

# COMP 875: Introductions

---

- Name, year, research area/group
- Why are you interested in machine learning and how does it relate to your research?
- What topics would you like to see covered in this course?

# What is Machine Learning?

---

- Using past experiences to improve future performance (on a particular task)
- For a machine, experiences come in the form of data
- What does it mean to improve performance?
  - Learning is guided by a quantitative objective, associated with a particular notion of loss to be minimized (or gain to be maximized)
- Why machine learning?
  - Often it is too difficult to design a set of rules “by hand”
  - Machine learning is about automatically extracting relevant information from data and applying it to analyze new data

# Machine Learning Steps

---

- **Data collection:** Start with *training data* for which we know the correct outcome provided by a “teacher”
- **Representation:** Decide how to encode the input to the learning program
- **Modeling:** Choose a *hypothesis class* – a set of possible explanations for the data
- **Estimation:** Find best hypothesis you can in the chosen class
- **Model selection:** We may reconsider the class of hypotheses given the outcome
  - Each of these steps can make or break the learning outcome

# Learning and Probability

---

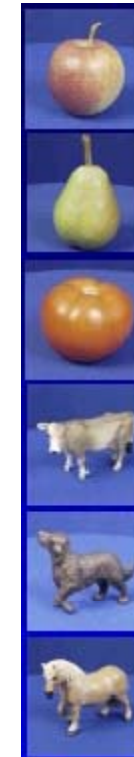
- There are many sources of uncertainty with which learning algorithms must cope:
  - Variability of the data
  - Dataset collection
  - Measurement noise
  - Labeling errors
- Probability and statistics provide an appropriate framework to deal with uncertainty
- Some basic statistical assumptions:
  - Training data is sampled from the “true” underlying data distribution
  - Future test data will be sampled from the same distribution

# Example of a learning problem

---



Given: training images and their categories



What are the categories of these test images?

- Possible representation: image of size  $n \times n$  pixels  $\rightarrow$  vector of length  $n^2$  (or  $3n^2$  if color)

# The Importance of Representation

---

- Dimensionality
- Beyond vectors: complex or heterogeneous input objects
  - Web pages
  - Program traces
  - Images with captions or metadata
  - Video with sound
  - Proteins
- Feature extraction and feature selection
  - What measurements/information about the input objects are the most useful for solving the given problem?
- Successful representation requires domain knowledge!
  - If we could find the “ideal” feature representation, we would not even need learning!

# Types of learning problems

---

- Supervised
  - Classification
  - Regression
- Unsupervised
- Semi-supervised
- Reinforcement learning
- Active learning
- ....

# Supervised learning

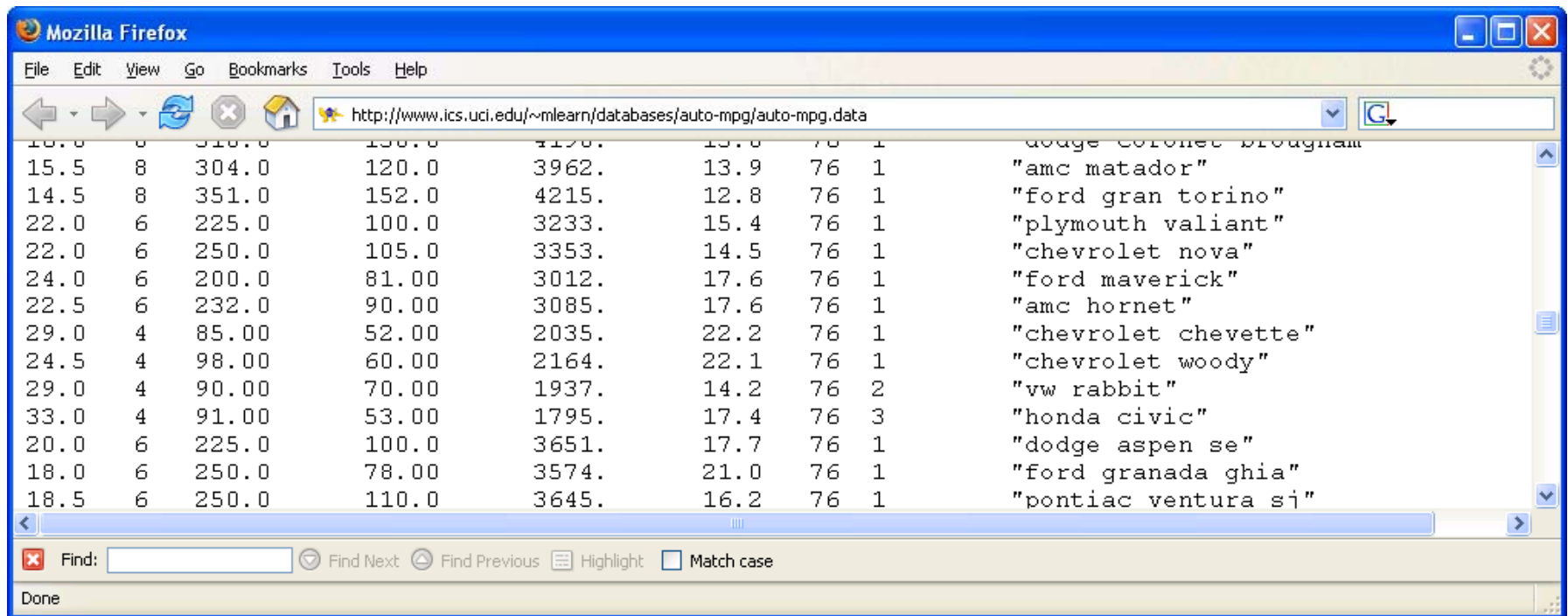
---

- Given training examples of inputs and corresponding outputs, produce the “correct” outputs for new inputs
- Two main scenarios:
  - **Classification:** outputs are discrete variables (category labels). Learn a decision boundary that separates one class from the other
  - **Regression:** also known as “curve fitting” or “function approximation.” Learn a continuous input-output mapping from examples (possibly noisy)

# Regression: example 1

---

- Suppose we want to predict gas mileage of a car based on some characteristics: number of cylinders or doors, weight, horsepower, year etc.



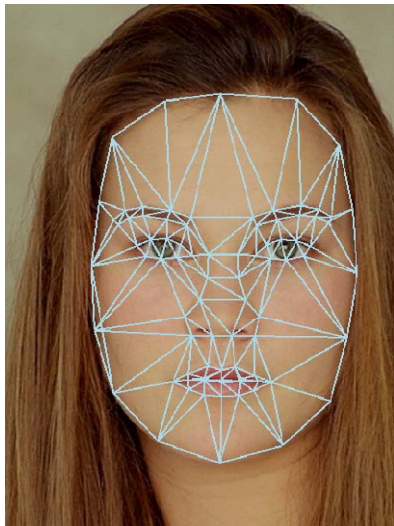
The screenshot shows a Mozilla Firefox browser window displaying a text file from the URL <http://www.ics.uci.edu/~mllearn/databases/auto-mpg/auto-mpg.data>. The data is presented as a table with 11 columns and 14 rows of car records. The columns represent: mpg (miles per gallon), num\_cyl (number of cylinders), weight (in pounds), horsepower, displacement (in cubic inches), year, num\_doors, and car name.

mpg	num_cyl	weight	horsepower	displacement	year	num_doors	car name
18.0	8	318.0	150.0	4176.0	1970	1	dodge coronet
15.5	8	304.0	120.0	3962.0	1971	1	"amc matador"
14.5	8	351.0	152.0	4215.0	1971	1	"ford gran torino"
22.0	6	225.0	100.0	3233.0	1971	1	"plymouth valiant"
22.0	6	250.0	105.0	3353.0	1971	1	"chevrolet nova"
24.0	6	200.0	81.00	3012.0	1971	1	"ford maverick"
22.5	6	232.0	90.00	3085.0	1971	1	"amc hornet"
29.0	4	85.00	52.00	2035.0	1971	1	"chevrolet chevette"
24.5	4	98.00	60.00	2164.0	1971	1	"chevrolet woody"
29.0	4	90.00	70.00	1937.0	1972	2	"vw rabbit"
33.0	4	91.00	53.00	1795.0	1974	3	"honda civic"
20.0	6	225.0	100.0	3651.0	1976	1	"dodge aspen se"
18.0	6	250.0	78.00	3574.0	1976	1	"ford granada ghia"
18.5	6	250.0	110.0	3645.0	1976	1	"pontiac ventura sj"

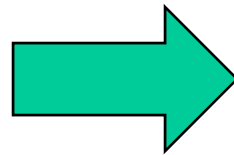
# Regression: example 2

---

- Training set: faces (represented as vectors of distances between keypoints) together with experimentally obtained attractiveness rankings
- Learn: function to reproduce attractiveness ranking based on training inputs and outputs



Vector of distances  $\mathbf{v}$



Attractiveness score  $f(\mathbf{v})$

T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski, Data-driven enhancement of facial attractiveness, SIGGRAPH 2008

# Regression: example 3

---

- Input: scalar (attractiveness score)
- Output: vector-valued object (face)

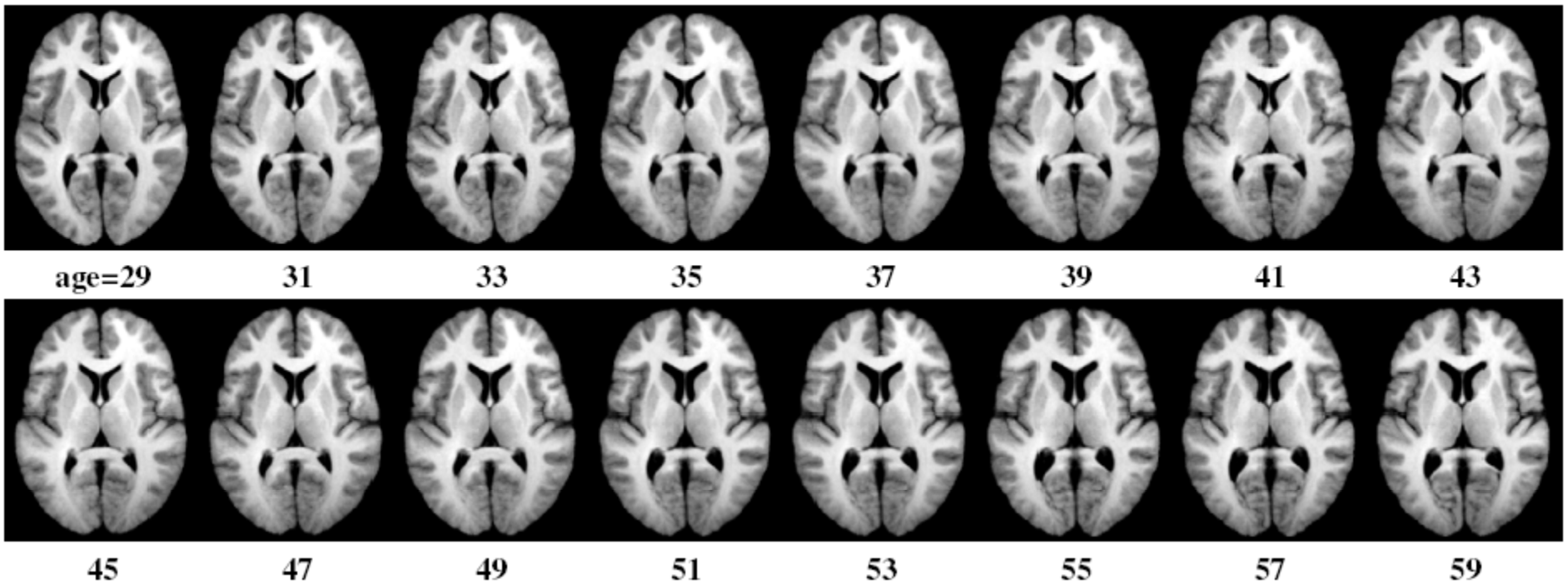


B. Davis and S. Lazebnik, "Analysis of Human Attractiveness Using Manifold Kernel Regression," ICIP 2008

# Regression: example 4

---

- Input: scalar (age)
- Output: vector-valued object (3D brain image)



B. C. Davis, P. T. Fletcher, E. Bullitt and S. Joshi, "Population Shape Regression From Random Design Data", ICCV, 2007.

# Structured Prediction

---



Image



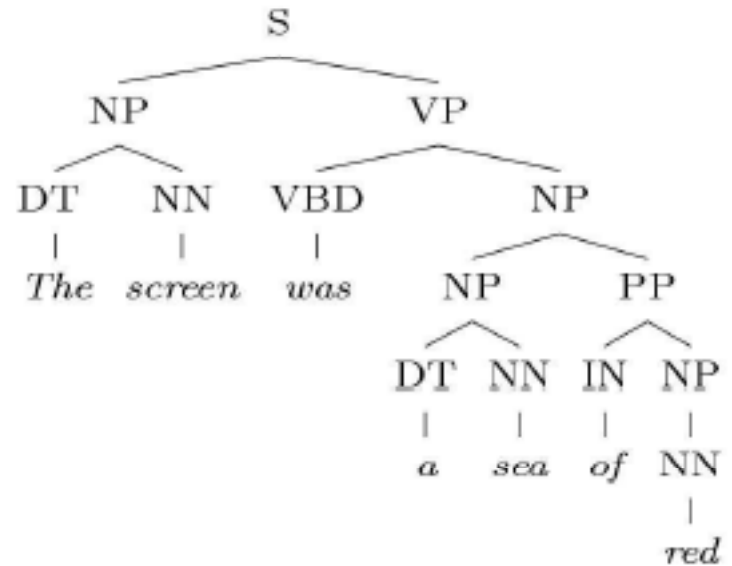
**brace**

Word

# Structured Prediction

---

The screen was  
a sea of red



Sentence

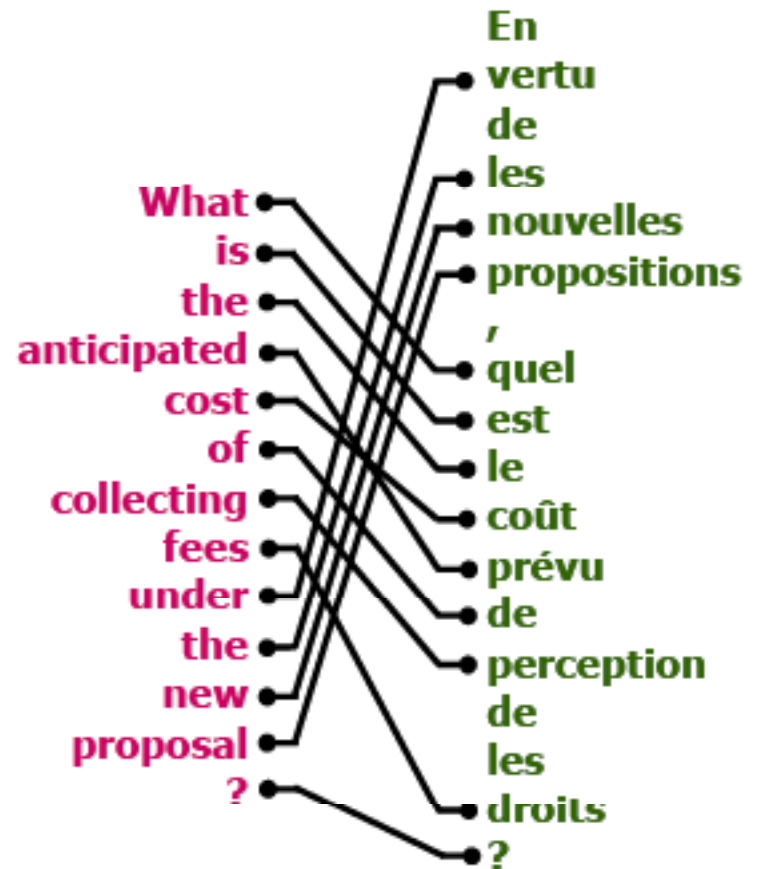
Parse tree

# Structured Prediction

---

**What is the anticipated cost of collecting fees under the new proposal?**

**En vertu des nouvelles propositions, quel est le coût prévu de perception des droits?**



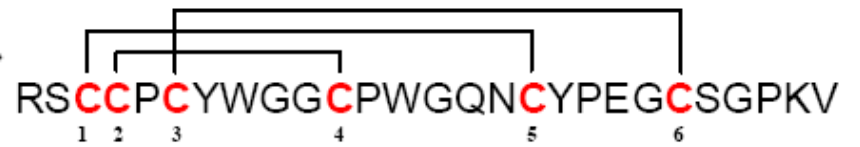
Sentence in two languages

Word alignment

# Structured Prediction

---

RSCCPCYWGGCPW  
GQNCYPEGCSGPKV



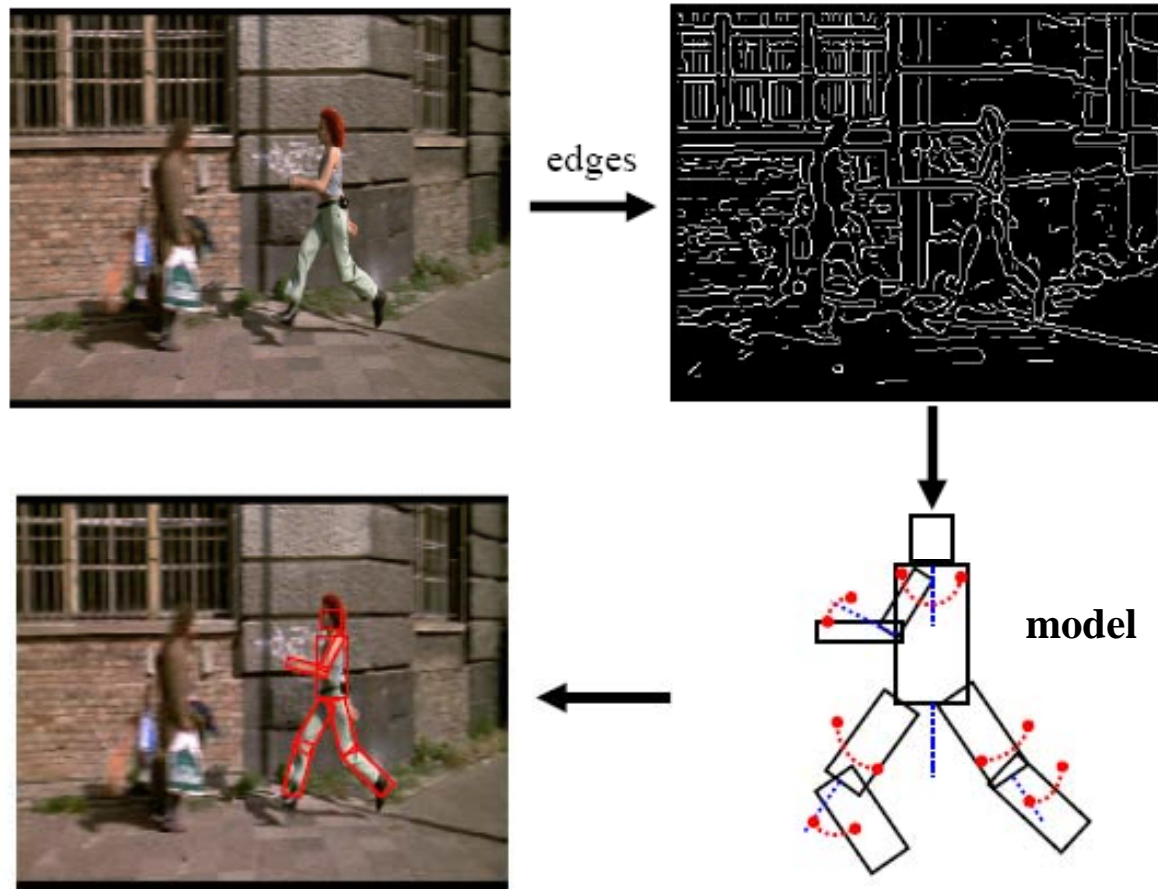
Amino-acid sequence

Bond structure

# Structured Prediction

---

- Many image-based inference tasks can loosely be thought of as “structured prediction”
- *Data association* problem



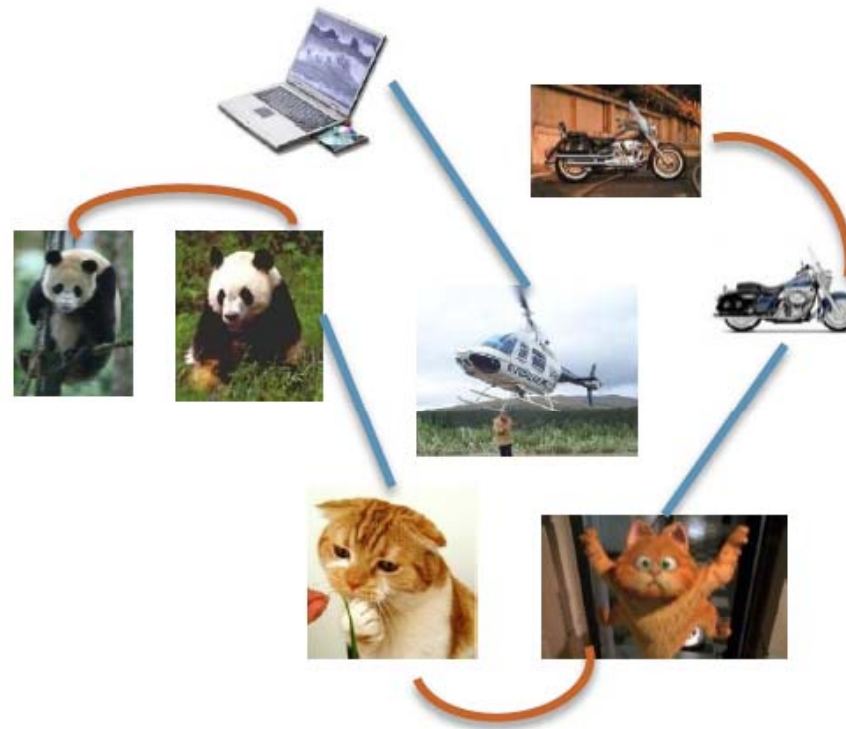
Source: D. Ramanan

# Other supervised learning scenarios

---

- Learning similarity functions from relations between multiple input objects

Pairwise constraints

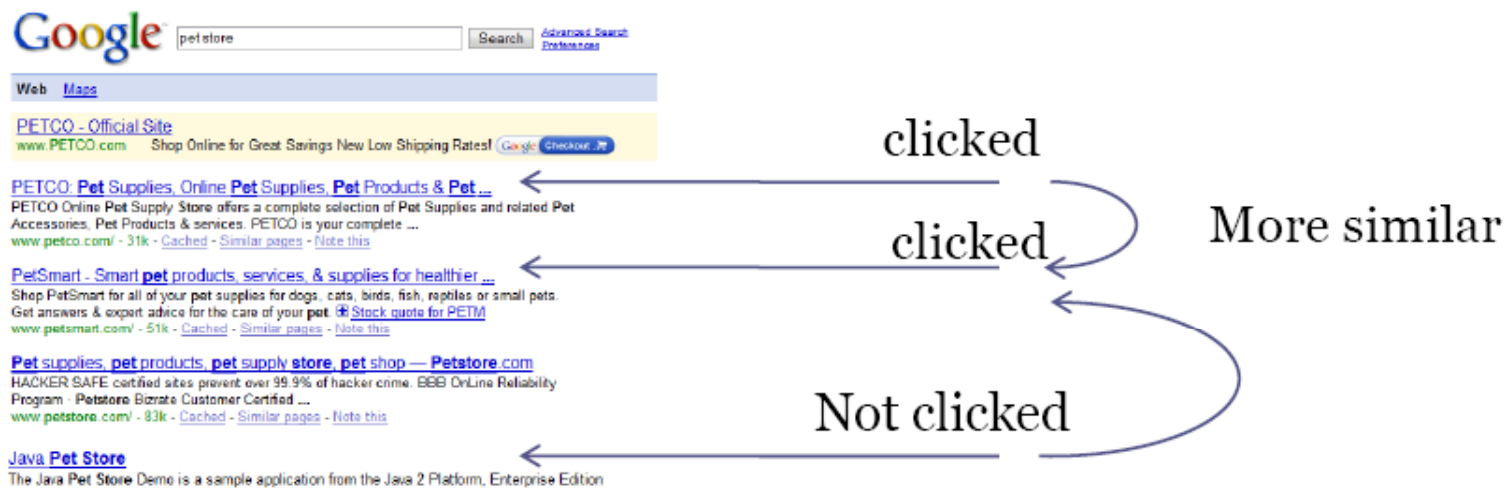


Red line: equivalence constraints  
Blue line: in-equivalence constraints

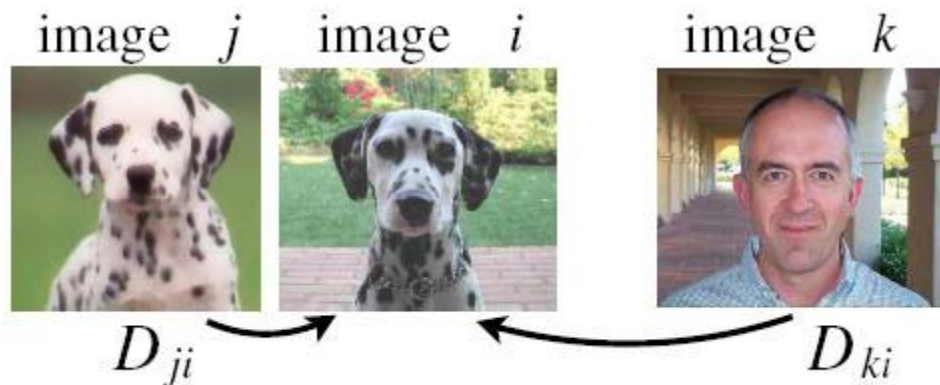
# Other supervised learning scenarios

- Learning similarity functions from relations between multiple input objects

## Triplet constraints



A screenshot of a Google search for "petstore". The search results are annotated with arrows and text to illustrate triplet constraints. The top result is "PETCO - Official Site" (www.PETCO.com), which is annotated with "clicked" and "More similar". The second result is "PetSmart - Smart pet products, services, & supplies for healthier ..." (www.petsmart.com), also annotated with "clicked" and "More similar". The third result is "Pet supplies, pet products, pet supply store, pet shop — Petstore.com" (www.petstore.com), annotated with "Not clicked". The fourth result is "Java Pet Store" (The Java Pet Store Demo is a sample application from the Java 2 Platform, Enterprise Edition), also annotated with "Not clicked".



Source: X. Sui, K. Grauman

# Unsupervised Learning

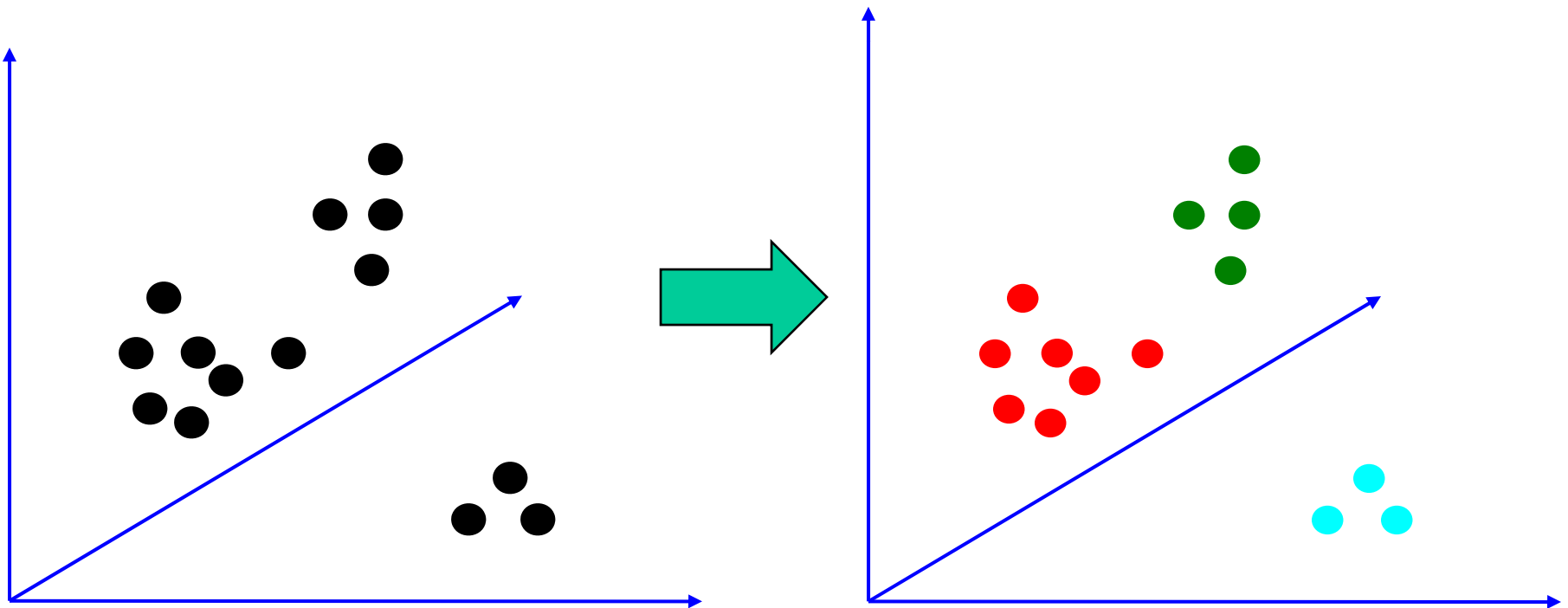
---

- Given only *unlabeled* data as input, learn some sort of structure
- The objective is often more vague or subjective than in supervised learning. This is more of an exploratory/descriptive data analysis

# Unsupervised Learning

---

- **Clustering**
  - Discover groups of “similar” data points

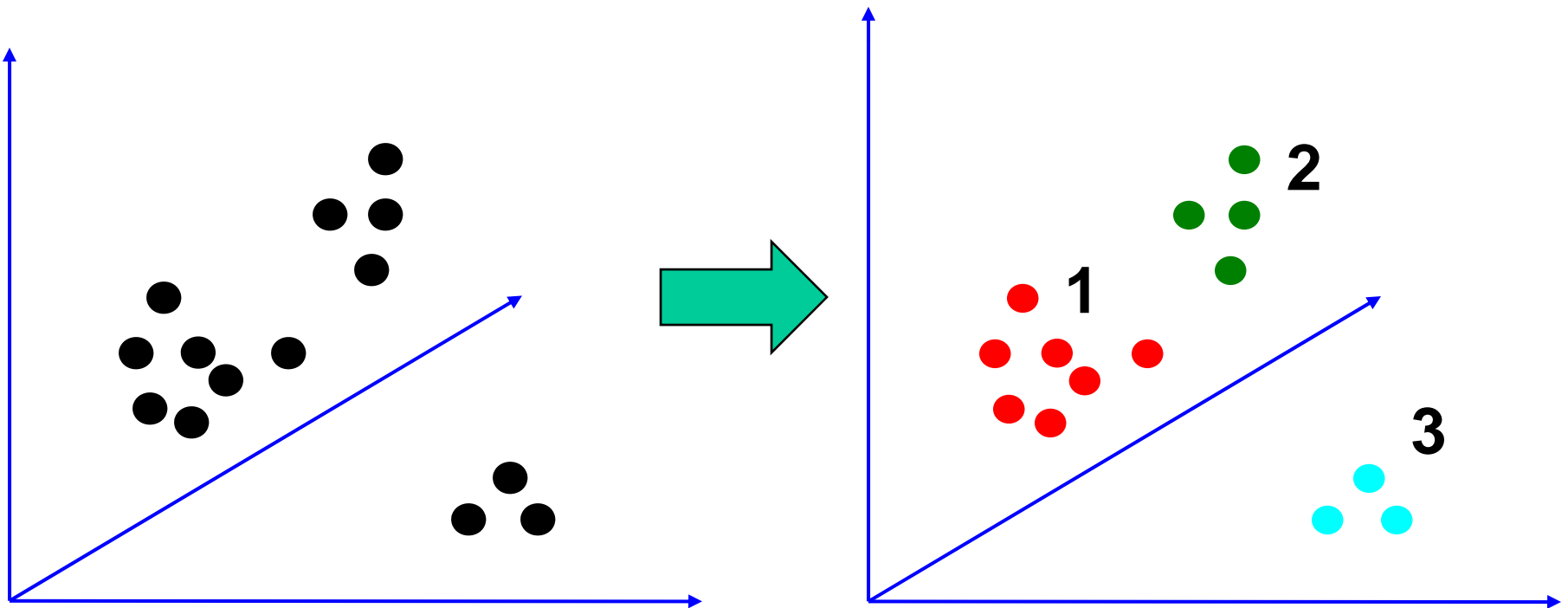


# Unsupervised Learning

---

- **Quantization**

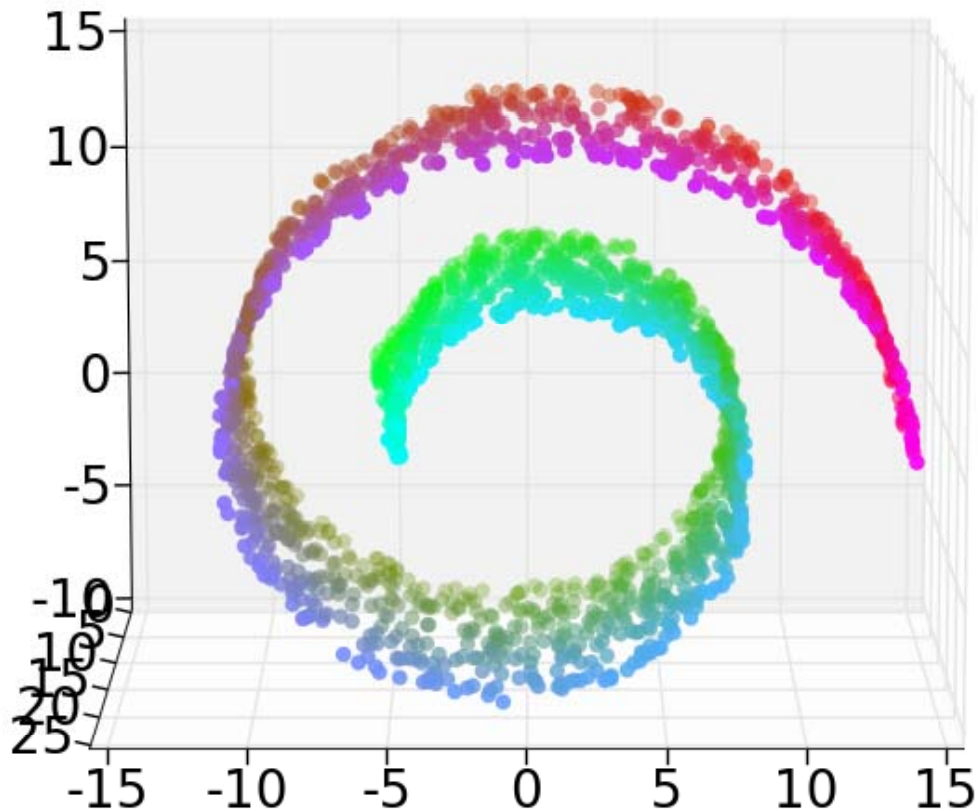
- Map a continuous input to a discrete (more compact) output



# Unsupervised Learning

---

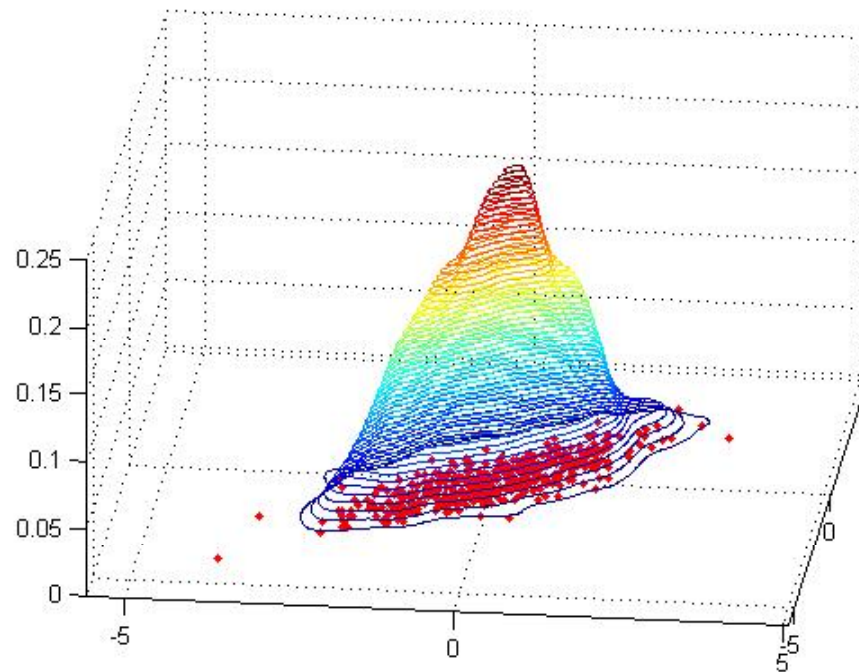
- **Dimensionality reduction, manifold learning**
  - Discover a lower-dimensional surface on which the data lives



# Unsupervised Learning

---

- **Density estimation**
  - Find a function that approximates the probability density of the data (i.e., value of the function is high for “typical” points and low for “atypical” points)
  - Can be used for **anomaly detection**



# Other types of learning

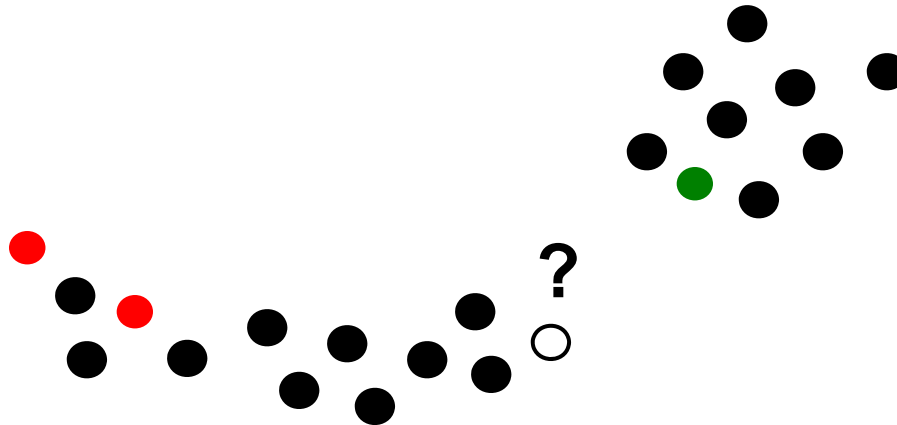
---

- **Semi-supervised learning:** lots of data is available, but only small portion is labeled (e.g. since labeling is expensive)

# Other types of learning

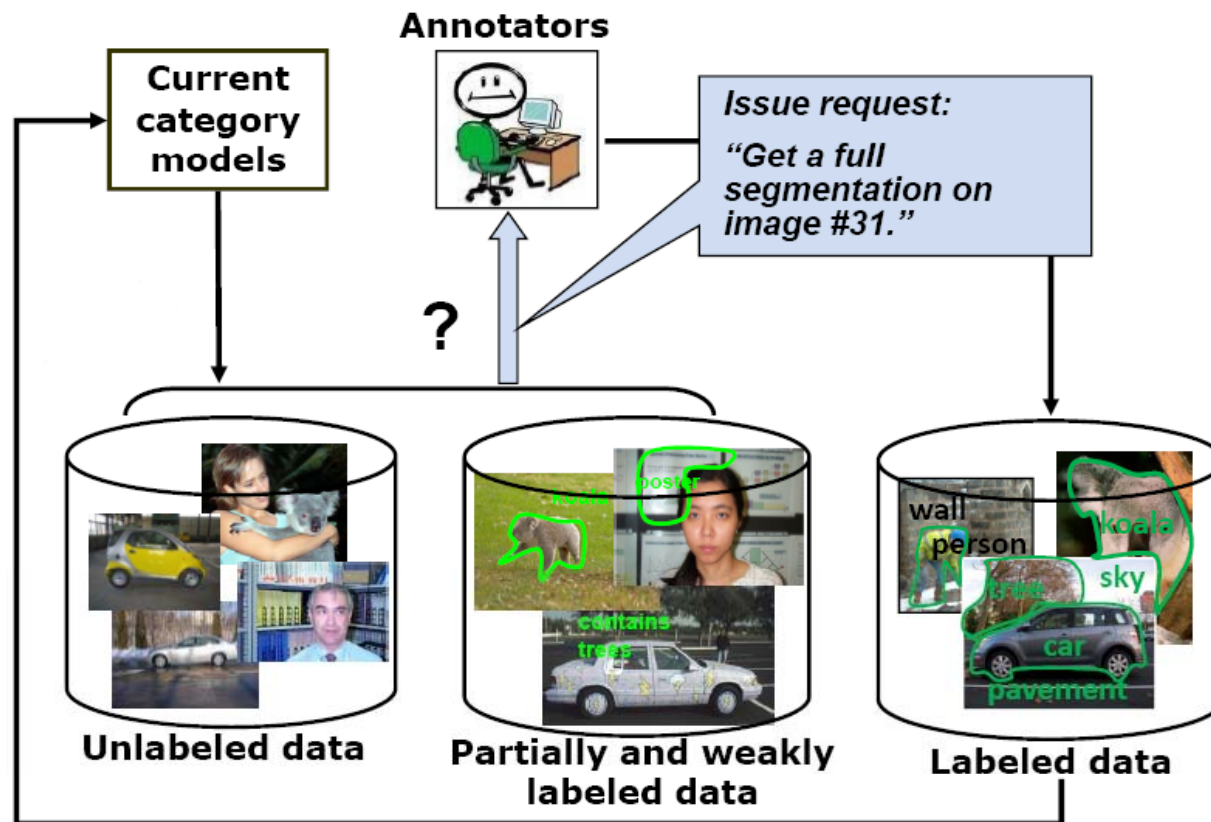
---

- **Semi-supervised learning:** lots of data is available, but only small portion is labeled (e.g. since labeling is expensive)
  - Why is learning from labeled and unlabeled data better than learning from labeled data alone?



# Other types of learning

- **Active learning:** the learning algorithm can choose its own training examples, or ask a “teacher” for an answer on selected inputs



S. Vijayanarasimhan and K. Grauman, “Cost-Sensitive Active Visual Category Learning,” 2009

# Other types of learning

---

- **Reinforcement learning:** an agent takes inputs from the environment, and takes actions that affect the environment. Occasionally, the agent gets a scalar reward or punishment. The goal is to learn to produce action sequences that maximize the expected reward (e.g. driving a robot without bumping into obstacles)
- **Apprenticeship learning:** learning from demonstrations when the reward function is initially unknown
  - Autonomous helicopter flight: Pieter Abbeel  
<http://heli.stanford.edu/>

# Generalization

---

- The ultimate goal is to do as well as possible on new, unseen data (a *test set*), but we only have access to labels (“ground truth”) for the training set
- What makes generalization possible?
- **Inductive bias:** set of assumptions a learner uses to predict the target value for previously unseen inputs
  - This is the same as modeling or choosing a target hypothesis class
- Types of inductive bias
  - Occam’s razor
  - Similarity/continuity bias: similar inputs should have similar outputs
  - ...

# Achieving good generalization

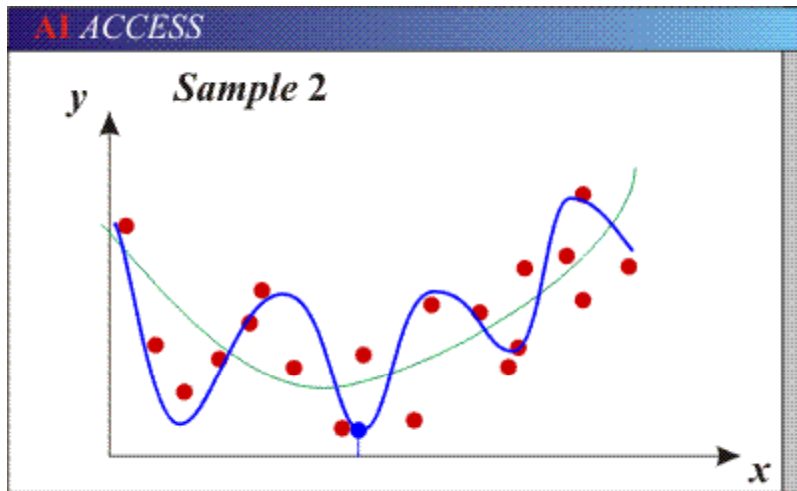
---

- Consideration 1: **Bias**
  - How well does your model fit the observed data?
  - It may be a good idea to accept some fitting error, because it may be due to noise or other “accidental” characteristics of one particular training set
- Consideration 2: **Variance**
  - How robust is the model to the selection of a particular training set?
  - To put it differently, if we learn models on two different training sets, how consistent will the models be?

# Bias/variance tradeoff

---

- Models with too many parameters may fit the training data well (**low bias**), but are sensitive to choice of training set (**high variance**)

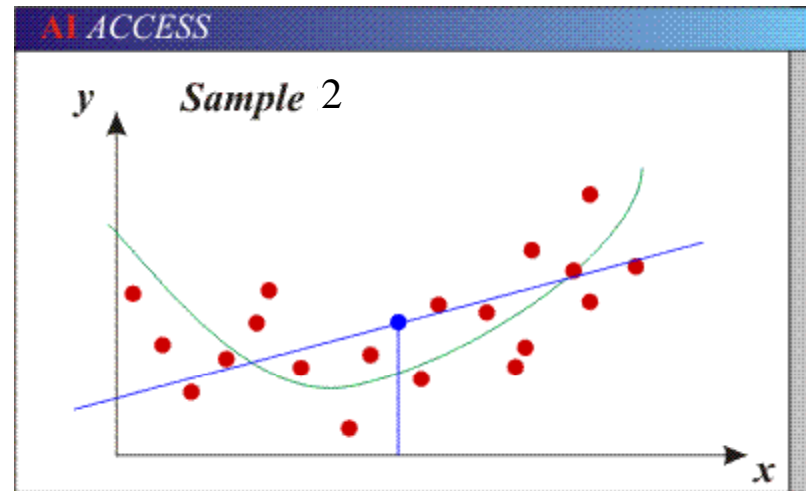
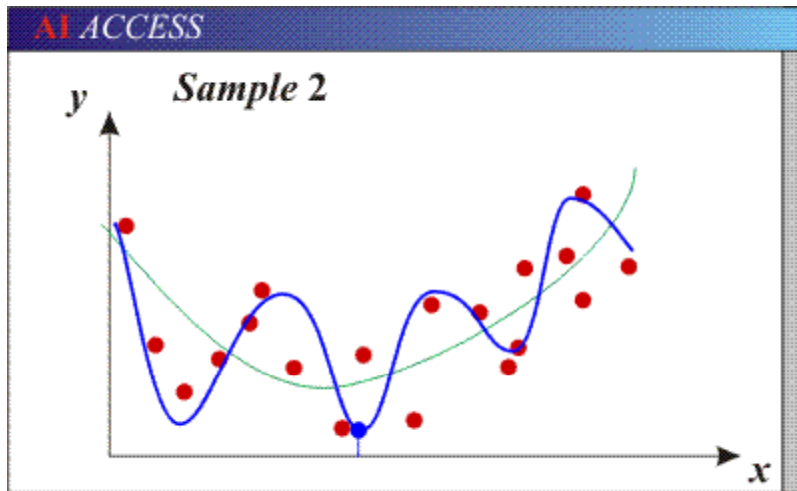


# Bias/variance tradeoff

---

- Models with too many parameters may fit the training data well (**low bias**), but are sensitive to choice of training set (**high variance**)

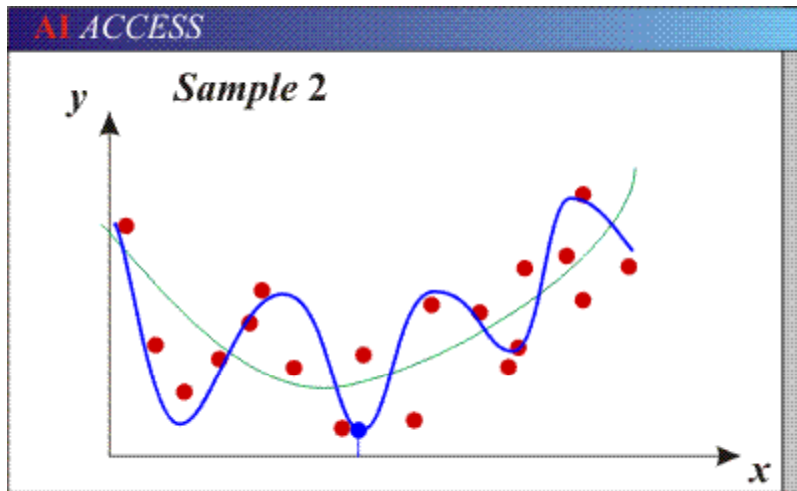
- Models with too few parameters may not fit the data well (**high bias**) but are consistent across different training sets (**low variance**)



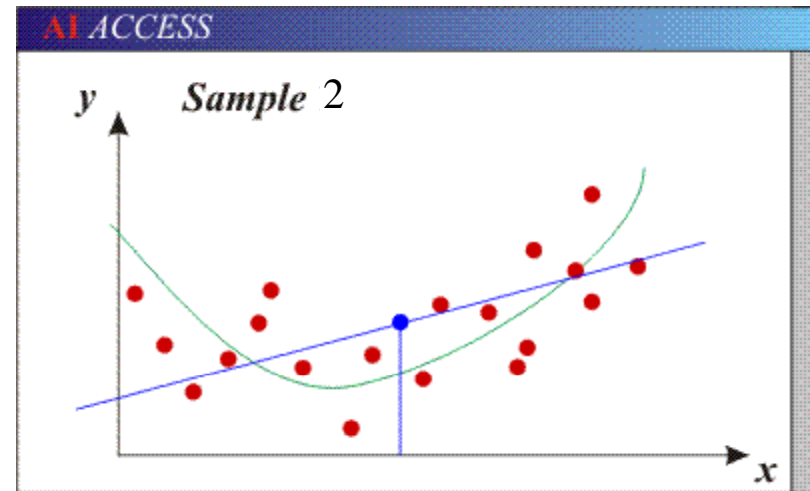
# Bias/variance tradeoff

---

- Models with too many parameters may fit the training data well (**low bias**), but are sensitive to choice of training set (**high variance**)
- Generalization error is due to **overfitting**



- Models with too few parameters may not fit the data well (**high bias**) but are consistent across different training sets (**low variance**)
- Generalization error is due to **underfitting**



# Underfitting and overfitting

---

- How to recognize underfitting?
  - High training error and high test error
- How to deal with underfitting?
  - Find a more complex model
  
- How to recognize overfitting?
  - Low training error, but high test error
- How to deal with overfitting?
  - Get more training data
  - Decrease the number of parameters in your model
  - Regularization: penalize certain parts of the parameter space or introduce additional constraints to deal with a potentially ill-posed problem

# Methodology

---

- Distinction between training and testing is crucial
  - Correct performance on training set is just memorization!
- Strictly speaking, the researcher should ***never look at the test data*** when designing the system
  - Generalization performance should be evaluated on a *hold-out* or *validation* set
  - Raises some troubling issues for learning “benchmarks”

# Next time

---

- The math begins...
  - Guest lecturer: Max Raginsky (Duke EE)
- **Reading lists due to me by email by the end of next Thursday, September 3<sup>rd</sup>**
  - A couple of sentences describing your topic
  - A list of ~3 papers (doesn't have to be final)
  - Date constraints/preferences
  - If you have more than one idea, send them all (will help with conflict resolution)