DSE 220: Machine Learning

Overview

Outline

- Introduction
  - Myself
  - TAs
  - Course
  - Machine Learning
  - ***Hands-On:*** *Intro to scikit-learn*
- Nonparametric Methods
  - Decision Trees
  - ***Hands-On***
  - ***Self – practice***
  - kNN
  - ***Hands-On***
  - ***Self – practice***
- Parametric Methods
  - Generative Models
  - ***Hands-On***
  - ***Self – practice***

Myself

- Volkan Vural (vvural@ucsd.edu)
- Ph.D. in Machine Learning
- Research in Classification/ Computer Aided Diagnosis
- 6+ years of experience in Investment Technologies
- Boston College / MBA classes
  - Forecasting in Business and Economics
  - Machine Learning for Business Intelligence

TAs

- Ismail Ocak (iocak@eng.ucsd.edu)

- Ajit Kumar(akumar@eng.ucsd.edu)

- Grad Students at UCSD

Course

- Machine Learning
  - Important concepts & methods
  - Both theory and applications
- Requires introductory level statistics background
- Practical and application oriented
- Python will be used for implementations

- Important links
  - Website: https://mas-dse.github.io/DSE220/
  - GitHub: https://github.com/mas-dse/DSE220SP20
  - Piazza: <TBA>
  - Gradescope: https://gradescope.com (Entry code : **9KGV38**)
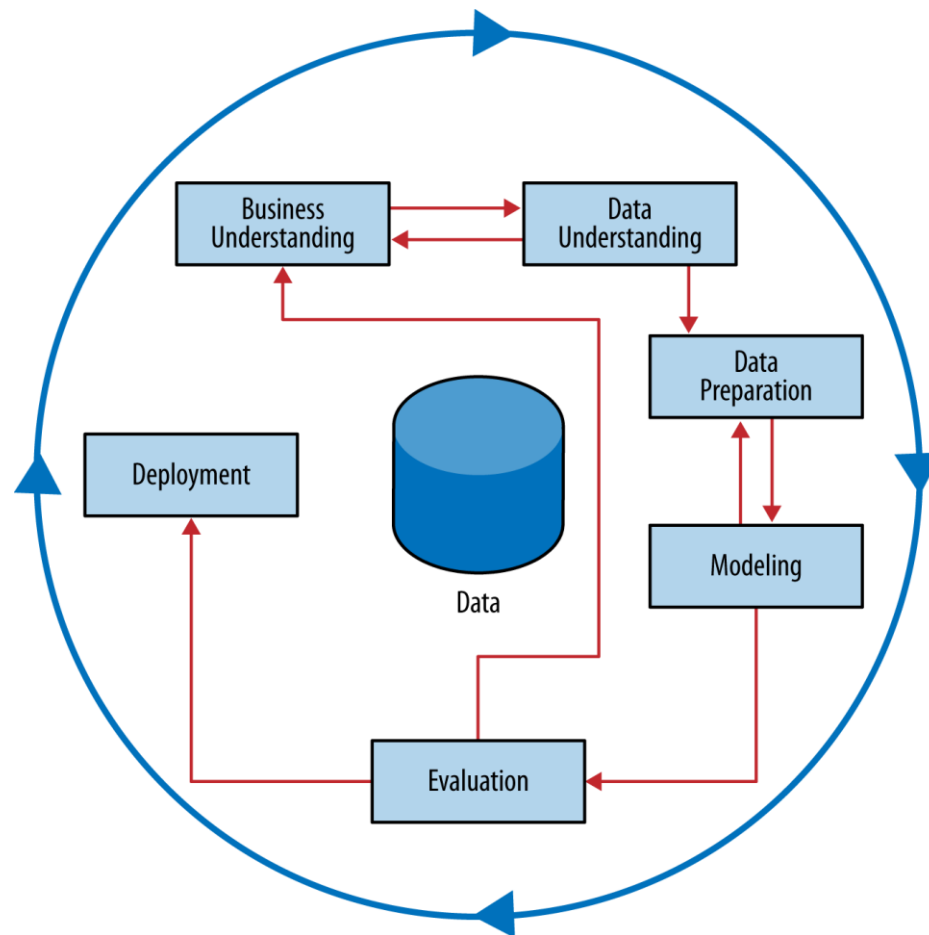
Machine Learning
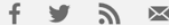
Target has got you in its aim
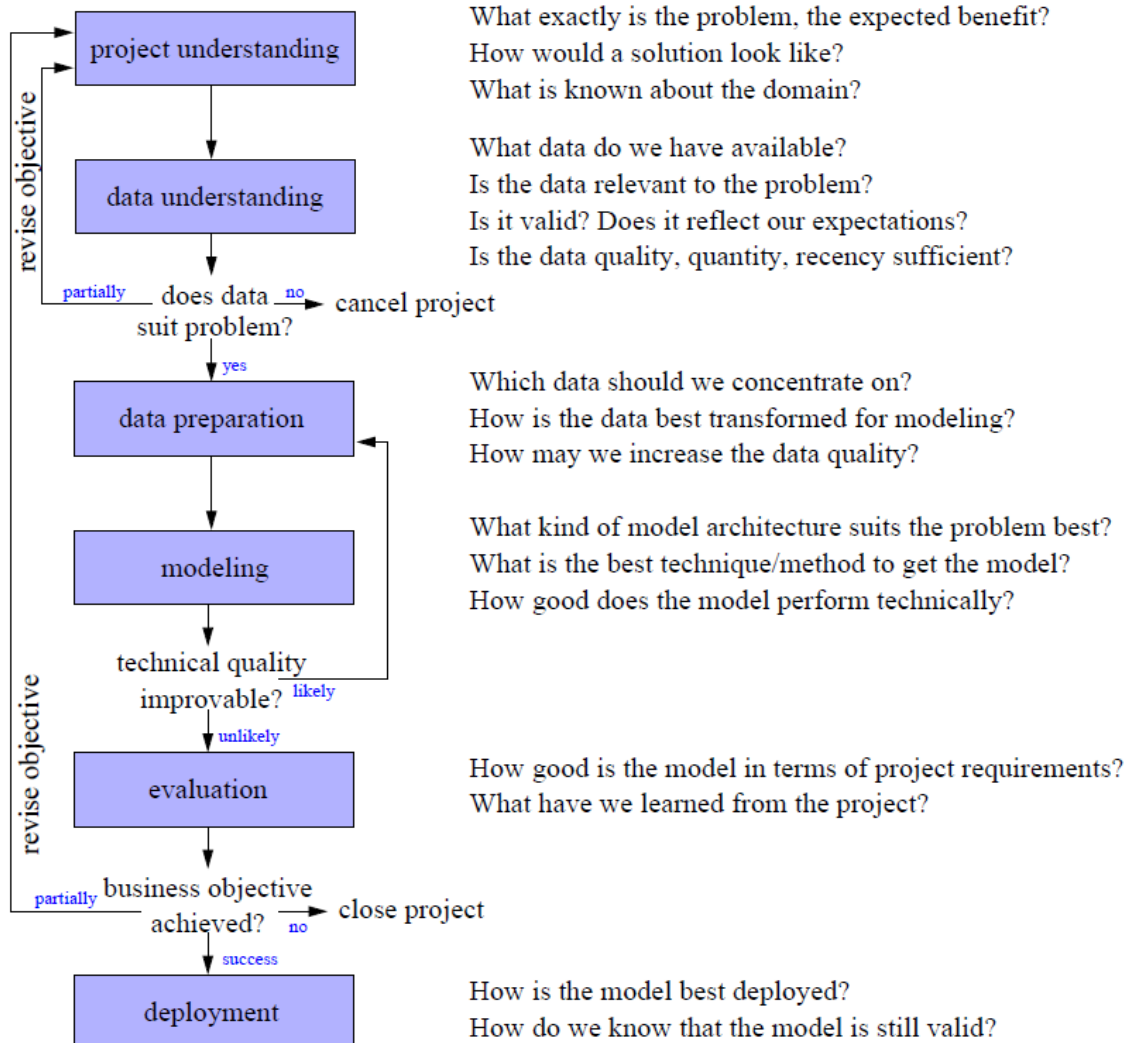
Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are studying those details to figure out what you like, what you need, and which coupons are most likely to make you happy. Target, for example, has figured out how to data-mine its way into your

CRoss-Industry Standard Process for Data Mining (CRISP-DM)

CRISP-DM

What exactly is the problem, the expected benefit?
How would a solution look like?
What is known about the domain?

What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?

What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How good does the model perform technically?

How good is the model in terms of project requirements?
What have we learned from the project?

How is the model best deployed?
How do we know that the model is still valid?

Cross
Industry
Standard
Process for
Data **M**ining

Iteration as
a rule

Process of
data
exploration

Intro to Machine Learning

- Goals

    - Learn objectives (patterns, decision functions, mapping) from information at hand (training data)

    - Make accurate predictions on test subjects (test data)
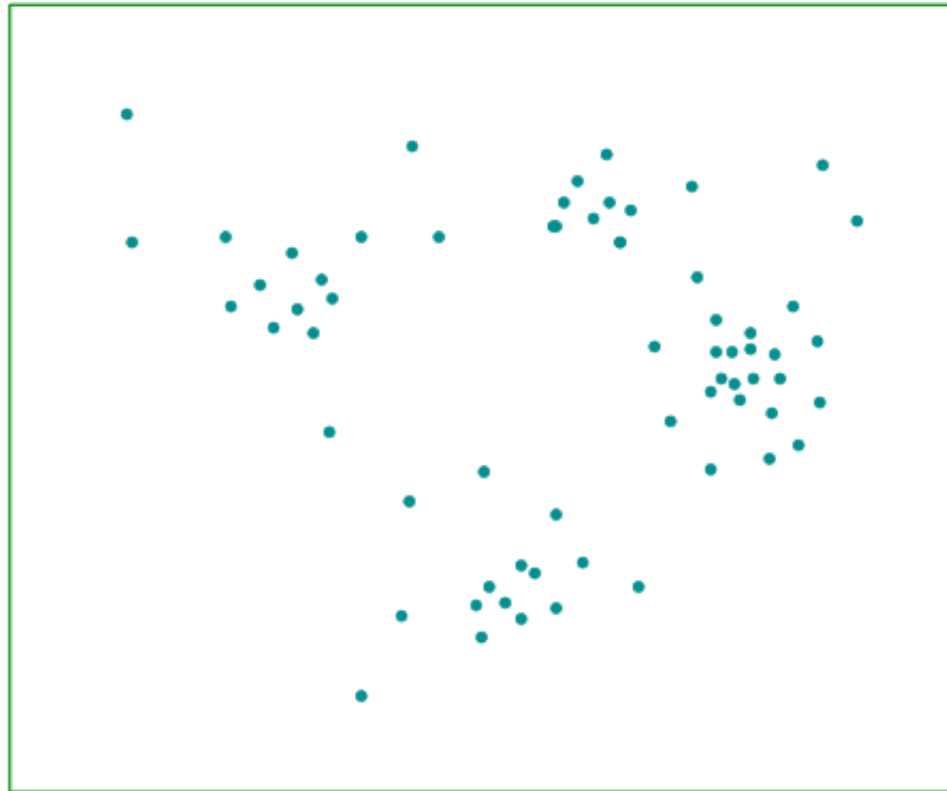
Intro to Machine Learning

- Machine Learning Tasks

    - Supervised Learning
        - Each sample in training data is labeled by a field expert

    - Unsupervised Learning
        - Learning without the supervision of a field expert

    - Semi-supervised Learning
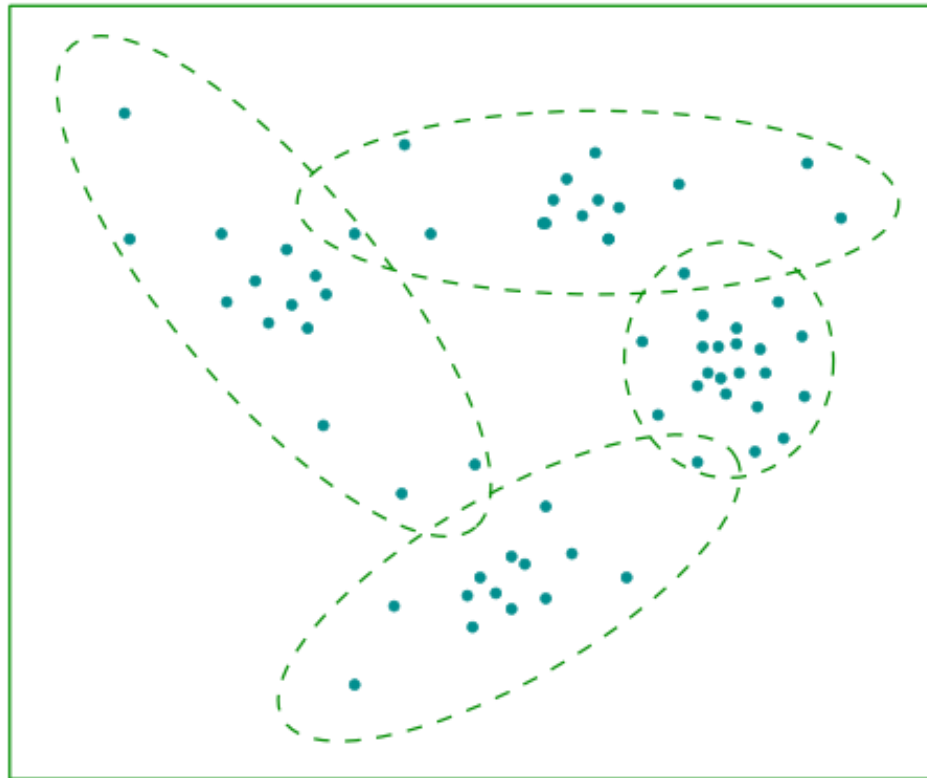   - A portion of the training data is labeled

Supervised Learning

Domain Expert

Class 1: Male

Class 2: Female
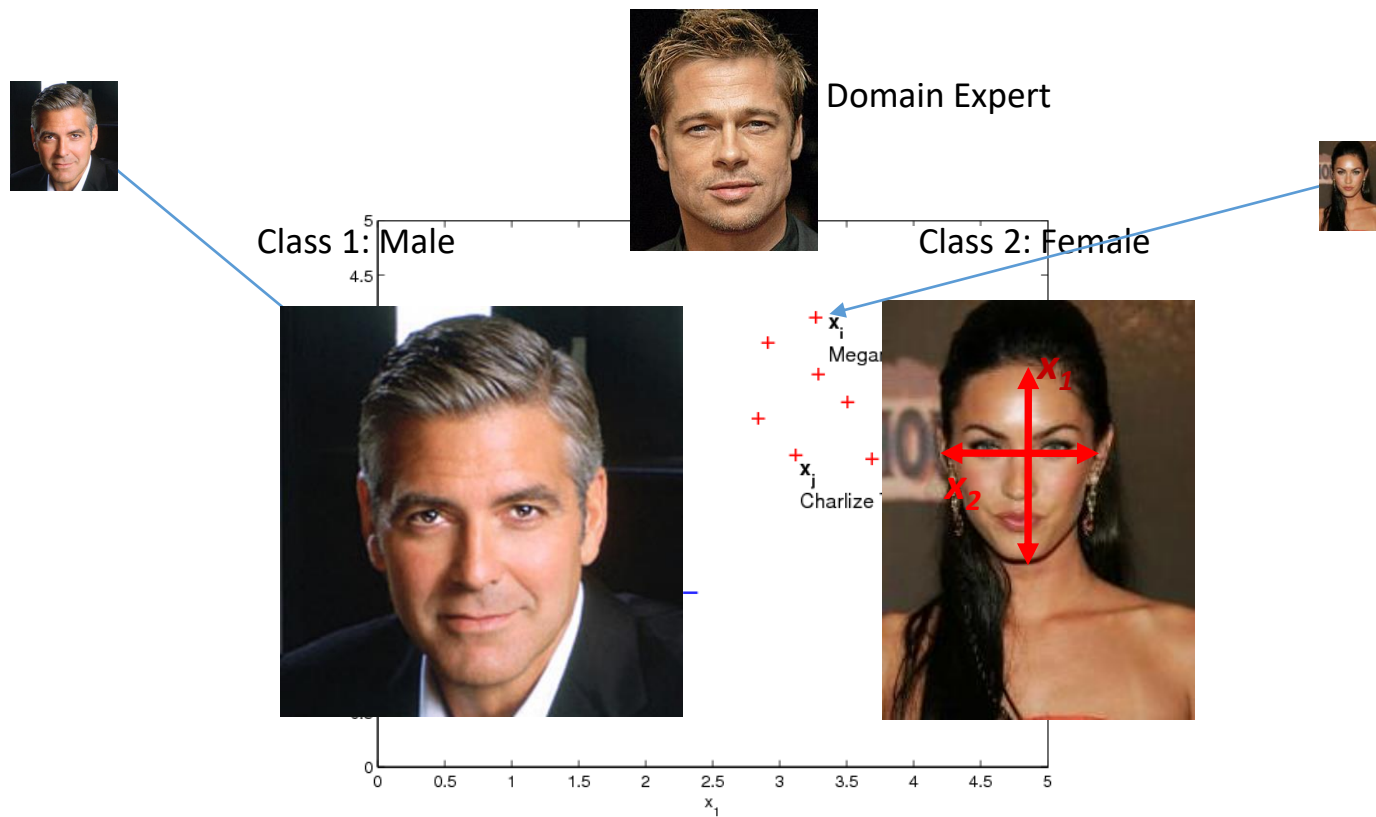
Unsupervised Learning

Unsupervised Learning
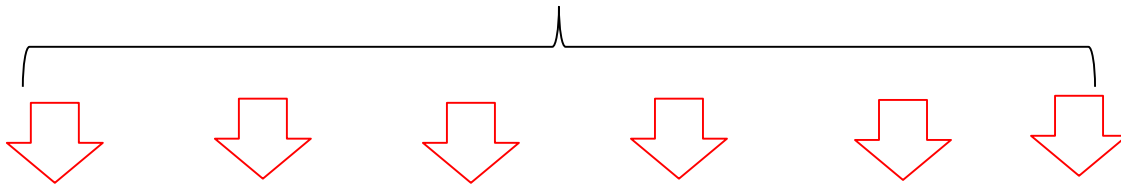
Data Terminology

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

Data Terminology

Variables
(columns)

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

Data Terminology

Target variable
Label
Dependent variable
Output space
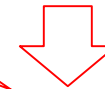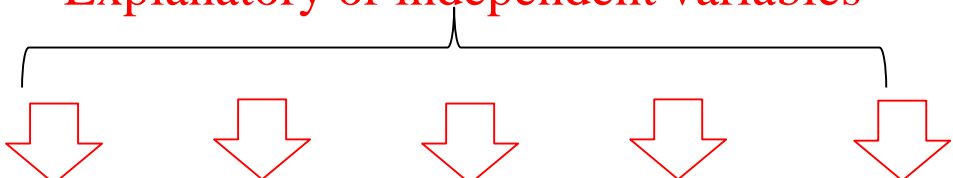
Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
….	….	….	….	….	….

Data Terminology

Attributes
Features
Explanatory or independent variables

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

Data Terminology

Records
(Data) Instances

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

Data Terminology

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

(17824, 49, M, 12000, -3000) is a feature vector

Data Terminology

Target variable = Income
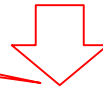
Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

Data Terminology

NO Target variable

Person ID	Age	Gender	Income	Balance	Mortgage payment
123213	32	F	25000	32000	Y
17824	49	M	12000	-3000	N
232897	60	F	8000	1000	Y
288822	28	M	9000	3000	Y
....	....	....	....	....	....

Machine learning versus Algorithms

In both fields, the goal is to develop

procedures that exhibit a desired input-output behavior.

- **Algorithms**: the input-output mapping can be precisely defined.
  Input: Graph G .
  Output: Minimum Spanning Tree of G

- **Machine learning**: the mapping cannot easily be made precise.
  Input: Picture of an animal.
  Output: Name of the animal.

Instead, we simply provide examples of (input,output) pairs and ask the machine to *learn* a suitable mapping itself.

Inputs and outputs

Basic terminology:

- The input space, X.

  E.g. 32 x 32 RGB images of animals

x :

- The output space, Y.

  E.g Names of 100 animals. y : "bear"

.

After seeing a bunch of examples (x, y), pick a mapping

$$f : X \rightarrow Y$$

that accurately replicates the input-output pattern of the examples.

Learning problems are often categorized according to the type of *output space:* (1) discrete, (2) continuous, (3) probability values, or (4) more general structures

Discrete output space: classification

Binary classification:

- Spam detection

  $X = \{$email messages$\}$

  $Y = \{$spam, not spam$\}$

- Credit card fraud detection

  $X = \{$descriptions of credit card transactions$\}$

  $Y = \{$fraudulent, legitimate$\}$

Multiclass classification:

- Animal recognition

  $X = \{$animal pictures$\}$

  $Y = \{$dog, cat, giraffe, . . .$\}$

- News article classification

  $X = \{$news articles$\}$ $Y = \{$politics, business, sports, . . .$\}$

Continuous output space: regression

- A parent's concerns

  How cold will it be tomorrow morning?

  $Y = [-273, \infty)$

- For the asthmatic

  Predict tomorrow's air quality (max over the whole day)

  $Y = [0, \infty)$ (< 100: okay, > 200: dangerous)

- Insurance company calculations

  In how many years will this person die?

  $Y = [0, 200]$

What are suitable predictor variables (X) in each case?

Conditional probability functions

Here $Y = [0, 1]$ represents probabilities.

- Dating service

  What is the probability these two people will go on a date if introduced to each other?

  If we modeled this as a classification problem, the binary answer would basically always be "no". The goal is to find matches that are slightly less unlikely than others.

- Credit card transactions

  What is the probability that this transaction is fraudulent?

  The probability is important, because – in combination with the amount of the transaction – it determines the overall risk and thus the right course of action.

Structured output spaces

The output space consists of structured objects, like sequences or trees.

Dating service

Input: description of a person
Output: rank-ordered list of all possible matches

Y = space of all permutations

Example:
x = *Tom*
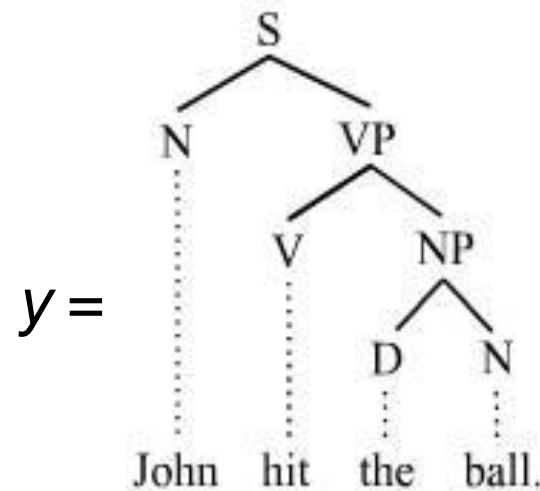y = (*Nancy , Mary , Chloe, . . .*)

Language processing

Input: English sentence
Output: parse tree showing grammatical structure
Y = space of all trees

Example:
x = "John hit the ball"

y =

Outline

- Introduction
  - Myself
  - TAs
  - Course
  - Machine Learning
  - ***Hands-On:*** *Intro to scikit-learn*

- Nonparametric Methods
  - Decision Trees
  - ***Hands-On***
  - ***Self – practice***
  - kNN
  - ***Hands-On***
  - ***Self – practice***

- Parametric Methods
  - Generative Models
  - ***Hands-On***
  - ***Self – practice***

Course outline

1. Nonparametric methods
2. Classification using parametrized models
3. Combining classifiers
4. Representation learning

Nonparametric methods: nearest neighbor
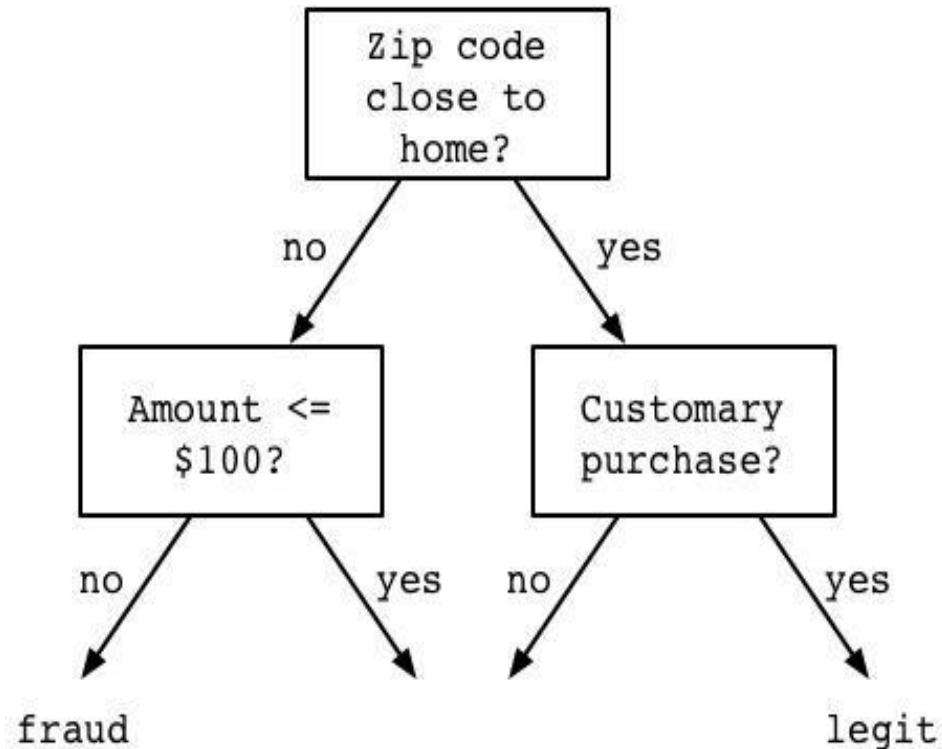
Training set: a collection of (x, y) pairs:
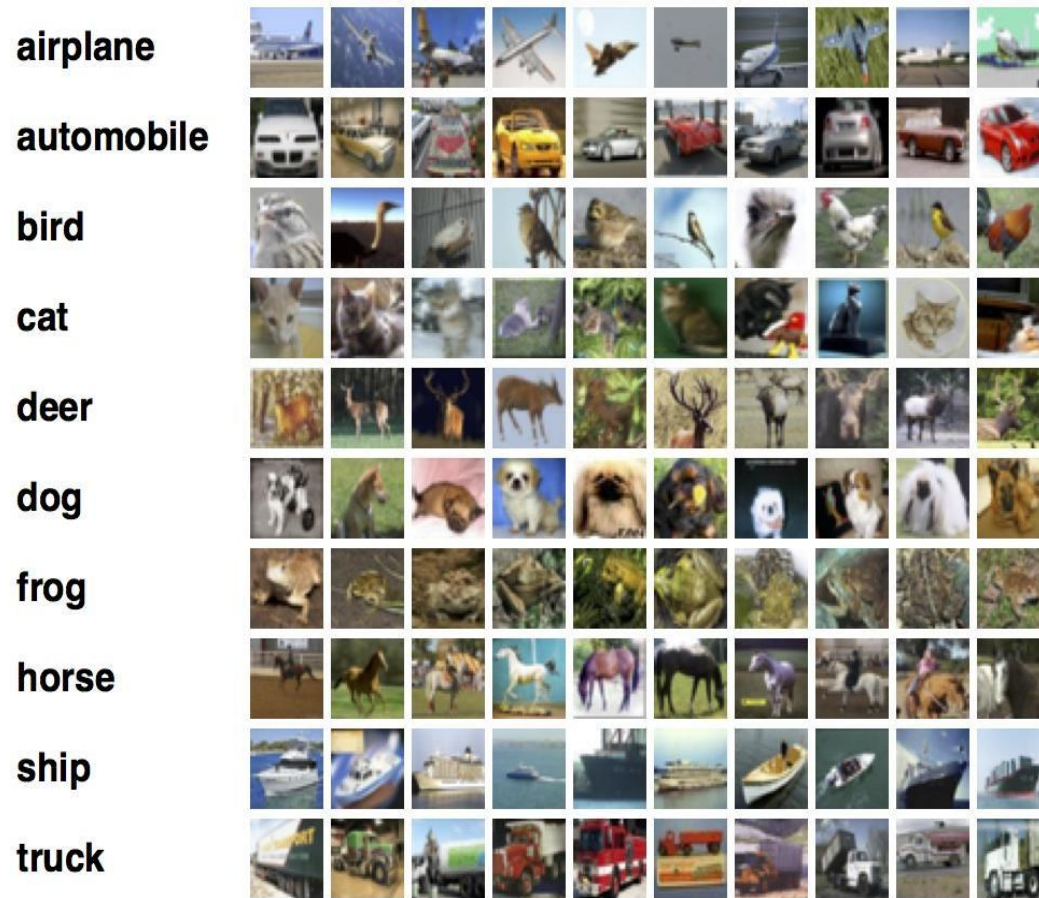
Given any x, find its nearest neighbor in the training set and predict that neighbor's y value.

Issues: (1) What distance function? (2) How to speed up search?

Nonparametric methods: decision tree

Credit card fraud detection: use training data to build a tree classifier

```
              ┌──────────────┐
              │   Zip code   │
              │   close to   │
              │    home?     │
              └──────────────┘
           no /              \ yes
             /                \
    ┌──────────────┐    ┌──────────────┐
    │  Amount <=   │    │  Customary   │
    │    $100?     │    │  purchase?   │
    └──────────────┘    └──────────────┘
    no /        \ yes   no /        \ yes
      /          \        /          \
   fraud                                legit
```

What do nearest neighbor and decision trees have in common?

- Unbounded in size

- Can model arbitrarily complex functions

They are *nonparametric methods*.

Classification with parametrized models

Classifiers with a fixed number of parameters can represent a limited set of functions. Learning a model is about picking a good approximation.

Typically the x's are points in d-dimensional Euclidean space, R^d
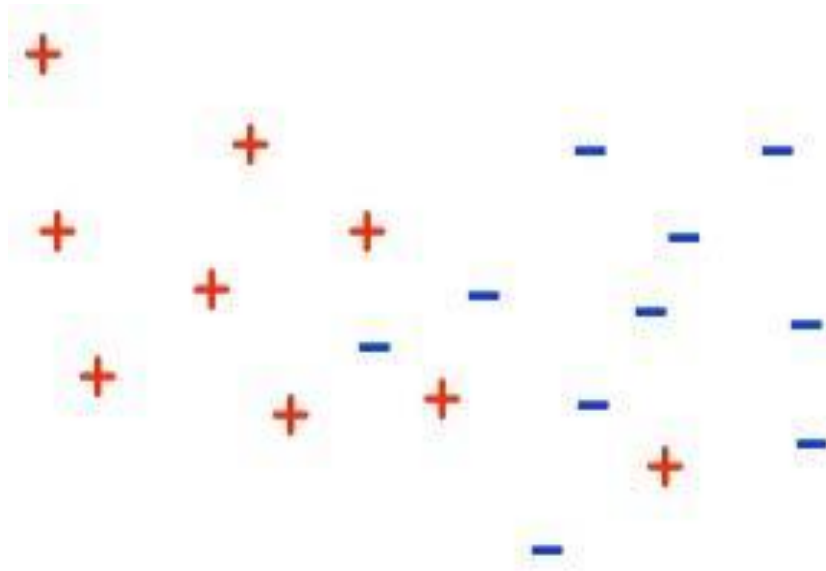
Two ways to classify:

- *Generative*: model the individual classes.
- *Discriminative*: model the decision boundary between the classes.

Generative models

Fit a probability distribution – like a multivariate Gaussian – to each class.
Thereafter use this *summary* rather than the data points themselves.

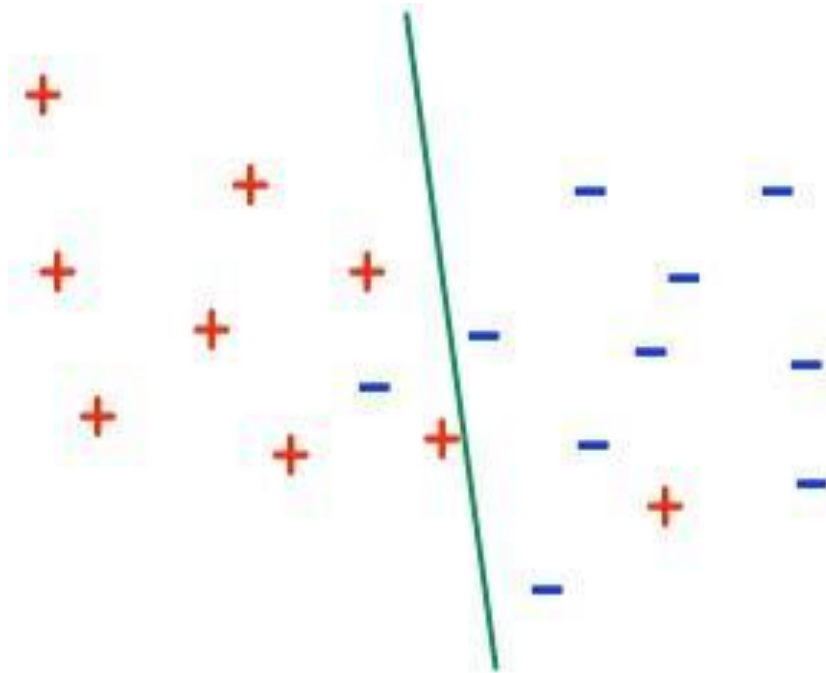To classify a new point: find the most probable class.
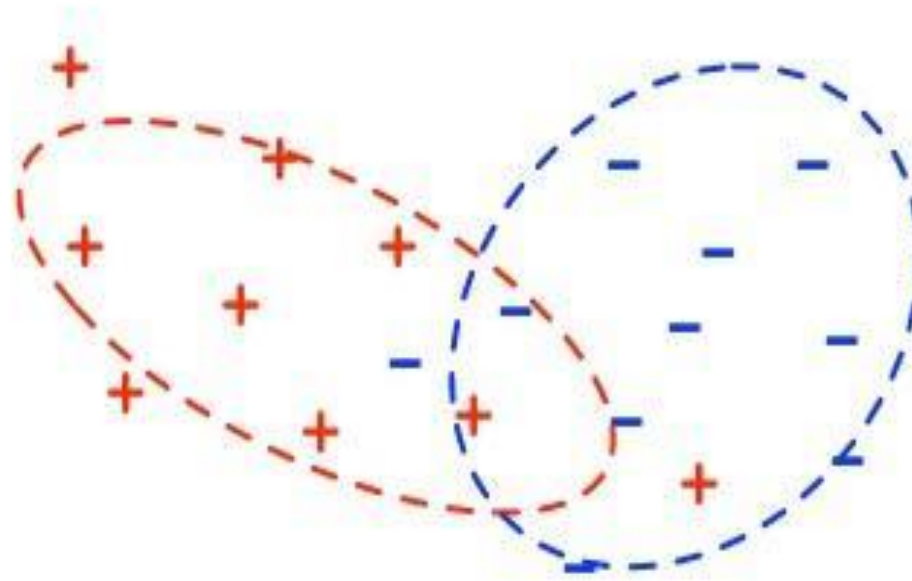
Examples: Naive Bayes, Fisher discriminant.

Under the hood: Bayes' rule, linear algebra (eigenvalues, eigenvectors).

Discriminative models

Approximate the boundaries between classes by simple – e.g. linear functions.

To classify a new point: figure out which side of the boundary it lies on.

Examples: support vector machine, logistic regression.
Under the hood: convex duality, optimization.

Generalization theory

- Complex, e.g. nonparametric, classifiers require a lot of training data to learn accurately.

- Simple, e.g. linear, classifiers require less.

What is the right notion of complexity? Are there formulas for how much data is enough? The answers are based on *large deviation theory*.

Richer classifiers via the kernel trick

We are good at finding linear classifiers in Euclidean space. But what if:

- The boundary between classes is far from linear?

  Example: quadratic, or higher-order polynomial, or even stranger.
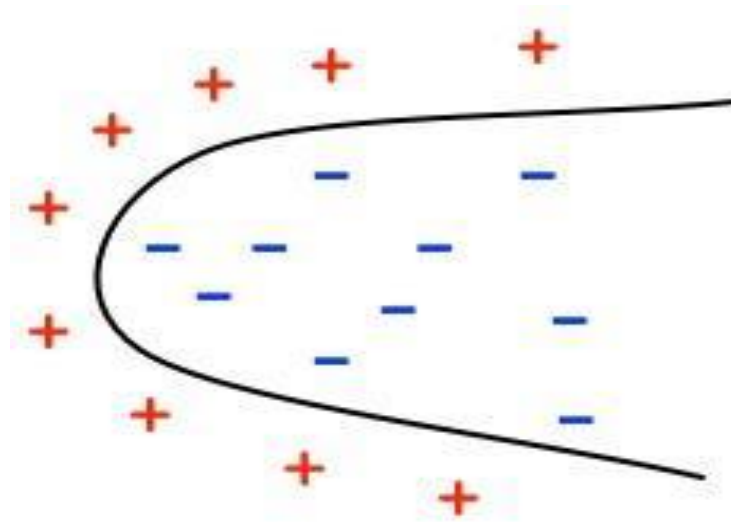
- The data aren't even vectors of numbers?

  Example: documents, DNA sequences, parse trees.

The *kernel trick* handles these scenarios seamlessly, by mapping the data to a suitable Euclidean space in which linear classification is possible!

Richer output spaces

Many classification methods were developed for the binary (two-label) case. Usually the output space is larger than this.

- Y = several classes.

  Examples:

  x = image, y = name of object in image

  x = news article, y = category (sports, politics, business, . . .)

- Y = structured objects.

  Examples:

  x = sentence in Swahili, y = transcription into English

  x = sentence in English, y = parse tree
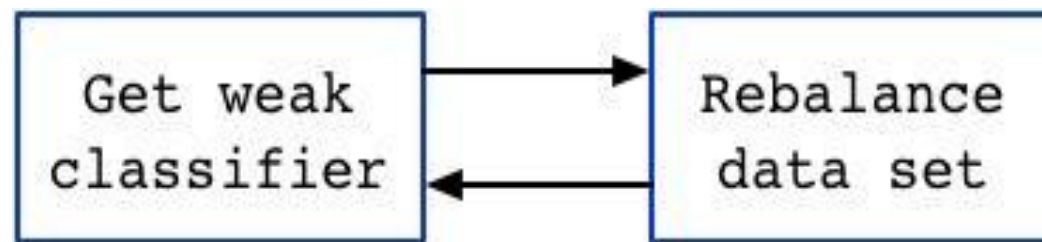
Extend binary classification to handle such cases!

Under the hood: error-correcting codes, dynamic programming.

Composing simple classifiers

A common situation in classifier learning:

Easy to find **weak classifiers** *– not very accurate, but better than random* to increase accuracy, *compose* weak classifiers.

Example: *boosting*

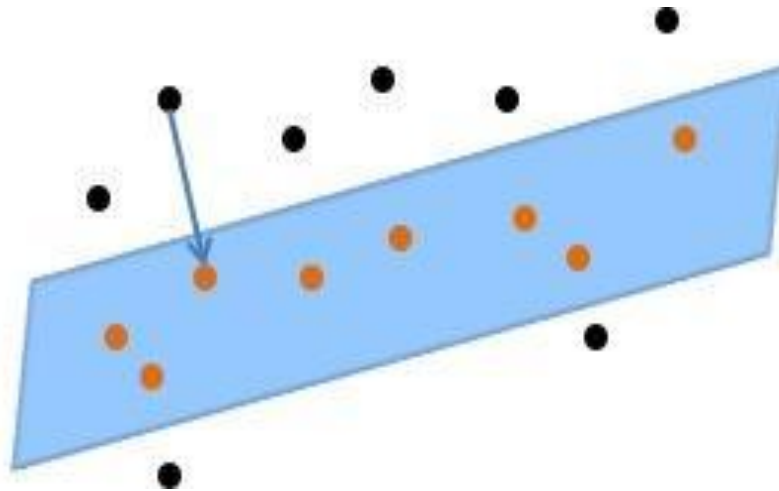Final classifier is a linear combination of all these weak classifiers.

Generically improve the performance of *any* kind of classifier!

Representation learning

A handful of key primitives:

❶ Dimensionality reduction and denoising.

Given data in high-dimensional Euclidean space, project to a low-dimensional linear subspace while retaining as much of the signal as possible.
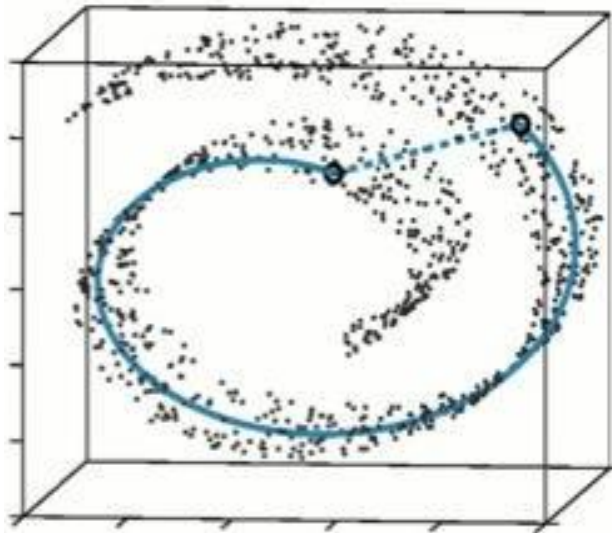

❷ Embedding and manifold learning.

❸ Metric learning.

Representation learning

A handful of key primitives:

1 Dimensionality reduction and denoising.

2 Embedding and manifold learning.

Given data that lie in a non-Euclidean space, find an embedding into Euclidean space that preserves as much of the geometry as possible.

3 Metric learning.

Representation learning

A handful of key primitives:

1. Dimensionality reduction and denoising.

2. Embedding and manifold learning.

3. Metric learning.

   Given data with only vague positional information, impose an Euclidean geometry that is suitable for classification.

   Example: $X = \{$a collection of m books$\}$.

   A user supplies $\binom{m}{2}$ similarity ratings, such as:

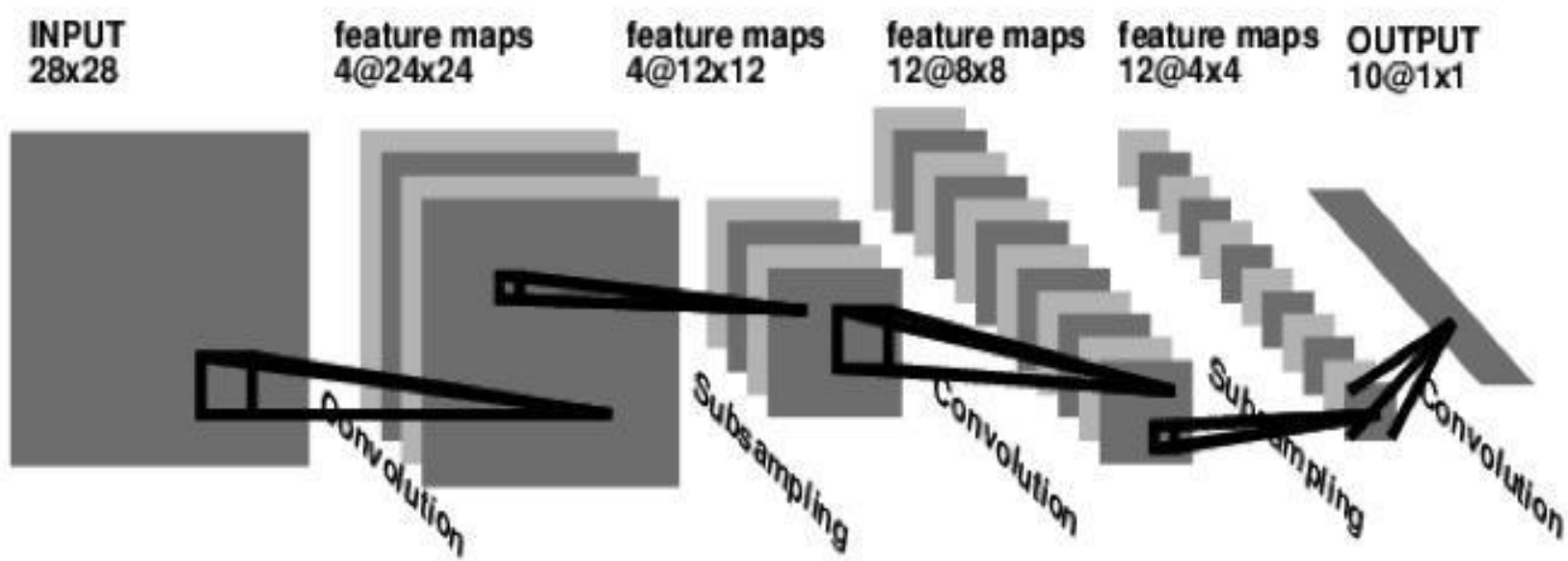   ("Pride and Prejudice", "Great Expectations"): similar

   ("Hamlet", "Great Expectations"): dissimilar

   Represent each book by a vector, respecting these ratings.

Under the hood: linear algebra, semidefinite programming.

Deep learning

Multi-layer neural nets achieve state-of-the-art performance across a range of benchmark problems in natural language processing, speech, and vision.

INPUT 28x28 — feature maps 4@24x24 — feature maps 4@12x12 — feature maps 12@8x8 — feature maps 12@4x4 — OUTPUT 10@1x1
Convolution — Subsampling — Convolution — Subsampling — Convolution

Under the hood: stochastic gradient descent, dictionary learning, autoencoders.

Identifying Machine Learning Tasks

- Will this customer purchase service S1 if given incentive I1 ?
    - Supervised Learning
        - Classification Problem
        - Binary Target(the customer either purchases or not)

- Which Service package (S1,S2,or none) will a customer likely purchase if given incentive I1 ?
    - Supervised Learning
        - Classification Problem
        - Three valued target

Identifying Machine Learning Tasks

- "I want to know which of my customers are the most profitable?"
  - Database query

- "I have a budget to target 10,000 existing customers with a special offer. I would like to identify those customers most likely to respond to the special offer"
  - Supervised Learning
    - Probability ranking - Classification problem

Identifying Machine Learning Tasks

- How can we categorize our customers?
  - Unsupervised Learning
    - Clustering

- "How much will this customer use the service?"
  - Supervised Learning
    - Regression problem
      - Numeric target
      - Target variable: amount of usage per customer

Identifying Machine Learning Tasks

- "I would like to segment my customers into groups based on their demographics and prior purchase activity. I am not focusing on improving a particular task, but would like to generate ideas."
  - Unsupervised Learning
    - Clustering

Course outline

① Nonparametric Methods
- Decision Trees
- Nearest neighbor

② Classification using parametrized models
- Generative models
- Discriminative models
- Richer decision boundaries using the kernel trick
- Richer output spaces
- Generalization theory

③ Combining classifiers
- Boosting, bagging, and random forests
- Online learning

④ Representation learning
- Linear projection
- Embeddings
- Metric learning
- Deep learning