Applied Machine Learning

Spring 2018, CS 519

Prof. Liang Huang

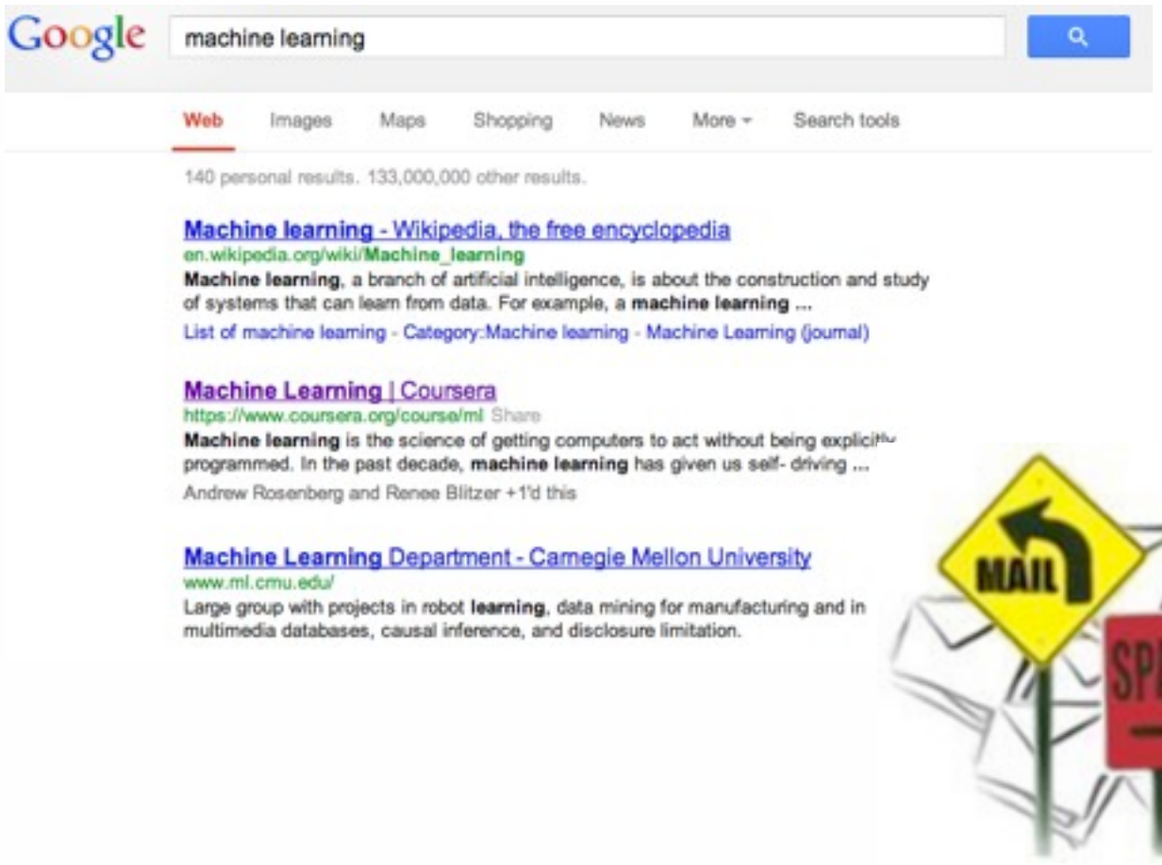
School of EECS

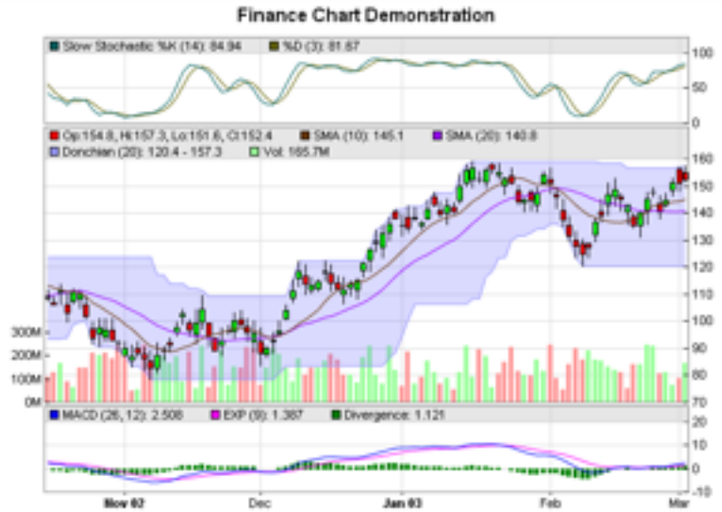
Oregon State University

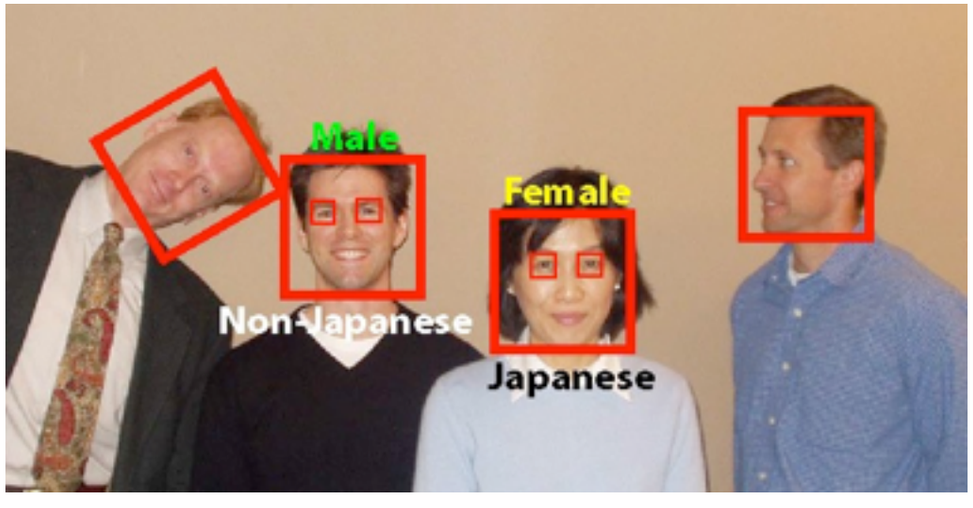
liang.huang@oregonstate.edu

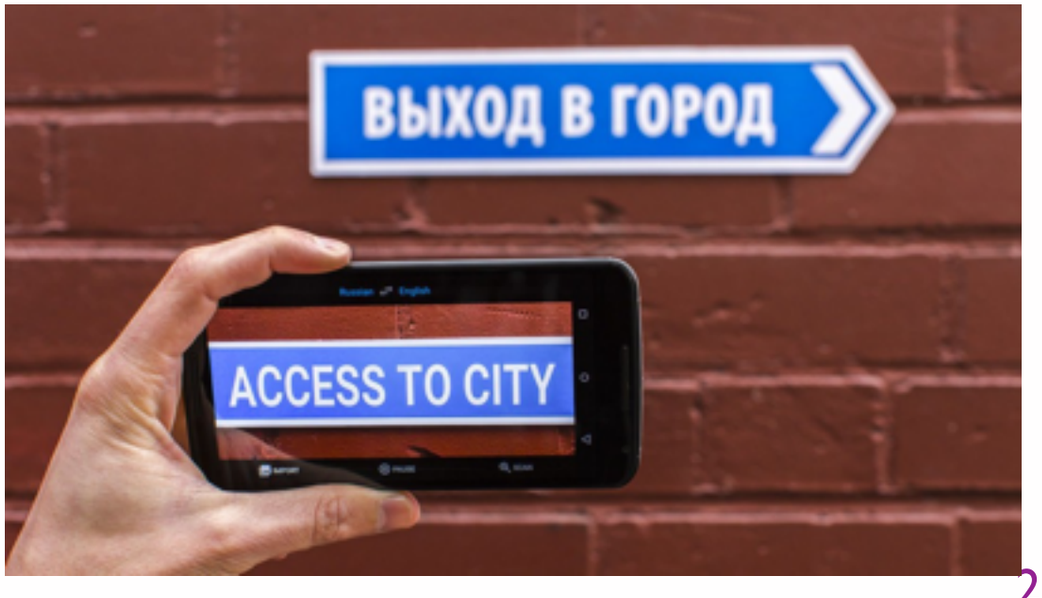
Machine Learning is Everywhere

- "A breakthrough in machine learning would be worth ten Microsofts" (Bill Gates)

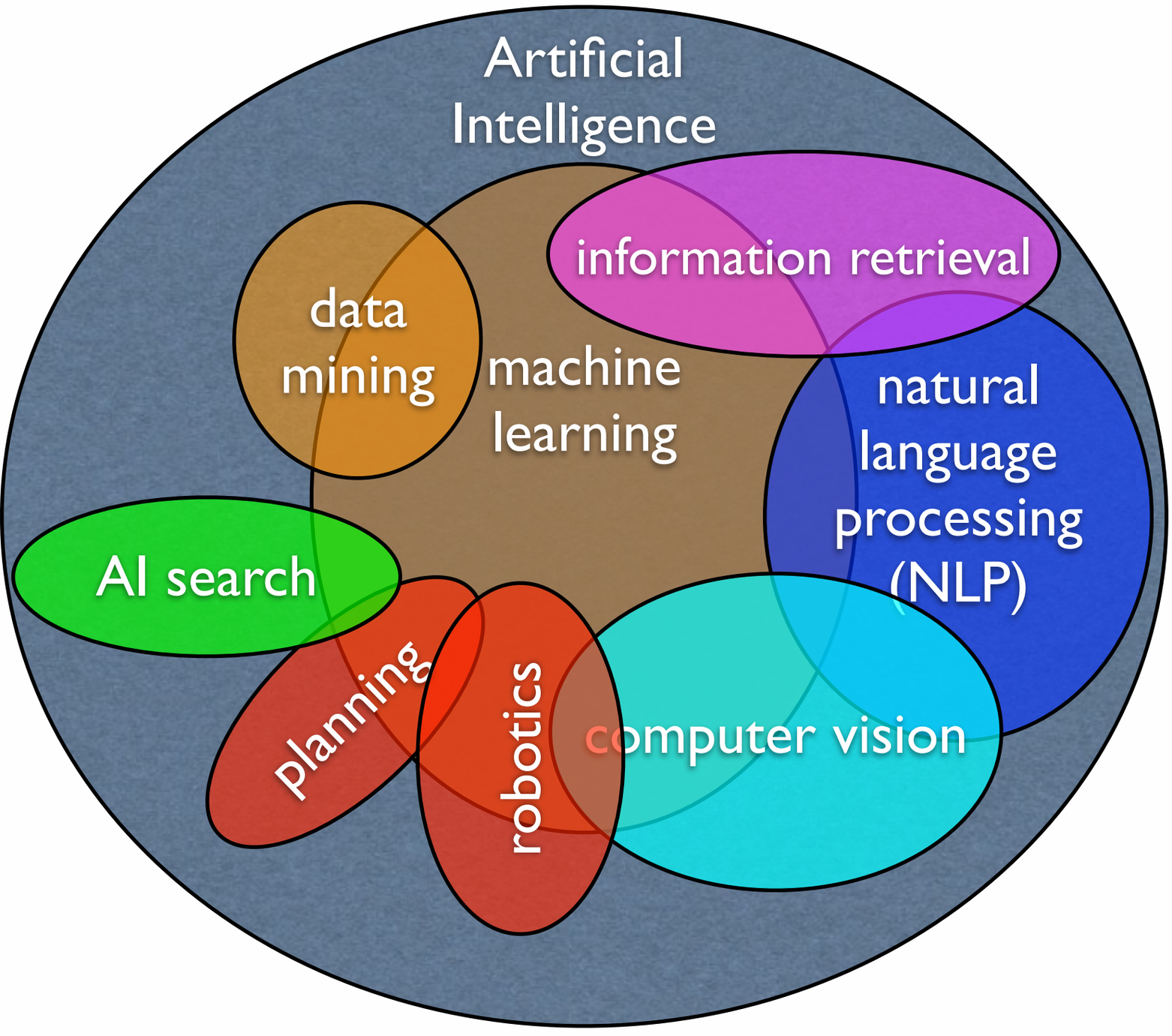




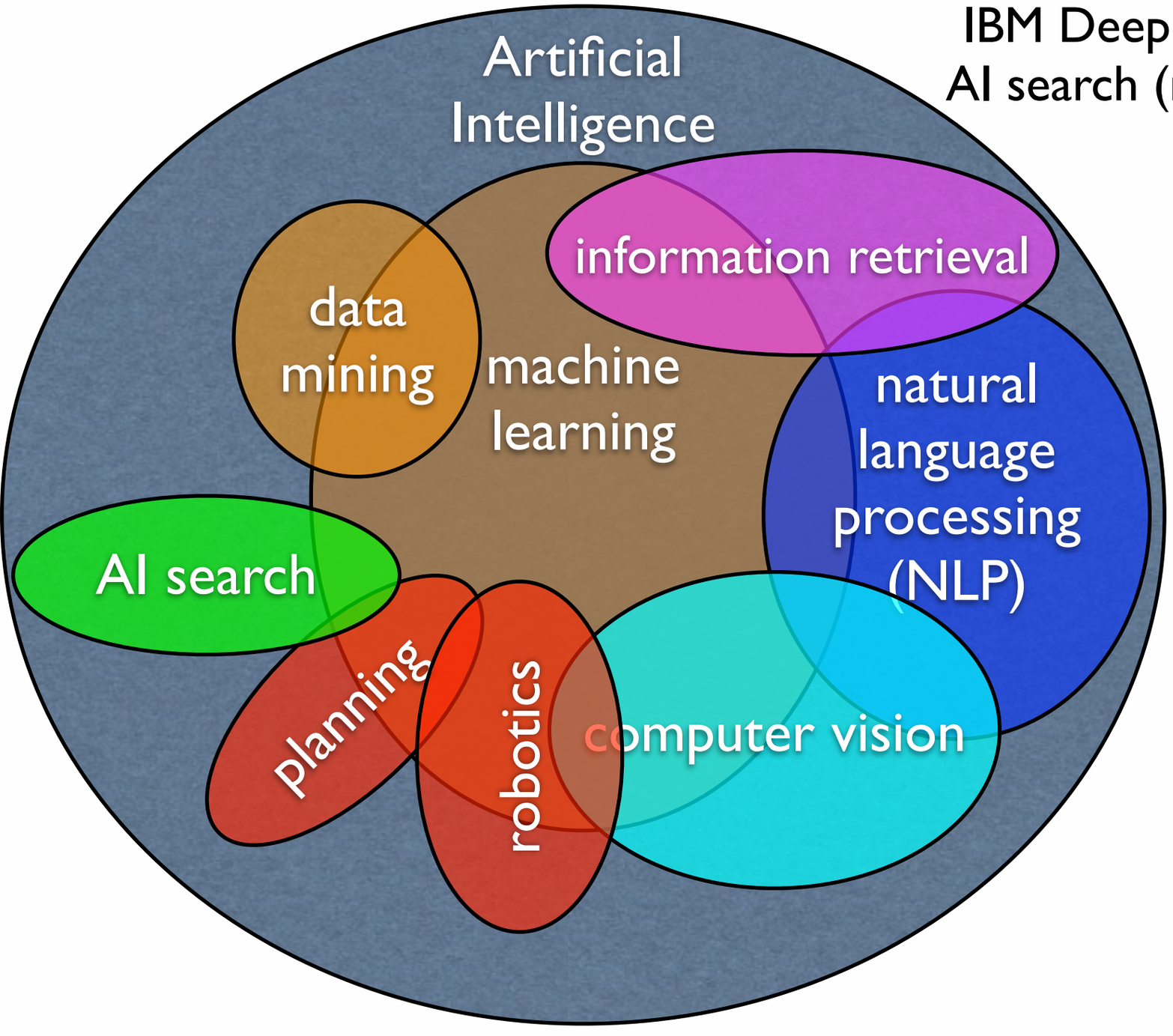




AI subfields and breakthroughs

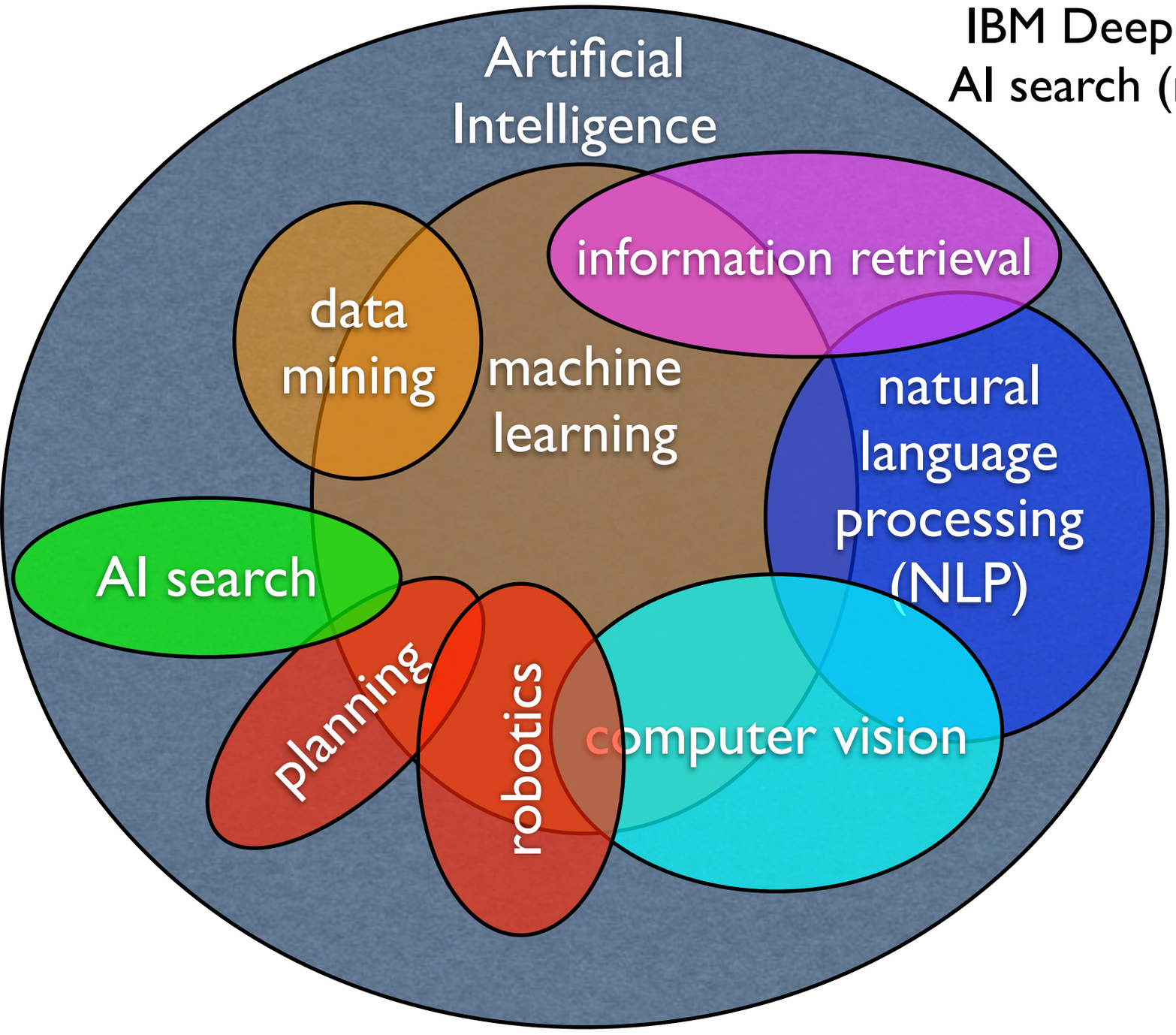


AI subfields and breakthroughs



IBM Deep Blue, 1997
AI search (no learning)

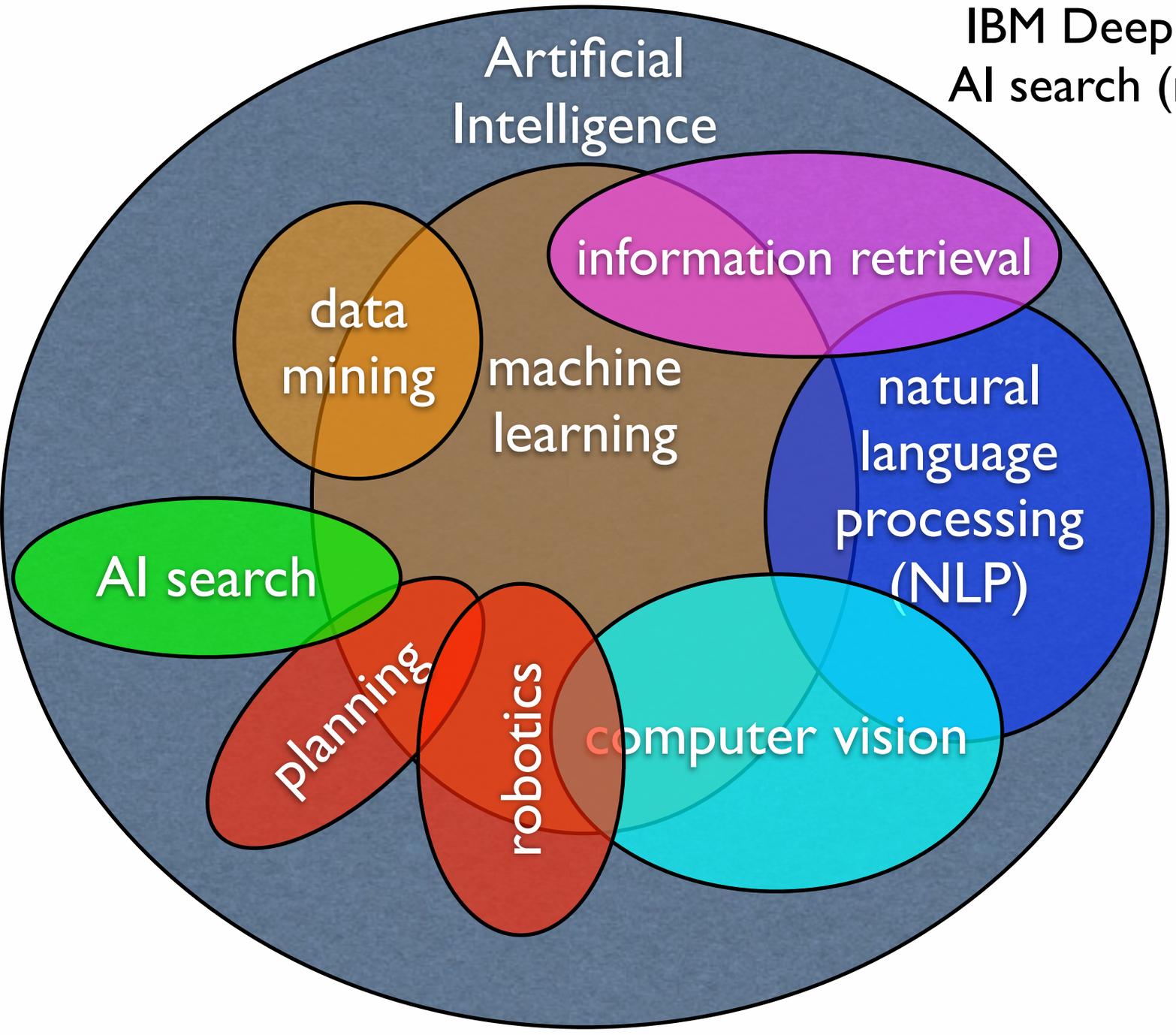
AI subfields and breakthroughs



IBM Deep Blue, 1997
AI search (no learning)

IBM Watson, 2011
NLP + very little ML

AI subfields and breakthroughs

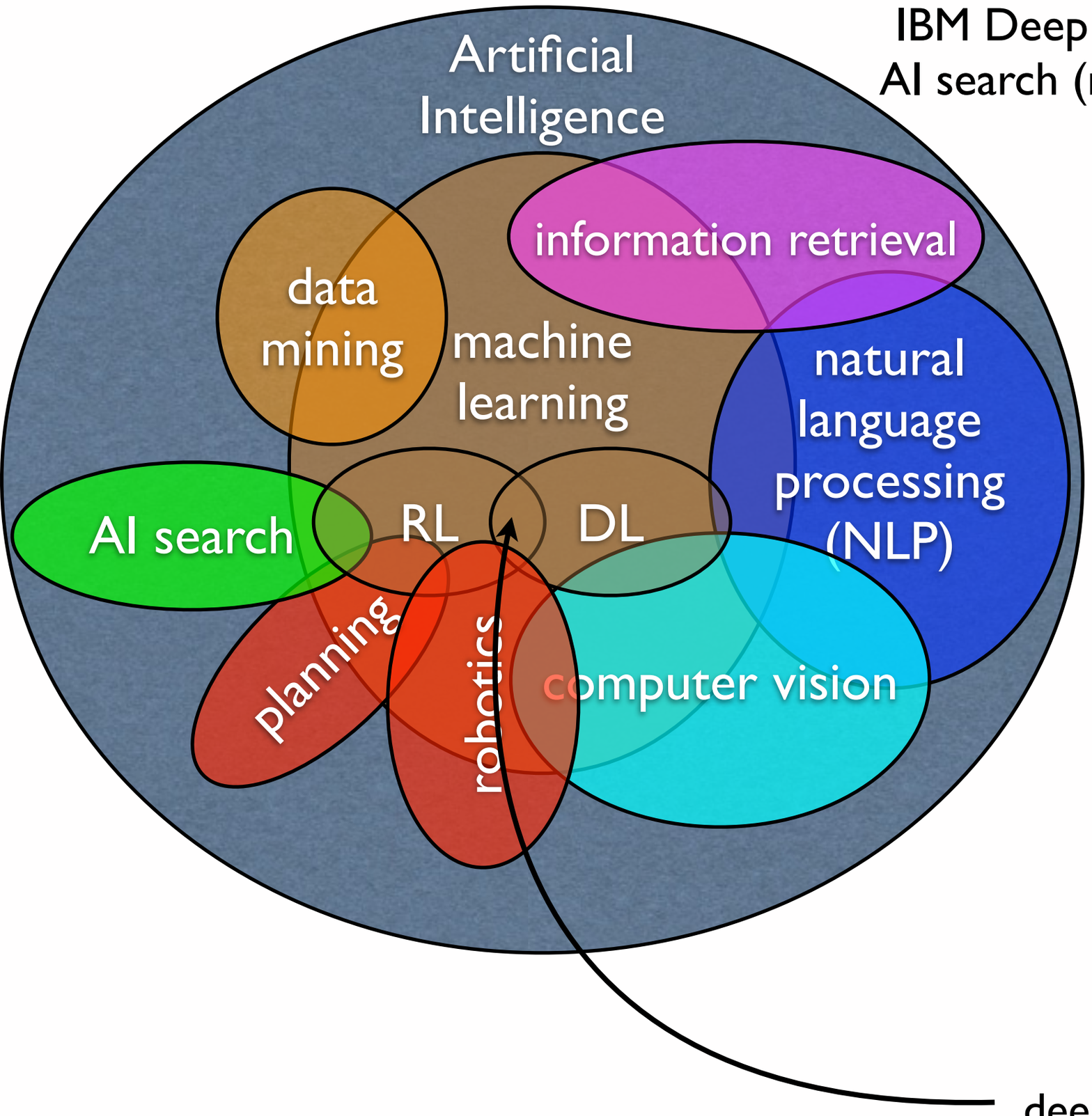


IBM Deep Blue, 1997
AI search (no learning)

IBM Watson, 2011
NLP + very little ML

Google DeepMind AlphaGo, 2017
deep reinforcement learning + AI search

AI subfields and breakthroughs



IBM Deep Blue, 1997
AI search (no learning)

IBM Watson, 2011
NLP + very little ML

Google DeepMind AlphaGo, 2017
deep reinforcement learning + AI search

The Future of Software Engineering

- "See when AI comes, I'll be long gone (being replaced by autonomous cars) but the programmers in those companies will be too, by automatic program generators." --- an Uber driver to an ML prof

Uber uses tons of AI/ML: route planning, speech/dialog, recommendation, etc.

Machine Learning Failures

Machine Learning Failures

liang's rule: if you see "X carefully" in China, just don't do it.

Machine Learning Failures

Machine Learning Failures

Machine Learning Failures

clear evidence that AI/ML is used in real life.

- Part II: Basic Components of Machine Learning Algorithms; Different Types of Learning

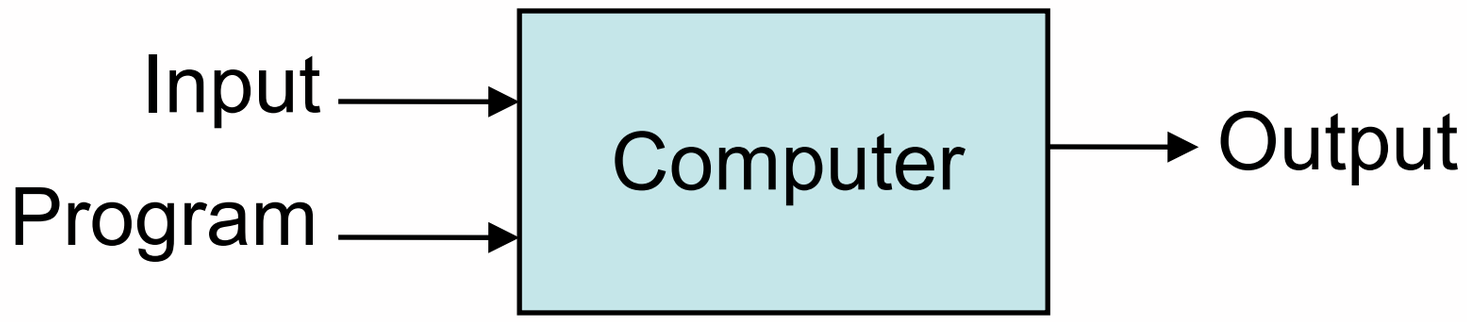
What is Machine Learning

- Machine Learning = Automating Automation
  - Getting computers to program themselves
  - Let the data do the work instead!

Traditional Programming

I love Oregon

rule-based translation (1950-2000)

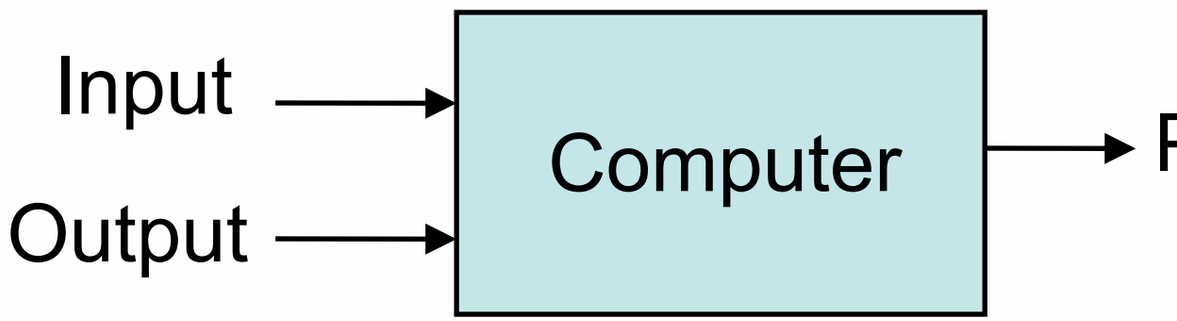


私はオレゴンが大好き

Machine Learning

I love Oregon

私はオレゴンが大好き



Google Translate (2003-now)

Magic?

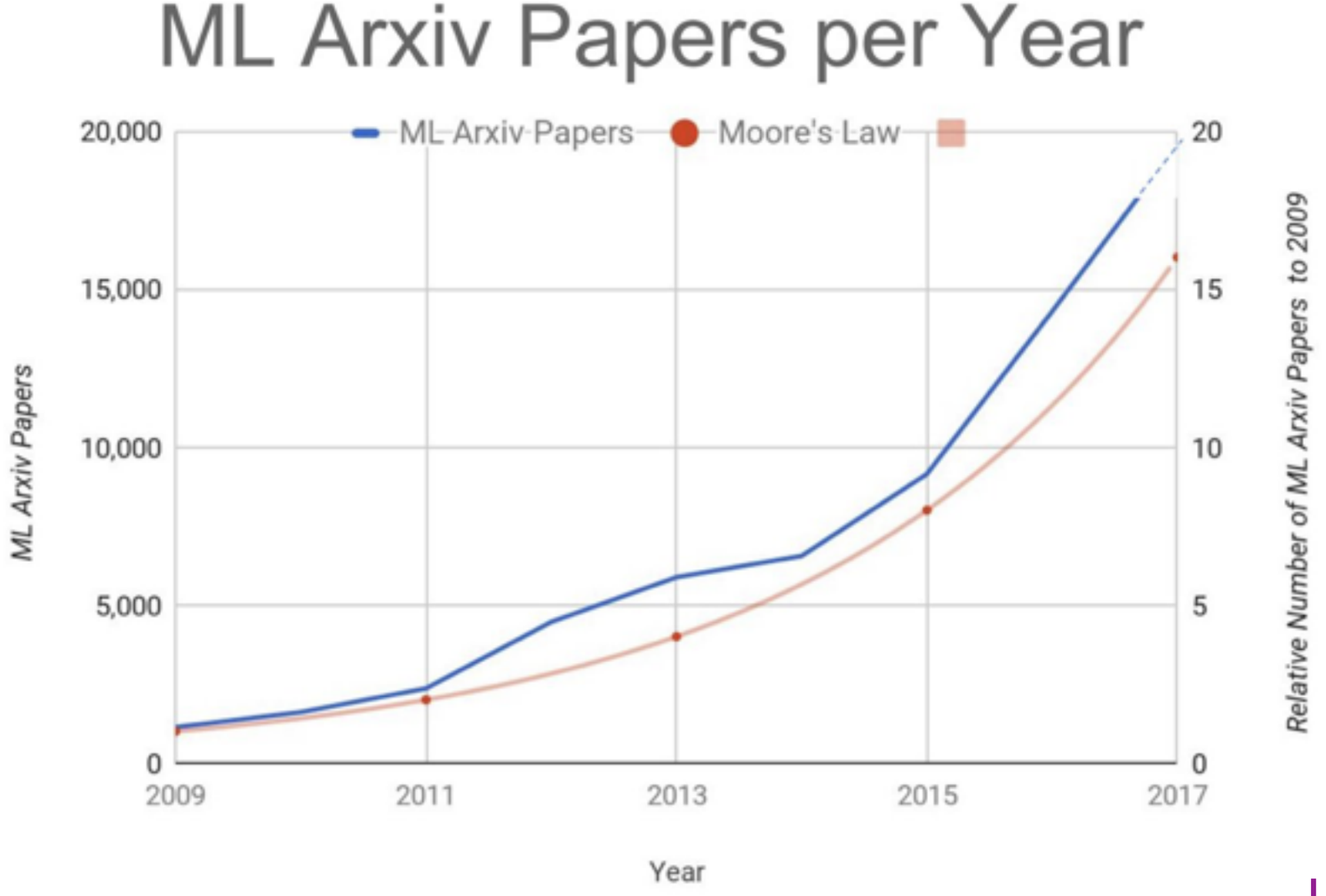
No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

"There is no better data than more data"

ML in a Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**



Representation

- Separating Hyperplanes
- Support vectors
- Decision trees
- Sets of rules / Logic programs
- Instances (Nearest Neighbor)
- Graphical models (Bayes/Markov nets)
- Neural networks
- Model ensembles
- Etc.

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Optimization

- Combinatorial optimization
  - E.g.: Greedy search, Dynamic programming
- Convex optimization
  - E.g.: Gradient descent, Coordinate descent
- Constrained optimization
  - E.g.: Linear programming, Quadratic programming

Gradient Descent

- if learning rate is too small, it'll converge very slowly
- if learning rate is too big, it'll diverge

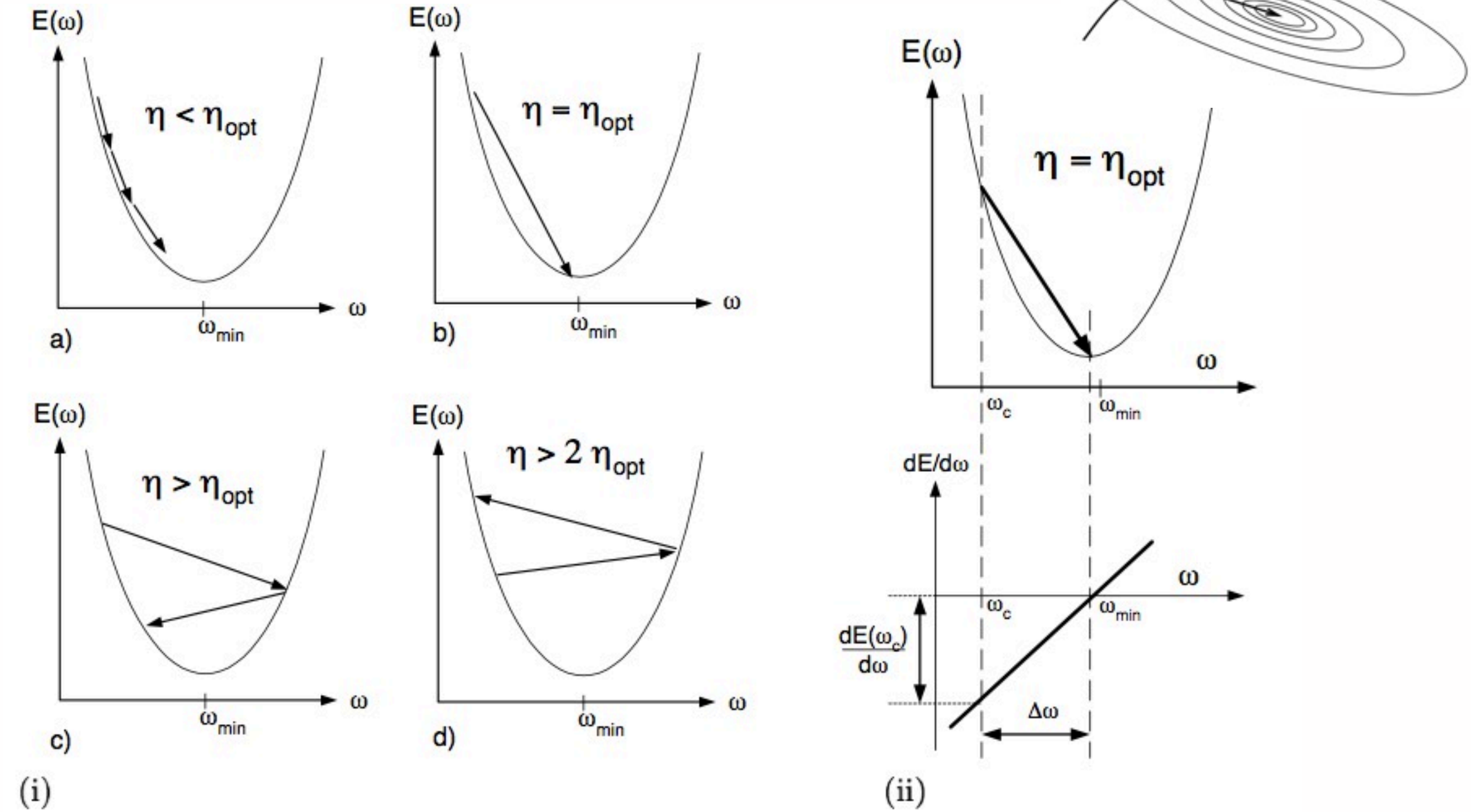
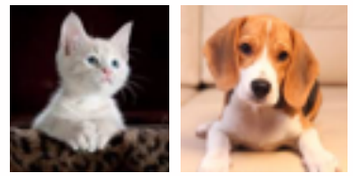


Fig. 6. Gradient descent for different learning rates.

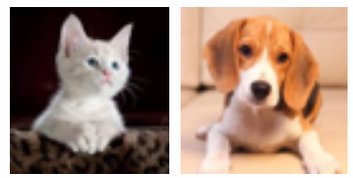
Types of Learning

- Supervised (inductive) learning
  - Training data includes desired outputs

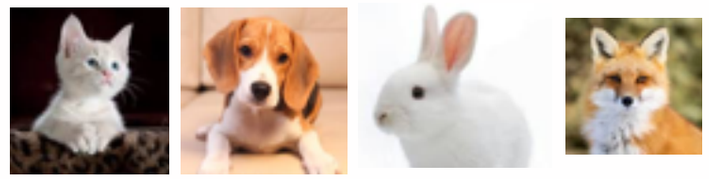


cat *dog*

- Unsupervised learning
  - Training data does not include desired outputs

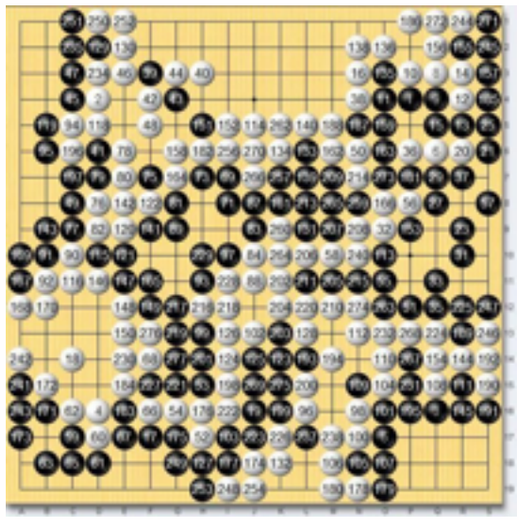


- Semi-supervised learning
  - Training data includes a few desired outputs



cat *dog*

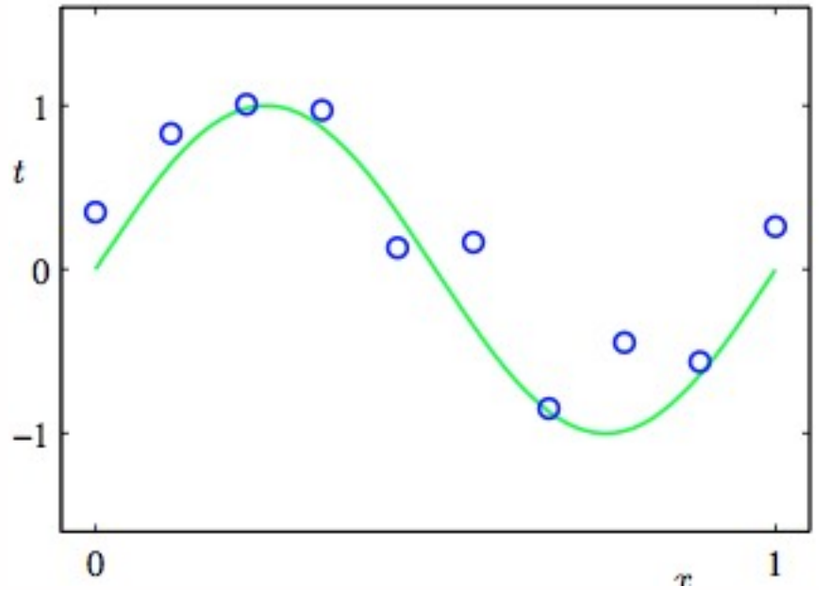
- Reinforcement learning
  - Rewards from sequence of actions

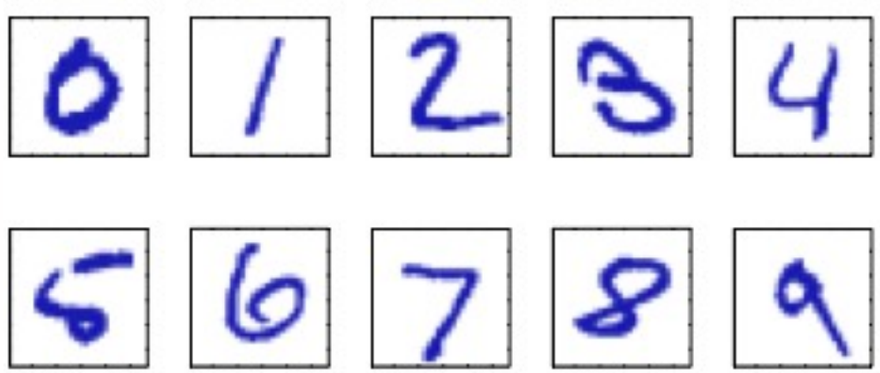


rules → white win

Supervised Learning

- Given examples (X, f(X)) for an unknown function f
- Find a good approximation of function f
  - Discrete f(X): Classification (binary, multiclass, structured)
  - Continuous f(X): Regression



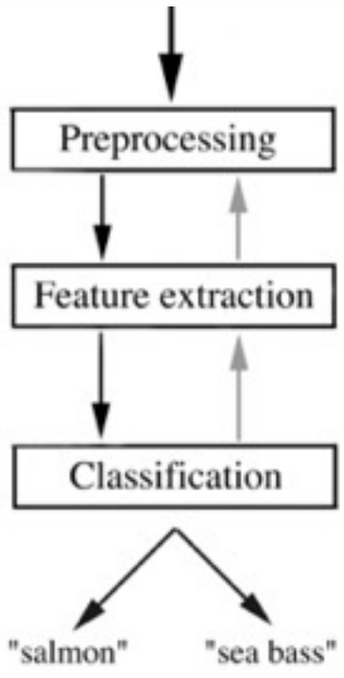


When is Supervised Learning Useful

- when there is no human expert
  - input *x*: bond graph for a new molecule
  - output *f*(*x*): predicted binding strength to AIDS protease
- when humans can perform the task but can't describe it
  - computer vision: face recognition, OCR
- where the desired function changes frequently
  - stock price prediction, spam filtering
- where each user needs a customized function
  - speech recognition, spam filtering

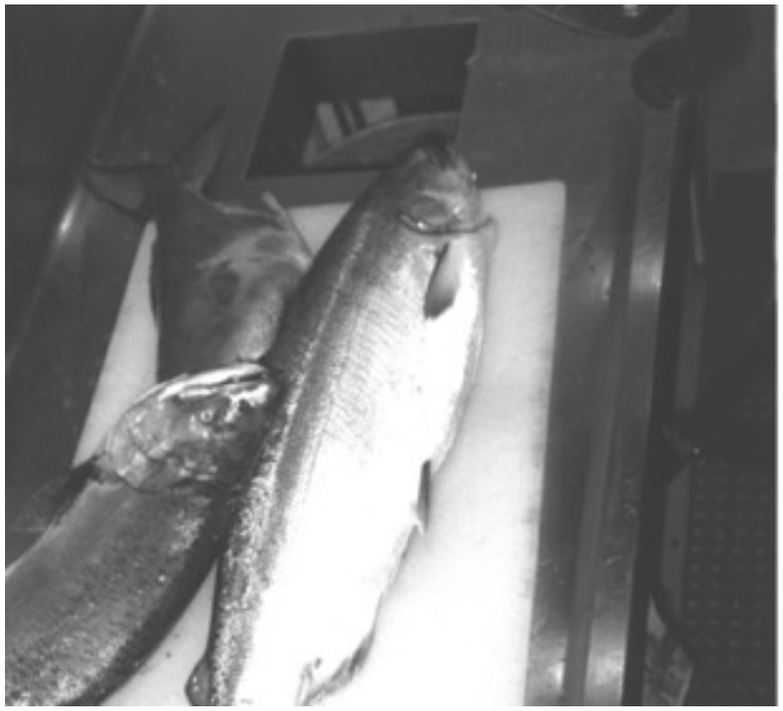
Supervised Learning: Classification

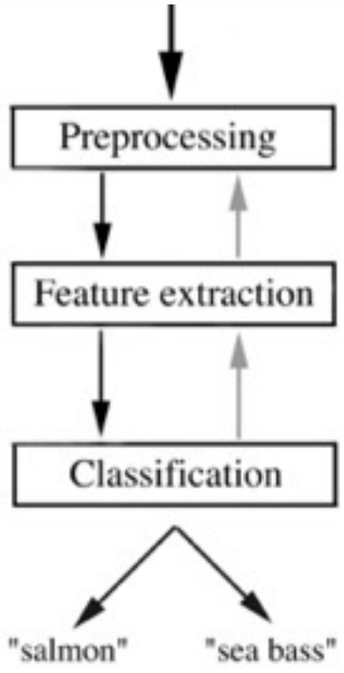
- input *X*: feature representation ("observation")

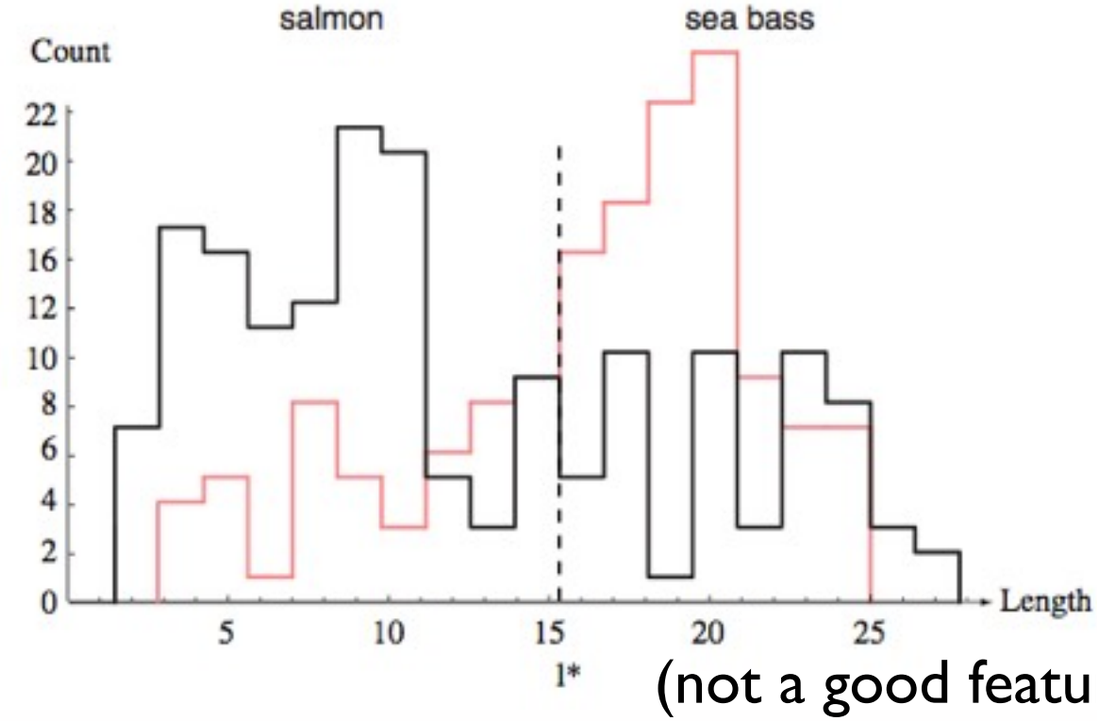


Supervised Learning: Classification

- input *X*: feature representation ("observation")



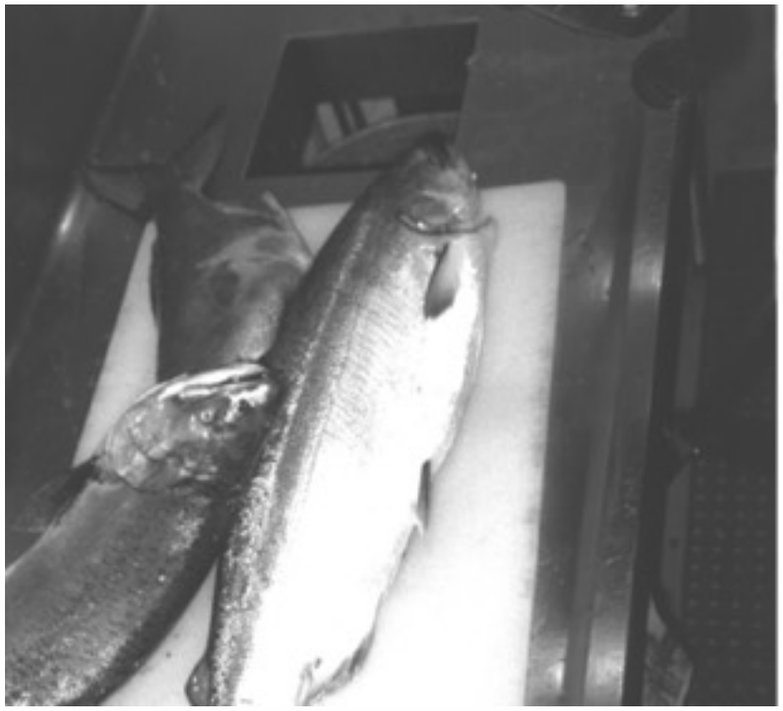


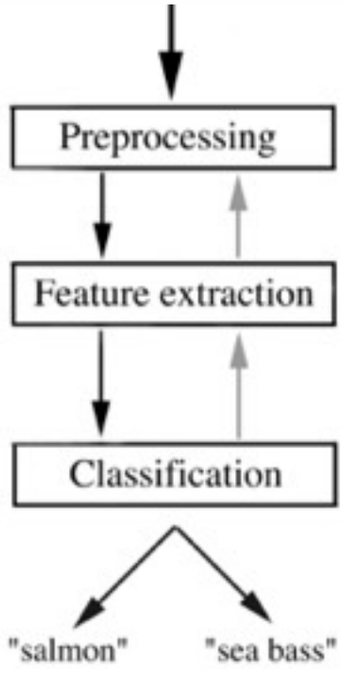


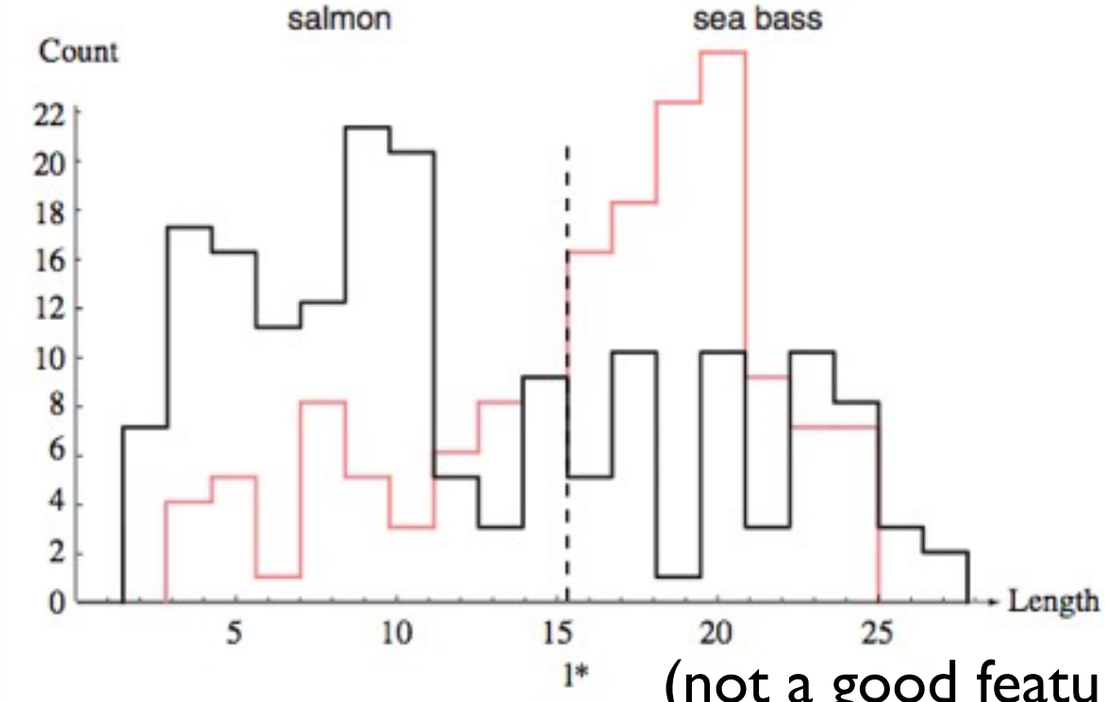
(not a good feature)

Supervised Learning: Classification

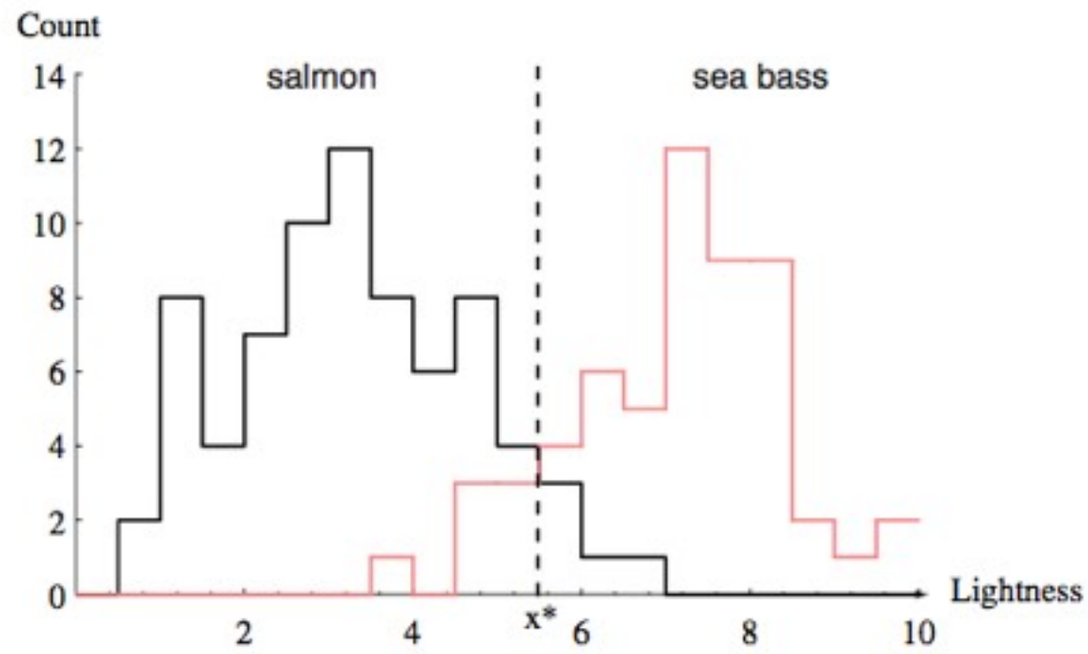
- input *X*: feature representation ("observation")







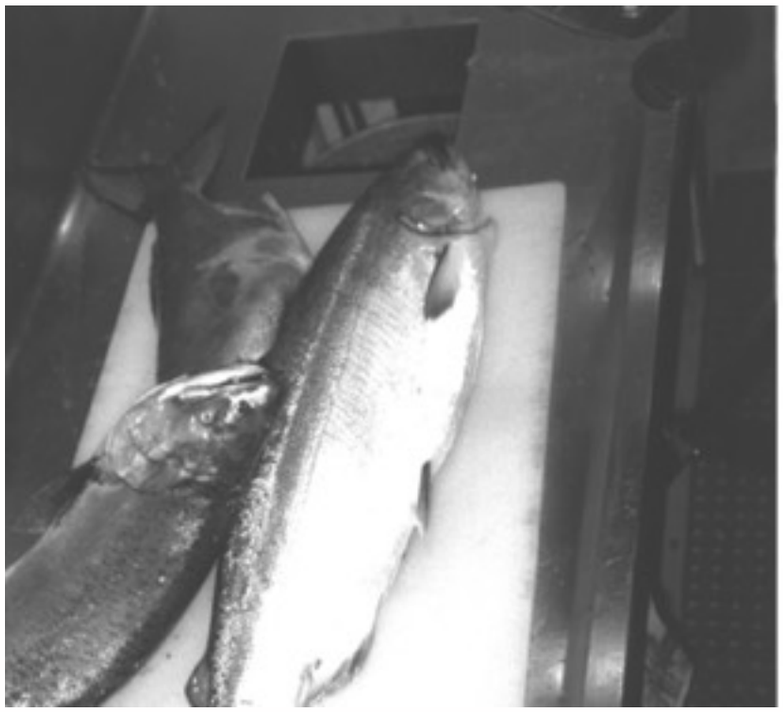
(not a good feature)

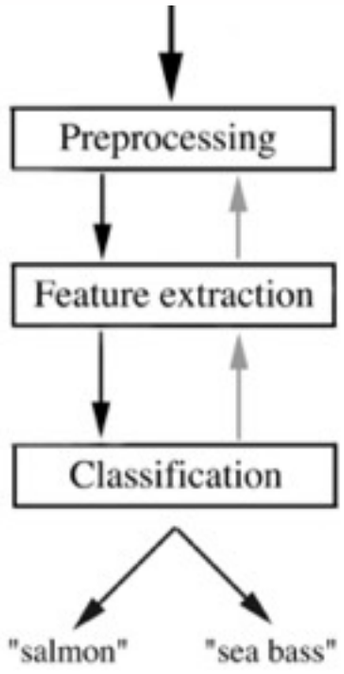


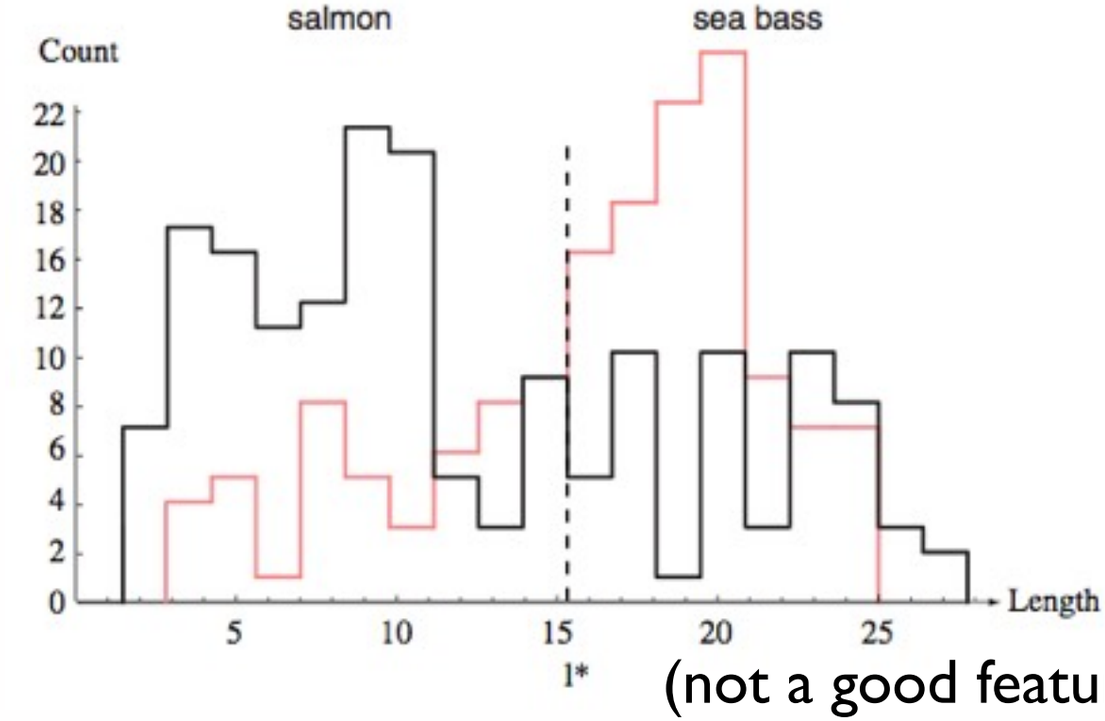
(a good feature)

Supervised Learning: Classification

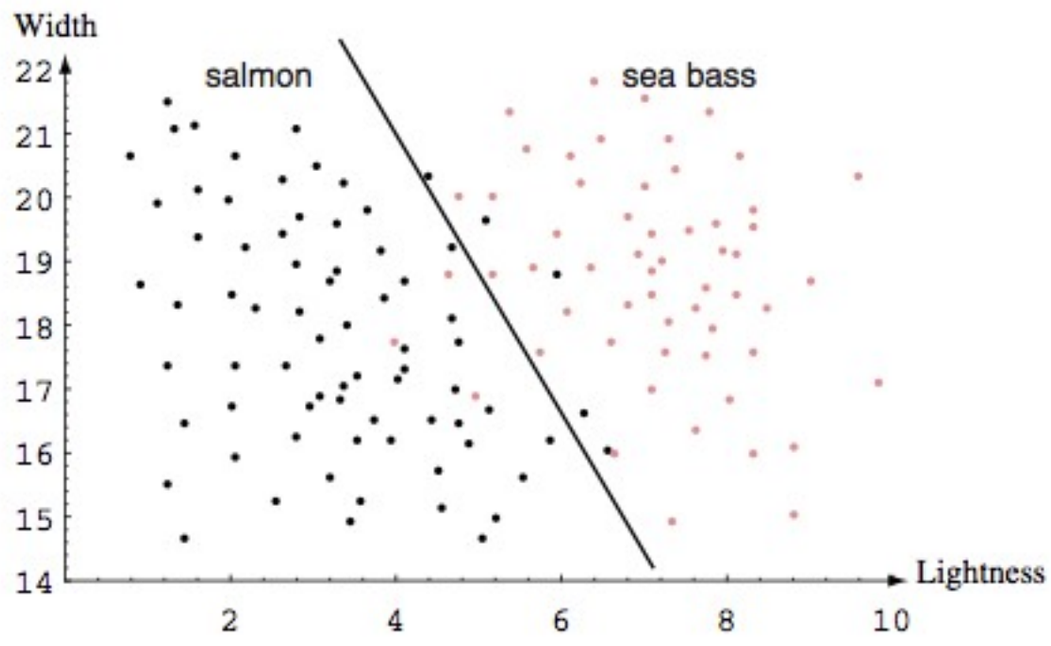
- input *X*: feature representation ("observation")

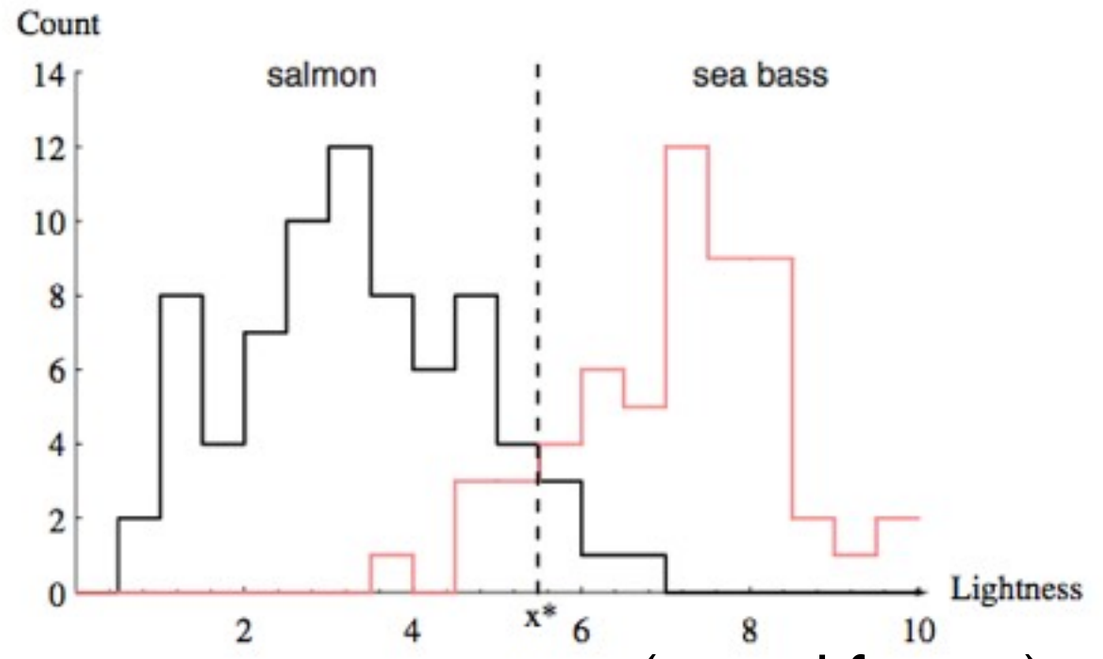






(not a good feature)

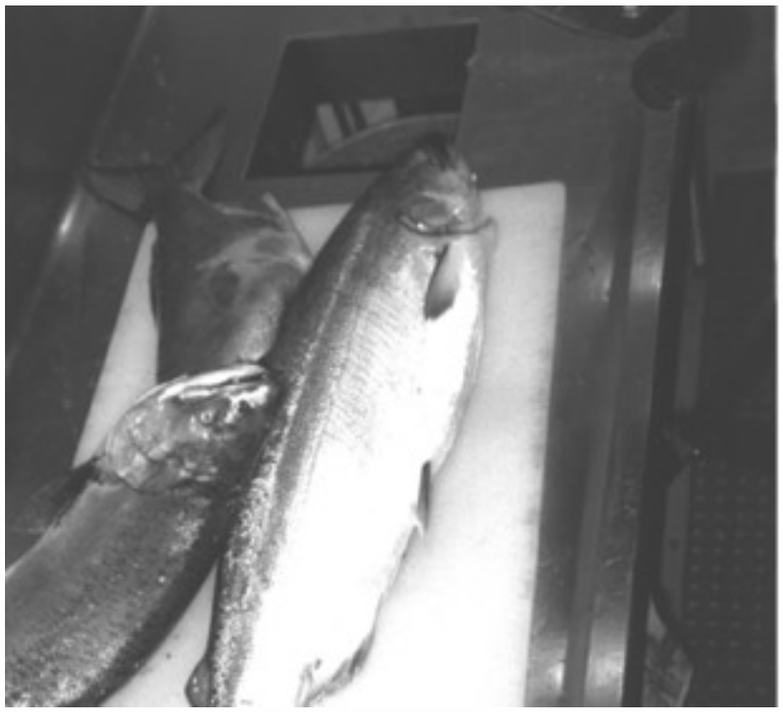


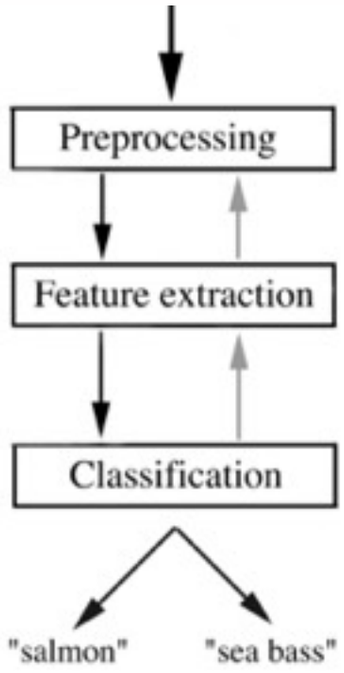


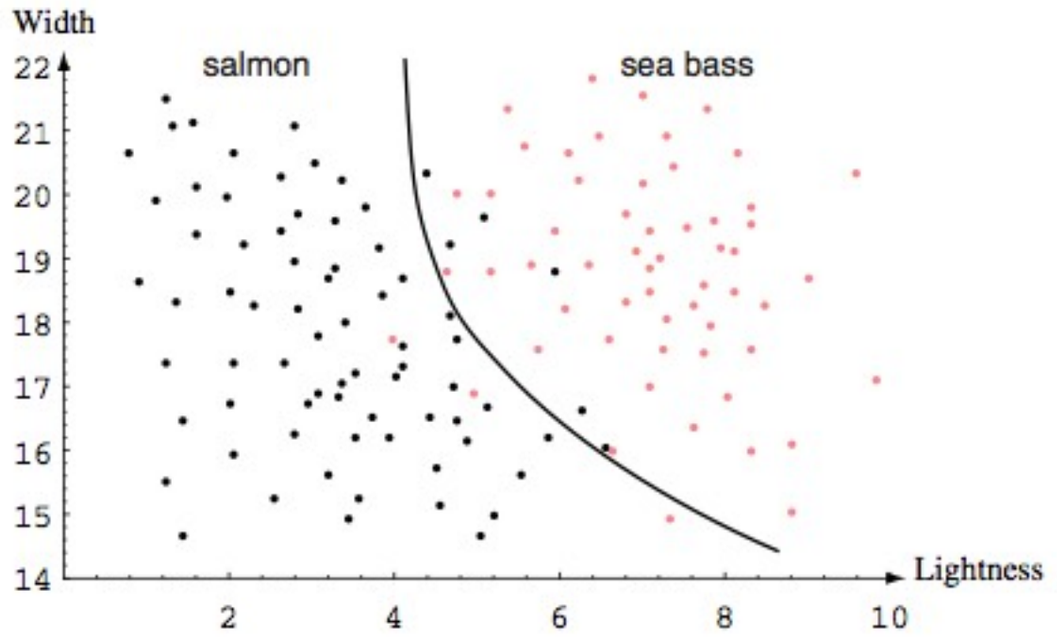
(a good feature)

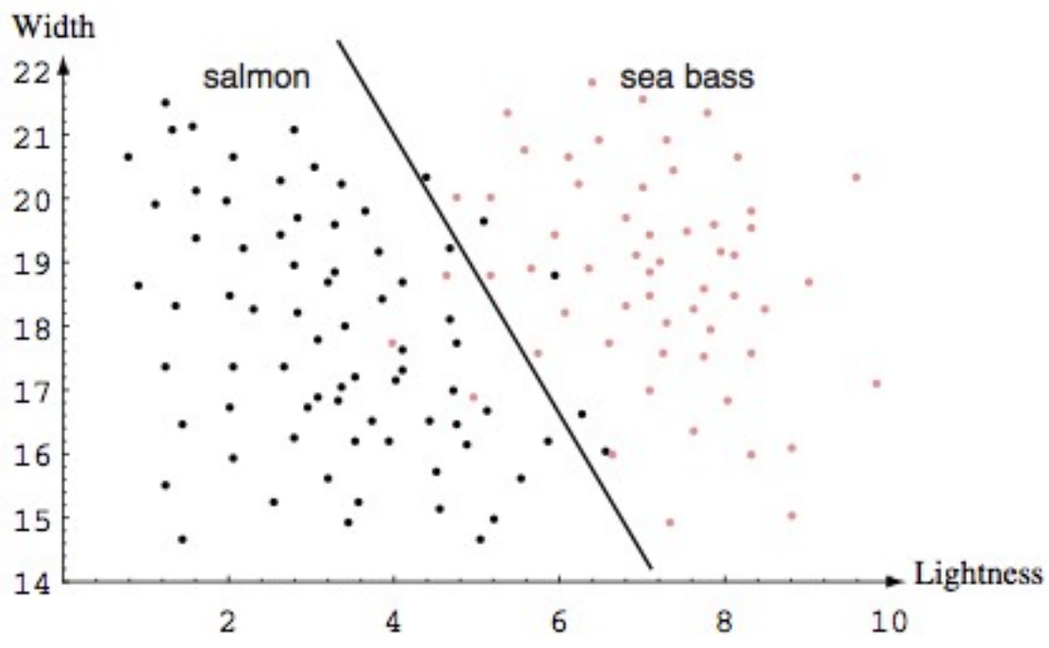
Supervised Learning: Classification

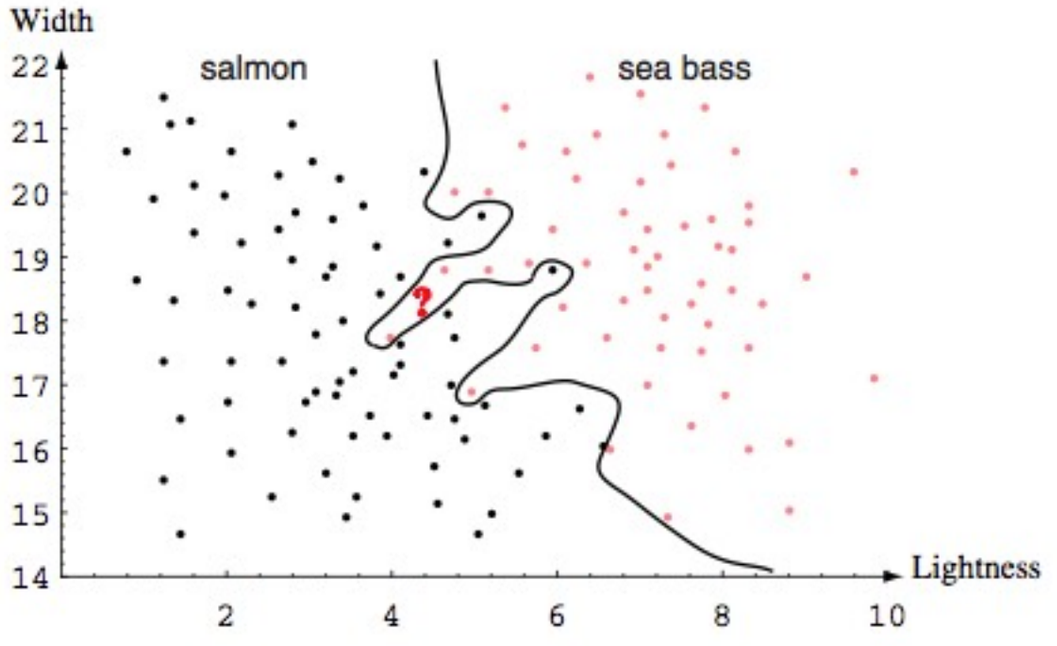
- input *X*: feature representation ("observation")





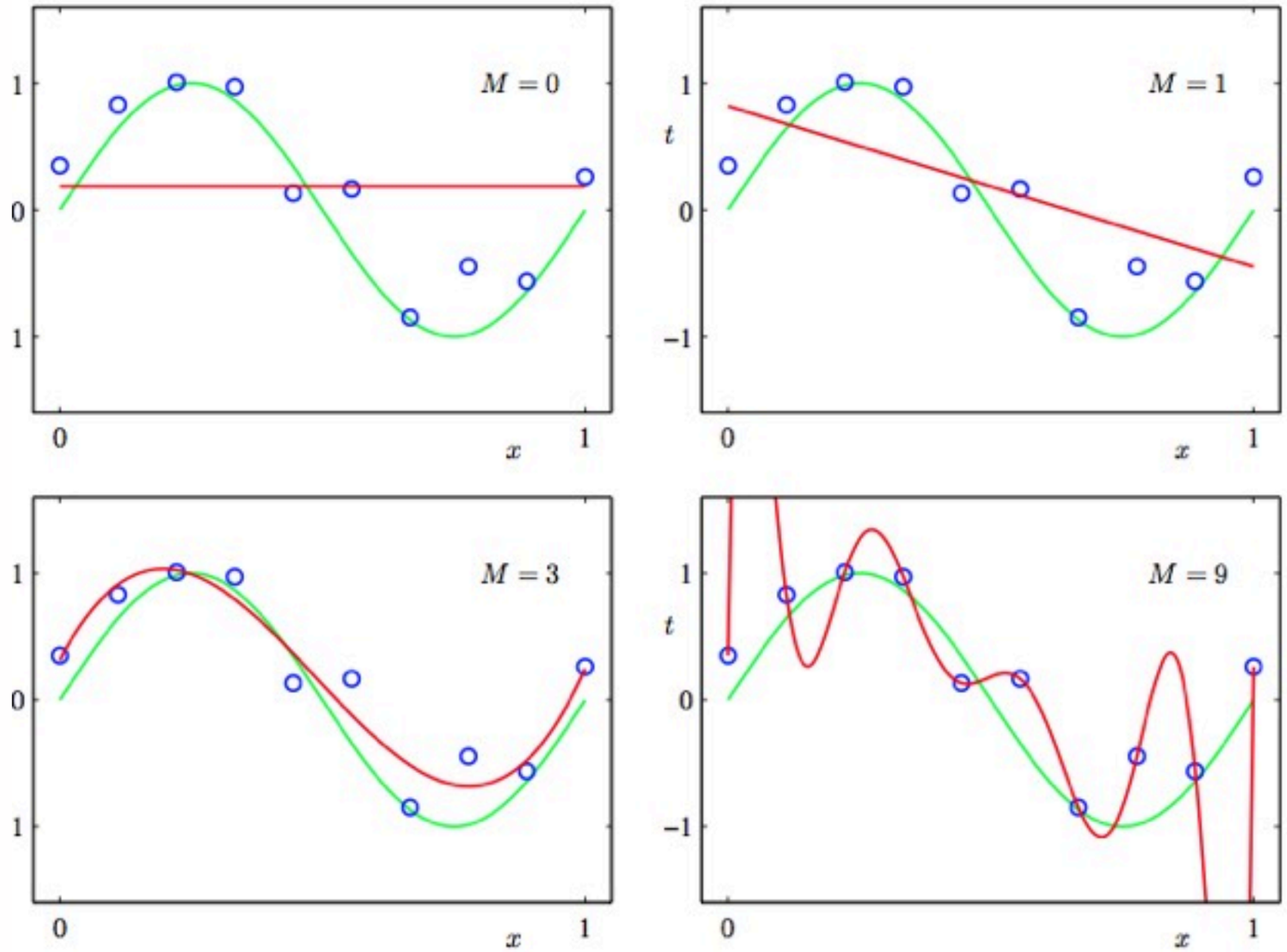






Supervised Learning: Regression

- linear and non-linear regression
- overfitting and underfitting (same as in classification)



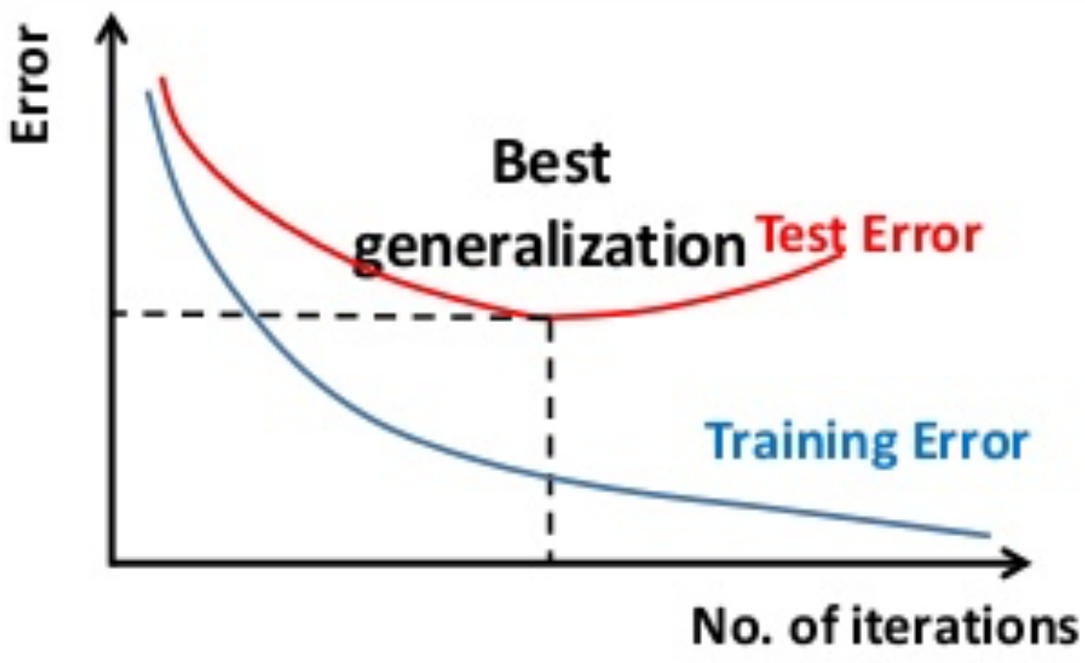
What We'll Cover

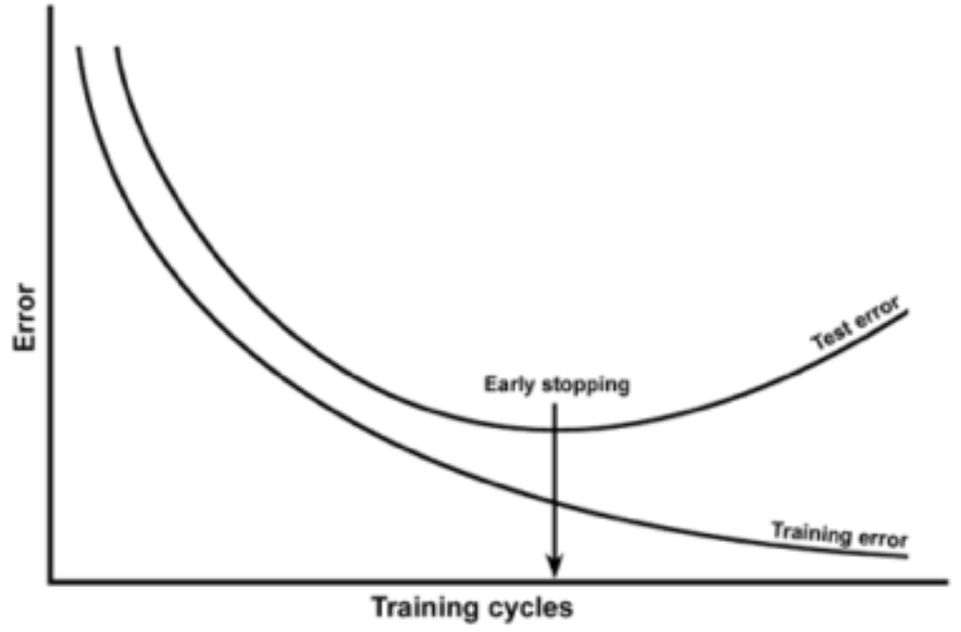
- Supervised learning
  - Nearest Neighbors (week 1)
  - Linear Classification (Perceptron and Extensions) (weeks 2-3)
  - Support Vector Machines (weeks 4-5)
  - Kernel Methods (week 5)
  - Structured Prediction (weeks 7-8)
  - Neural Networks and Deep Learning (week 10)
- Unsupervised learning (week 9)
  - Clustering (k-means, EM)
  - Dimensionality reduction (PCA etc.)

- Part III: Training, Test, and Generalization Errors; Underfitting and Overfitting; Methods to Prevent Overfitting; Cross-Validation and Leave-One-Out

Training, Test, & Generalization Errors

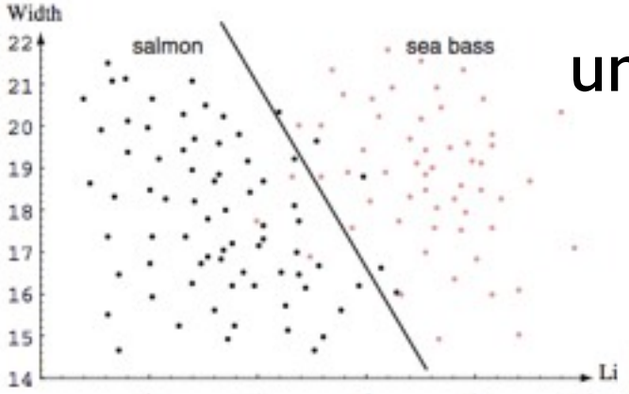
- in general, as training progresses, training error decreases
  - test error initially decreases, but eventually increases!
    - at that point, the model has overfit to the training data (memorizes noise or outliers)
- but in reality, you don't know the test data a priori ("blind-test")
  - generalization error: error on previously unseen data
  - expectation of test error assuming a test data distribution
  - often use a held-out set to simulate test error and do early stopping



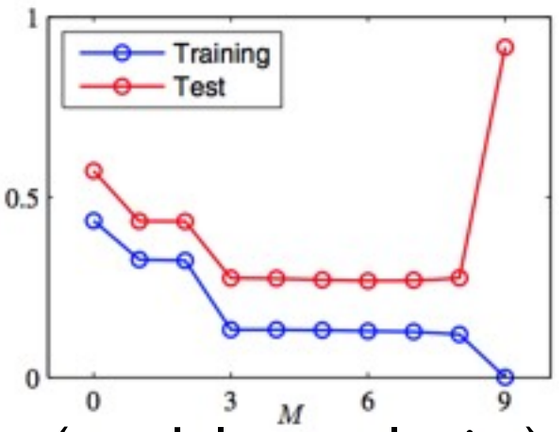


Under/Over-fitting due to Model

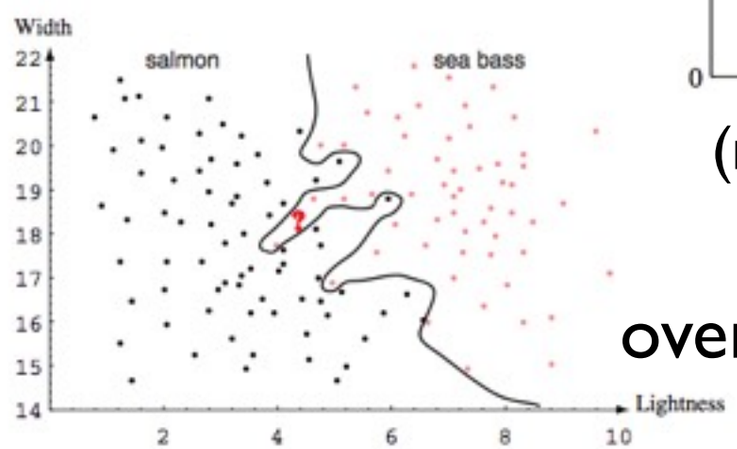
- underfitting / overfitting occurs due to under/over-training (last slide)
- underfitting / overfitting also occurs because of model complexity
  - underfitting due to oversimplified model ("*as simple as possible, but not simpler!*")
  - overfitting due to overcomplicated model (memorizes noise or outliers in data!)
    - extreme case: the model memorizes the training data, but no generalization!



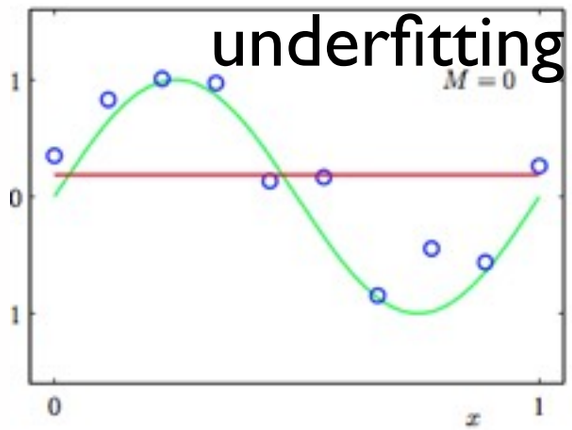
underfitting

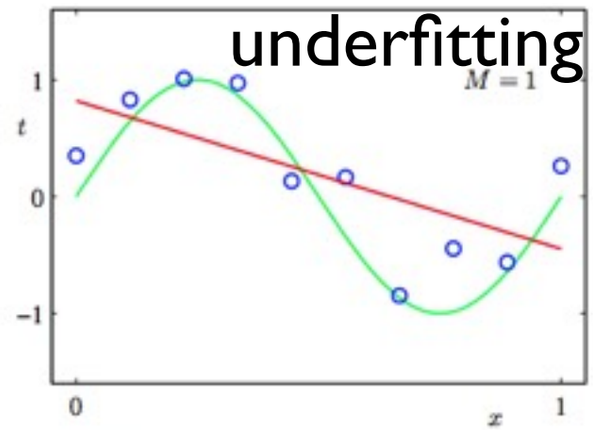


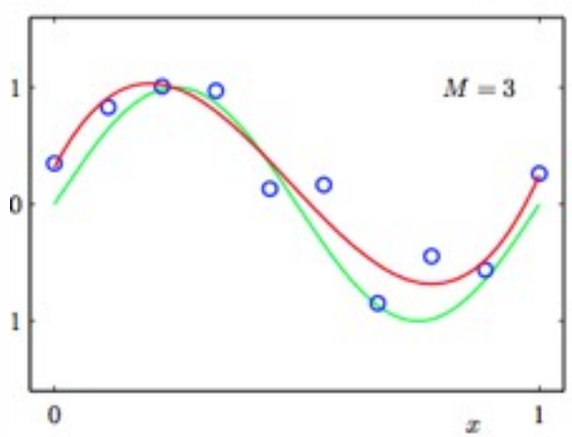
(model complexity)

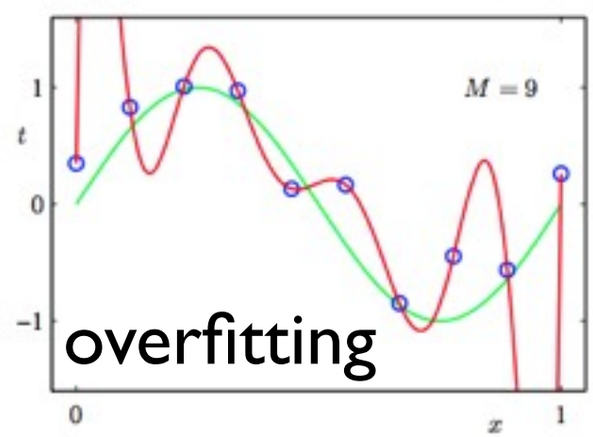


overfitting



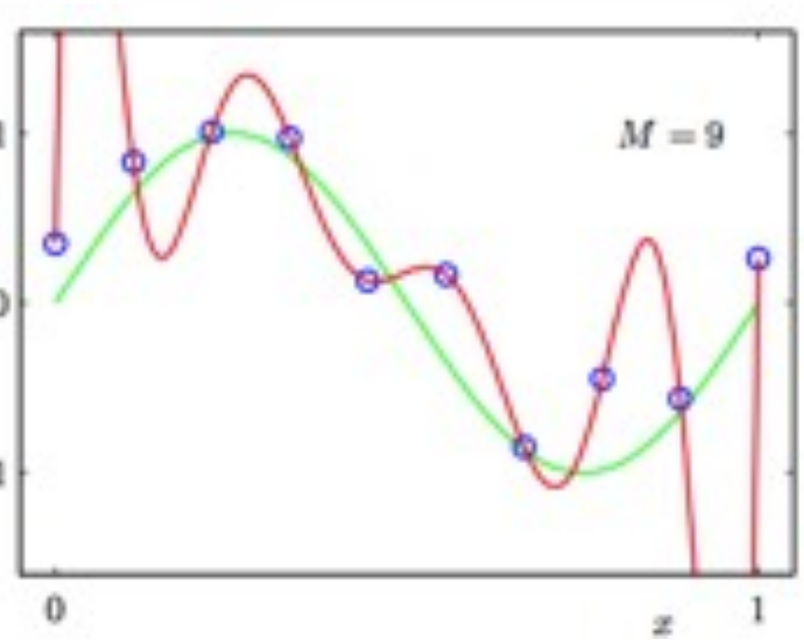


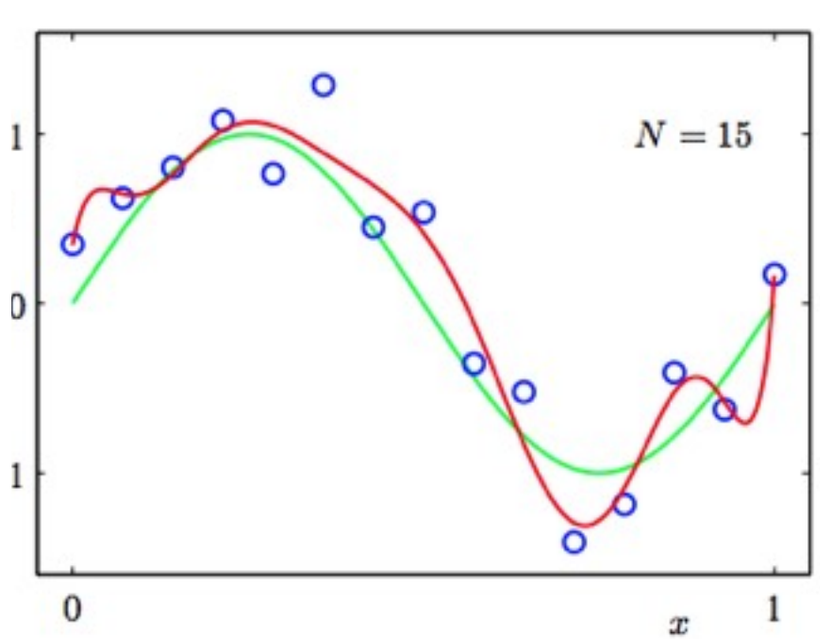


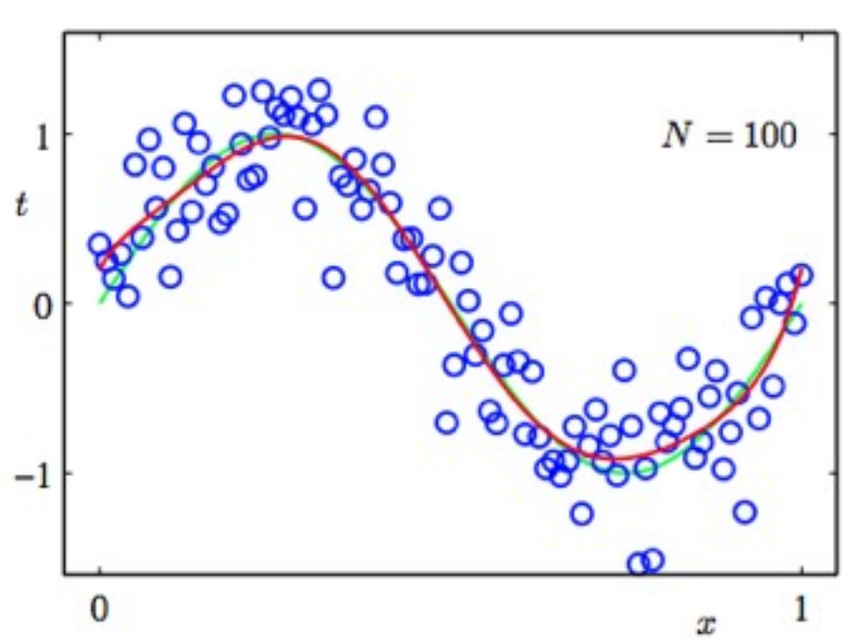


Ways to Prevent Overfitting

- use held-out training data to simulate test data (early stopping)
  - reserve a small subset of training data as "development set" (aka "validation set", "dev set", etc)
- regularization (explicit control of model complexity)
- more training data (overfitting is more likely on small data)
  - assuming same model complexity







polynomials of degree 9

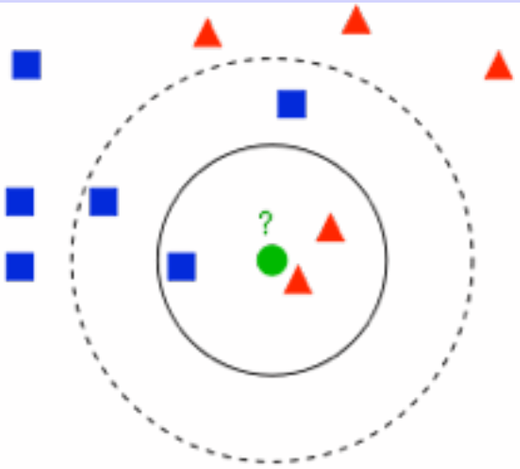
Leave-One-Out Cross-Validation

- what's the best held-out set?
  - random? what if not representative?
  - what if we use every subset in turn?
- leave-one-out cross-validation
  - train on all but the last sample, test on the last; etc.
  - average the validation errors
  - or divide data into N folds, train on folds 1..(N-1), test on fold N; etc.
- this is the best approximation of generalization error

- Part IV: *k*-Nearest Neighbor Classifier

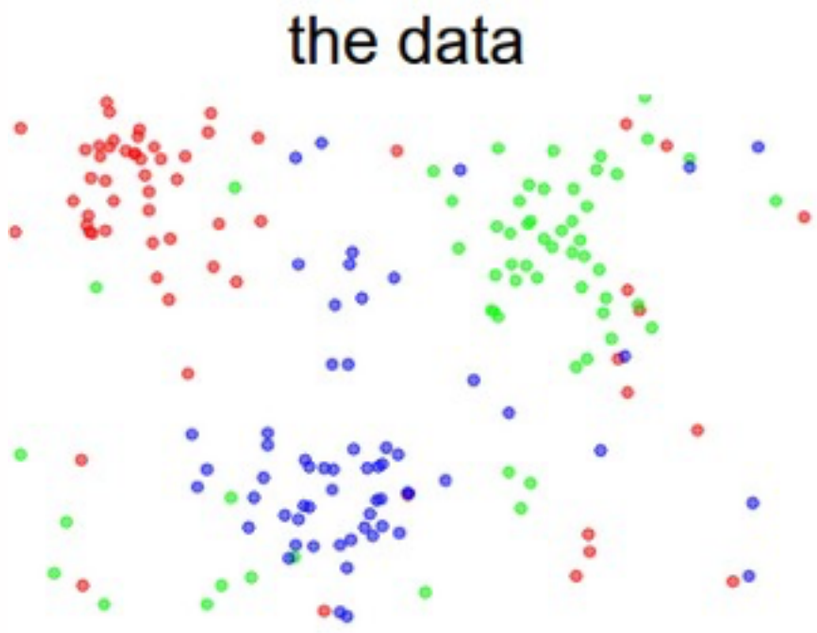
Nearest Neighbor Classifier

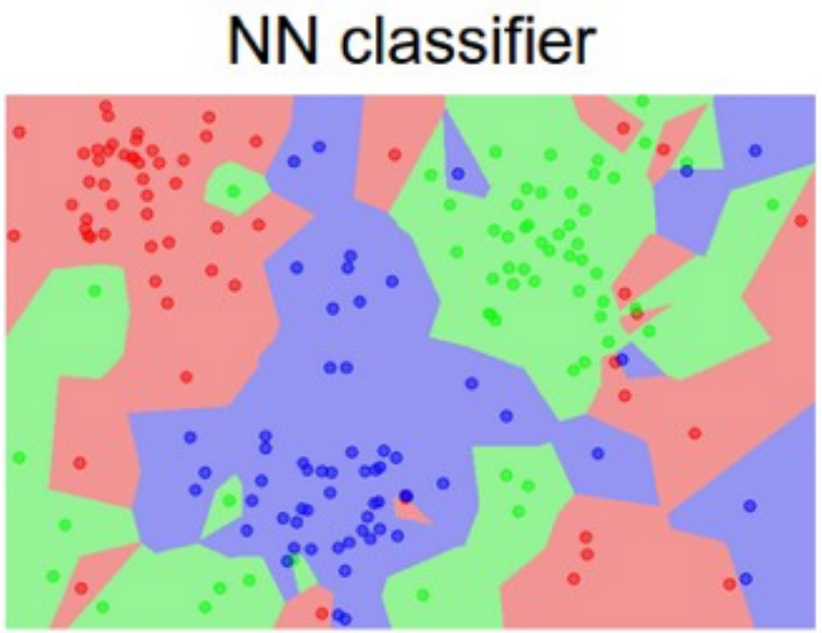
- assign label of test example according to the majority of the closest neighbors in training set
  - extremely simple: no training procedure!
- 1-NN: extreme overfitting; *k*-NN is better
  - as *k* increases, the boundaries become smoother

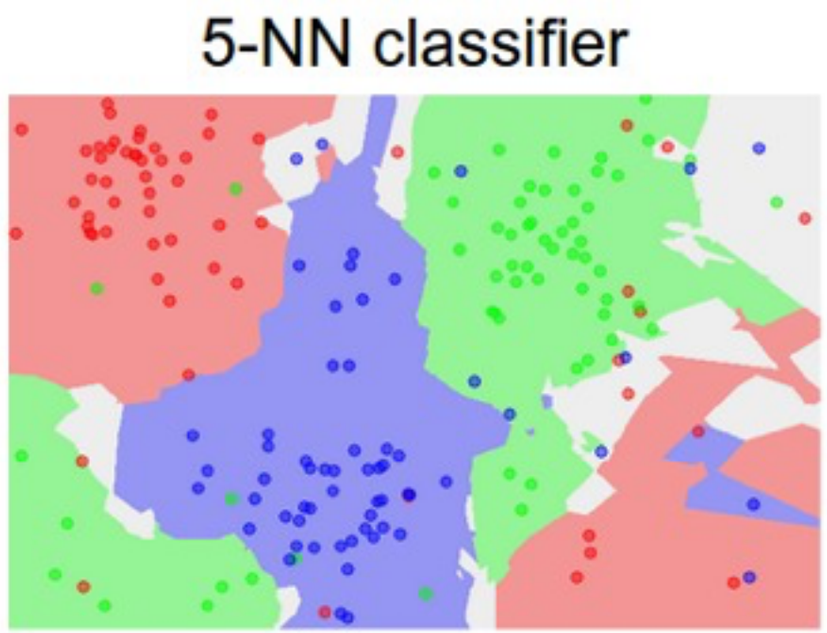


k=1: red
k=3: red
k=5: blue

- *k*=+∞? majority vote (extreme underfitting)







Quiz Question

- what are the leave-one-out cross-validation errors for the following data set, using 1-NN and 3-NN?

(a) Consider the following data set with two real-valued inputs x (i.e. the coordinates of the points) and one binary output y (taking values + or -). We want to use k-nearest neighbours (K-NN) with Euclidean distance to predict y from x.

Calculate the leave-one-out cross-validation error of 1-NN on this data set. That is, for each point in turn, try to predict its label y using the rest of the points, and count up the number of misclassification errors.

Quiz Question

- what are the leave-one-out cross-validation errors for the following data set, using 1-NN and 3-NN?

(a) Consider the following data set with two real-valued inputs x (i.e. the coordinates of the points) and one binary output y (taking values + or -). We want to use k-nearest neighbours (K-NN) with Euclidean distance to predict y from x.

Calculate the leave-one-out cross-validation error of 1-NN on this data set. That is, for each point in turn, try to predict its label y using the rest of the points, and count up the number of misclassification errors.

Ans: 1-NN: 5/10; 3-NN: 1/10