall
 - Used for information retrieval
 - BLEU Score
 - Used for machine translation
 - Perplexity
 - Used for language modeling
 - FID Score
 - Use for image generation

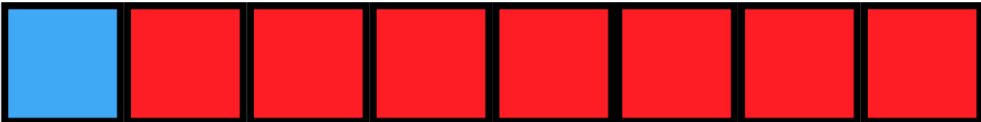
Train & Validation & Test

- Split the entire dataset into three sets
- Training set
 - Use this set to train your model
- Validation set (i.e. development set)
 - Use this set to evaluate your model, and decide when to stop training
- Test set
 - Use this “unseen” set to evaluate the final model
 - Don’t use this set during the training phase!!

N-fold Cross Validation

- Test your model's performance in diverse train/validation/test splits
 - Maybe you got lucky with an easy split, so test N times

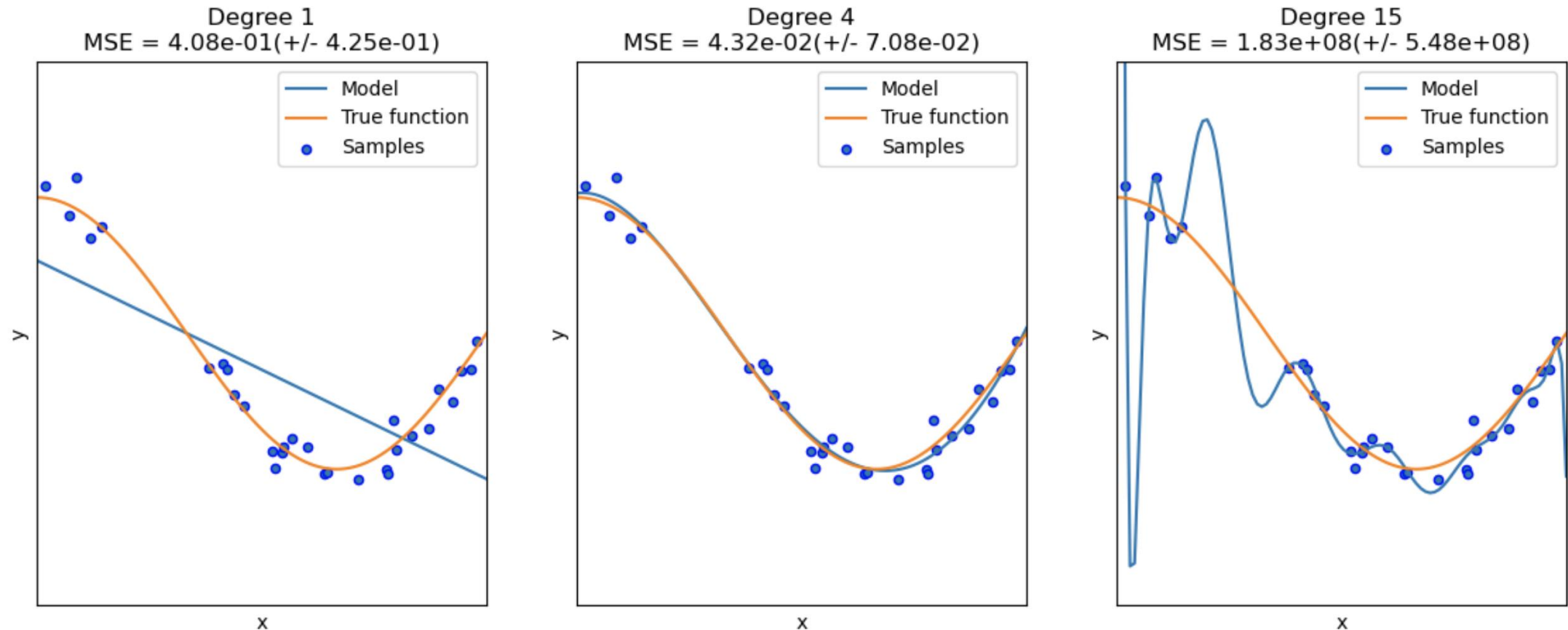
$n = 8$  Test  Train

Model 1 

- Not often used in modern deep learning
 - Dataset is too large → Time-consuming & statistically unnecessary

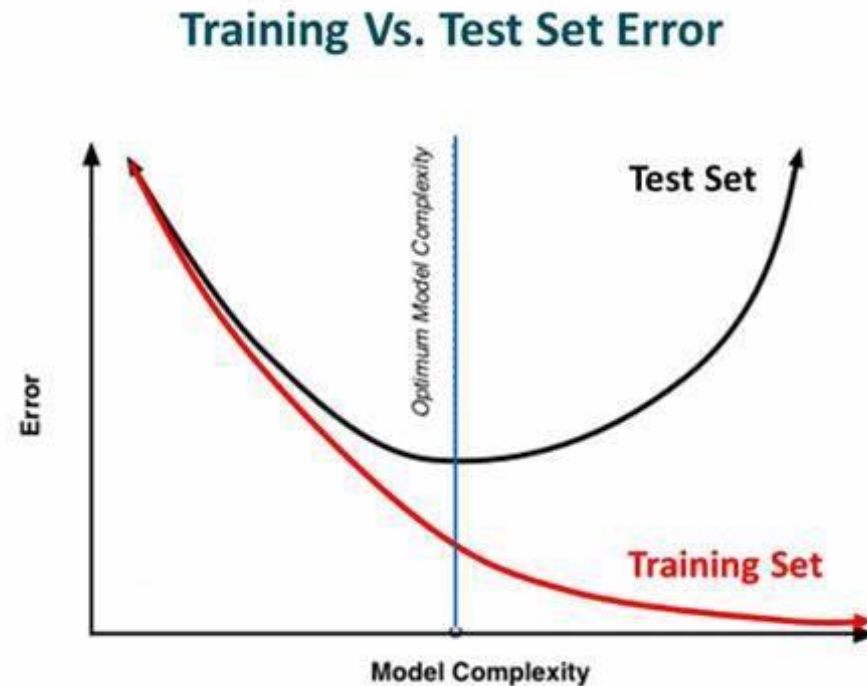
Overfitting & Underfitting

- Data complexity VS model capacity

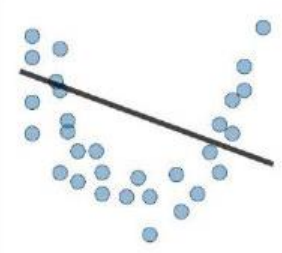
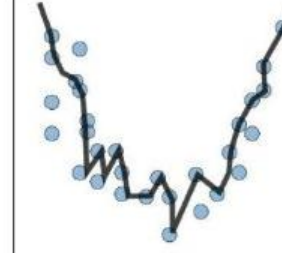
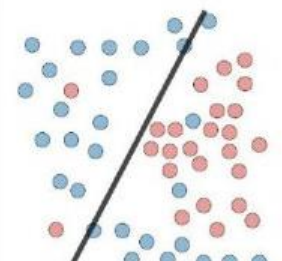
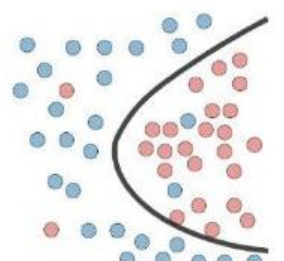
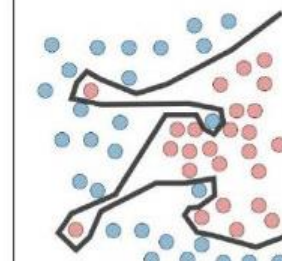
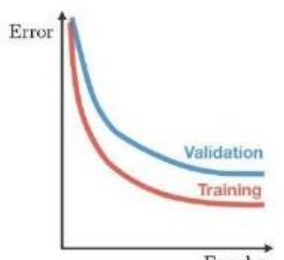
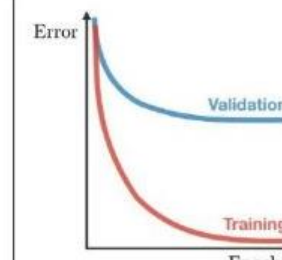


Overfitting to Training Data

- Doing too well on training data doesn't always lead to good test performance
 - Does not “generalize” well to unseen data

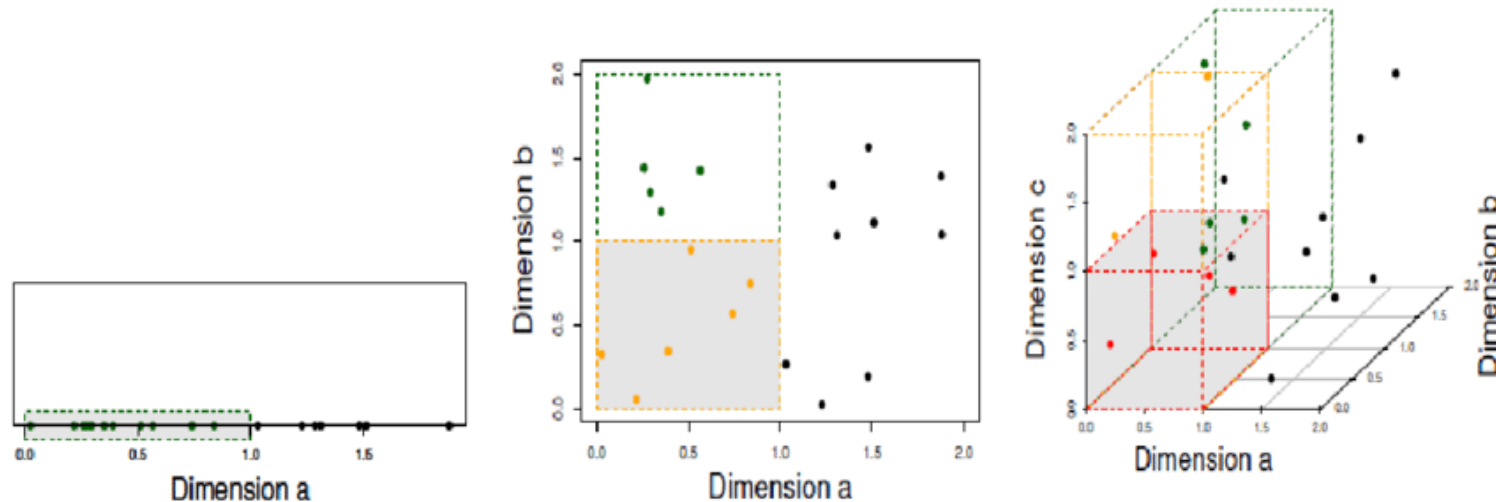


Overfitting & Underfitting

	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data

Curse of Dimensionality

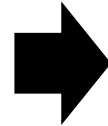
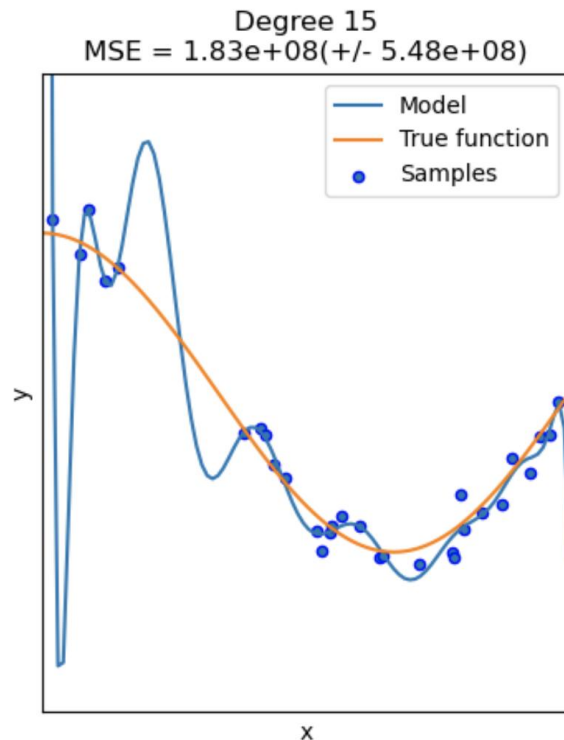
- When underfitting
 - Add more features
- linear $\uparrow$ in # feature $\Rightarrow$ exponential $\uparrow$ in # data to fill the space



- Still plays some role in modern deep learning

Regularization

- Restrict the freedom of your model
 - Downsizing the hypothesis space



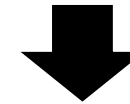
$$w_{15}x^{15} + w_{14}x^{14} + \dots + w_1x^1 + b = \mathbf{w}^T \mathbf{x} + b$$

$$w_{15} = 191.14$$

$$w_{14} = -89.32$$

$$w_{13} = 239.81$$

...

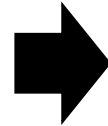
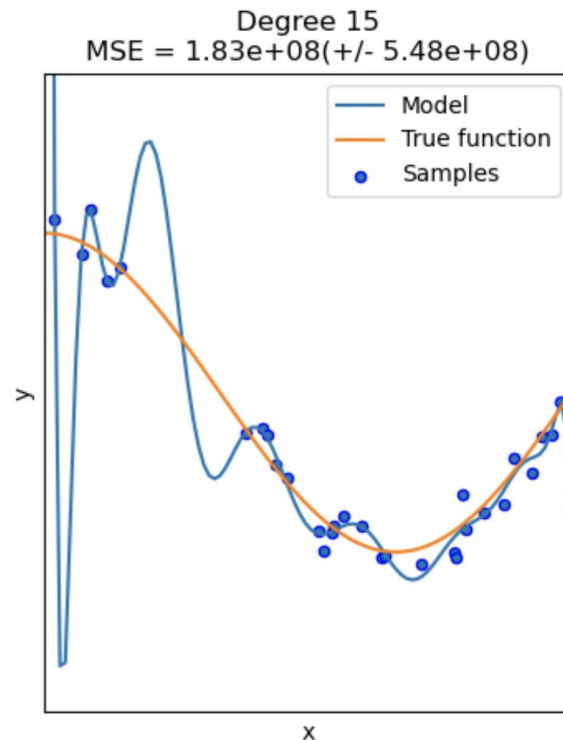


Objective function with L_2 regularization:

$$\mathcal{L}(y, y') + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Regularization

- Restrict the freedom of your model
 - Downsizing the hypothesis space



$$w_{15}x^{15} + w_{14}x^{14} + \dots + w_1x^1 + b = \mathbf{w}^T\mathbf{x} + b$$

$$w_{15} = 191.14$$

$$w_{14} = -89.32$$

$$w_{13} = 239.81$$

...

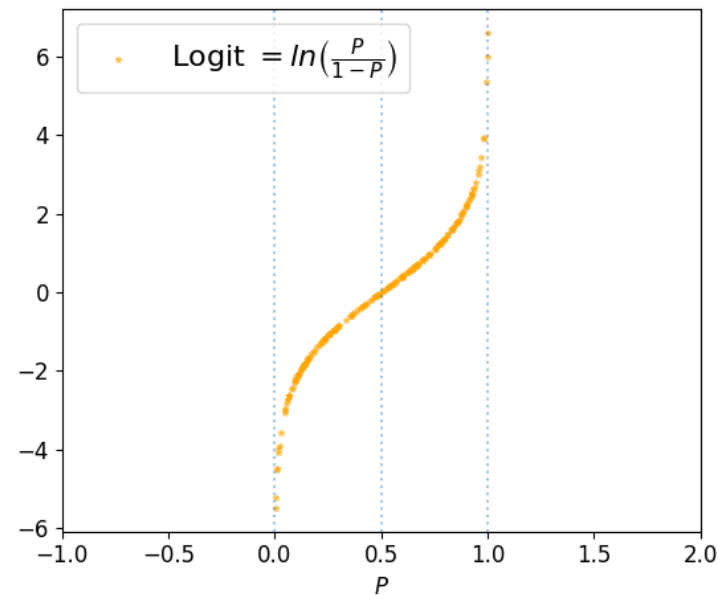


Objective function with L_1 regularization:

$$\mathcal{L}(y, y') + \lambda(|w_{15}| + |w_{14}| + \dots)$$

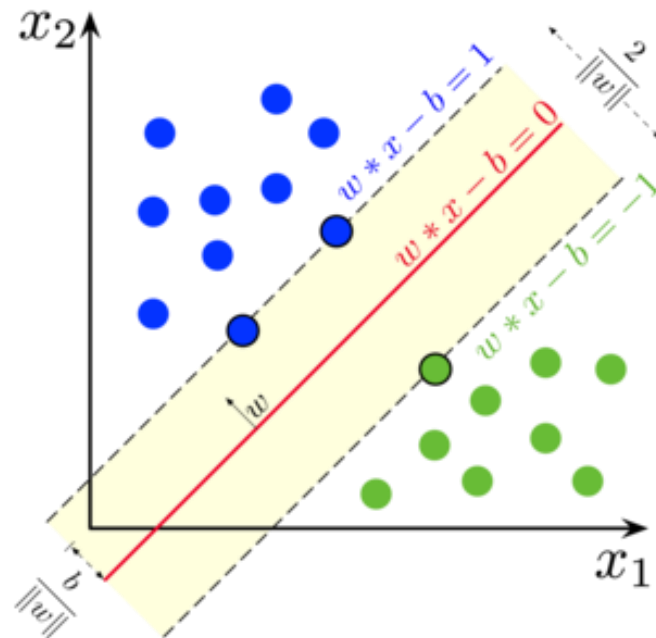
Popular Classifiers

- Logistic Regression
 - Probability $\rightarrow$ Odds $\rightarrow$ Log of odds (Logit)
 - Assume the logit can be modeled linearly
 - Trained via gradient descent



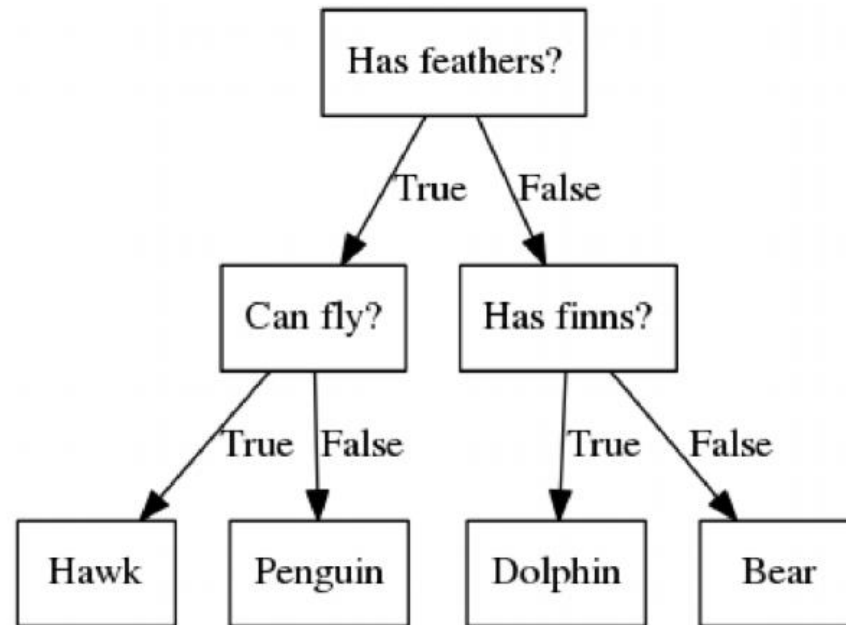
Popular Classifiers

- Support Vector Machine
 - Maximize the margin between two classes
 - Trained via constrained optimization (Lagrange Multiplier)
 - Or via gradient descent with hinge loss



Popular Classifiers

- Decision Tree
 - Build a tree based on features
 - Trained via the CART algorithm



Ensembles

- Use multiple classifiers (or regressors) to improve performance
- Bagging
 - Train multiple classifiers on different subsets of data
 - Train multiple classifiers on different subsets of features



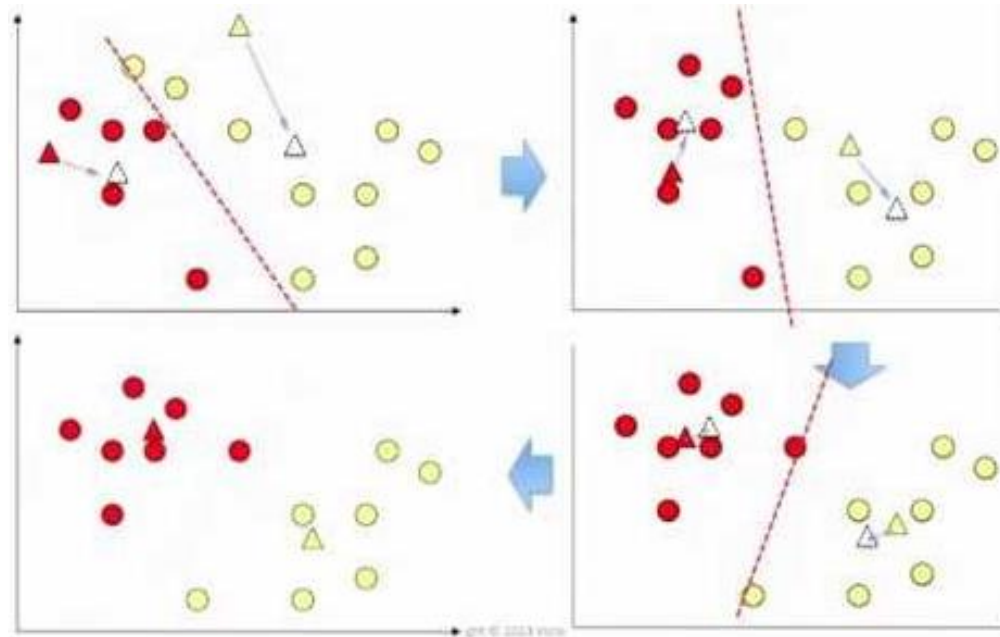
Ensembles

- Use multiple classifiers (or regressors) to improve performance
- Bagging
- Boosting
 - (k+1)-th classifier tries to correct the k-th classifiers errors.



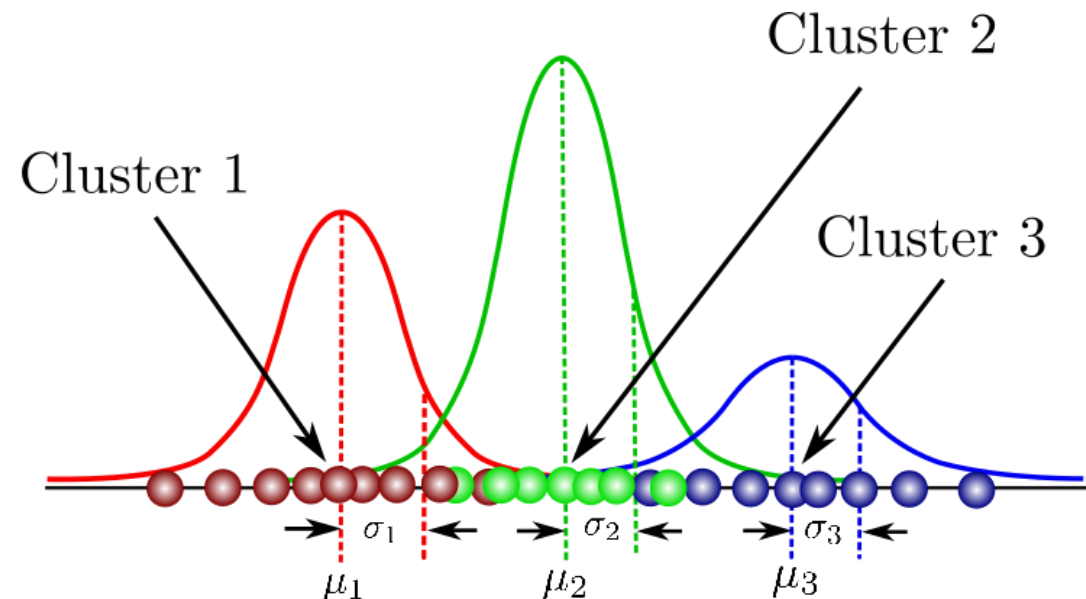
Popular Clustering

- K-means
 - Update membership of each sample using the closest centroid.
 - Update the centroid value using all the member samples.
 - Repeat the above steps



Popular Clustering

- Mixture of Gaussian
 - Generalization of K-means clustering
 - A sample has a probabilistic membership to each cluster
 - Trained via Expectation-Maximization (EM)



AI504: Programming for Artificial Intelligence

Week 2: Basic Machine Learning

Edward Choi

Grad School of AI

edwardchoi@kaist.ac.kr