Class overview and intro to ML

Subhransu Maji

CMPSCI 689: Machine Learning

20 January 2015

Course background

Course background

- What is the course about?
  - Finding (and exploiting) patterns in data
  - Replacing “humans writing code” with “humans supplying data”
    - System figures out what the person wants based on examples
    - Need to abstract from “training” examples to “test” examples
    - Most central issue in ML: “generalization”

Course background

- What is the course about?
  - Finding (and exploiting) patterns in data
  - Replacing “humans writing code” with “humans supplying data”
    - System figures out what the person wants based on examples
    - Need to abstract from “training” examples to “test” examples
    - Most central issue in ML: “generalization”

- Why is machine learning so cool?
  - Broad applicability
    - Finance, robotics, vision, machine translation, medicine, etc
    - Close connections between theory and practice
    - Open area, lots of room for new work

Course goals

Course goals

- By the end of the semester, you should be able to:
  - Look at a problem and identify if ML is an appropriate solution
  - If so, identify what types of algorithms might be applicable
  - Apply those algorithms
  - Conquer the world

Course goals

- By the end of the semester, you should be able to:
  - Look at a problem and identify if ML is an appropriate solution
  - If so, identify what types of algorithms might be applicable
  - Apply those algorithms
  - Conquer the world
- In order to get there, you will need to:
  - Do a lot of math (calculus, linear algebra, probability)
  - Do a fair amount of programming
  - Work hard (this is a 3-unit course)

Topics covered

Topics covered

- Supervised learning: learning with a teacher

Topics covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher

Topics covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher
- Complex settings: learning in a complicated world
  - Time-series models
  - Structured prediction
  - Semi-supervised learning
  - Large-scale learning

Topics covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher
- Complex settings: learning in a complicated world
  - Time-series models
  - Structured prediction
  - Semi-supervised learning
  - Large-scale learning

- Not a zoo tour!
- Not an introduction to tools!
- You will learn how these techniques work and how to implement them

Requirements and grading

- Weekly homework assignments: **20%**
- Mini-projects: **45%**
- Project: **30%**
- Class/forum participation: **5%**

Requirements and grading

- Weekly homework assignments: **20%**
  - About 12 in total, graded at 0, 0.5 or 1
  - Completed individually
  - May not be late at all

- Mini-projects: **45%**

- Project: **30%**

- Class/forum participation: **5%**

Requirements and grading

- Weekly homework assignments: **20%**
  - About 12 in total, graded at 0, 0.5 or 1
  - Completed individually
  - May not be late at all

- Mini-projects: **45%**
  - Three in total
  - Completed individually (but can be discussed with others)
  - May be 48 hours late, at 50% mark down

- Project: **30%**

- Class/forum participation: **5%**

Requirements and grading

- Weekly homework assignments: **20%**
  - About 12 in total, graded at 0, 0.5 or 1
  - Completed individually
  - May not be late at all
- Mini-projects: **45%**
  - Three in total
  - Completed individually (but can be discussed with others)
  - May be 48 hours late, at 50% mark down
- Project: **30%**
  - Canned or your choice, teams of two or more
  - Proposal, presentation (or poster), report
- Class/forum participation: **5%**

Who should take this course?

- Is this the right course for you?
  - Do you have all the pre-requisites?
    - good math and programming background
  - Balance of theory vs. practice. Other courses being offered:
    - 589 - Machine learning (but focus on applications)
    - 688 - Probabilistic graphical models (we will cover this only briefly)
- Still not sure?
  - talk to me after class
- Wait listed?
  - Will decide on a case by case basis

Course logistics

- My office hours: Tue 2:15-3:15 CS 142 (or by appointment)
- TA: Xiaojian Wu
  - TA office hours: TBD
- Course website: http://www-edlab.cs.umass.edu/~smaji/cmpsci689/
  - Class slides, homework assignments will the posted here
  - Check regularly for announcements
- Moodle for homework submission
  - Also a place for discussions
- Edlab for computational support (and MATLAB)
  - You may buy a student license of MATLAB for 100$

Things you need to know now!

- Finish homework 00
  - Due 22 Jan (that's Thursday! before class)
  - Submit in *.pdf* format *only* via **moodle**
    - Those who are not yet on moodle may email me
- Get started on p 0
  - Intro to MATLAB programming
  - Set up accounts on Edlab (if you don't have MATLAB)
  - Will **not** be graded
- Complete the first reading
- Read the web page!

Now, on to some real content …

(but first, questions?)

Classification

- How would you write a program to distinguish a picture of me from a picture of someone else?

- How would you write a program to determine whether a sentence is grammatical or not?

- How would you write a program to distinguish cancerous cells from normal cells?

Classification

- How would you write a program to distinguish a picture of me from a picture of someone else?
  - Provide examples pictures of me and pictures of other people and let a classifier learn to distinguish the two.
- How would you write a program to determine whether a sentence is grammatical or not?
  - Provide examples of grammatical and ungrammatical sentences and let a classifier learn to distinguish the two.
- How would you write a program to distinguish cancerous cells from normal cells?
  - Provide examples of cancerous and normal cells and let a classifier learn to distinguish the two.

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three principal components**

Data (“weather” prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three principal components**
  1. Class label (aka “label”, denoted by y)

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three principal components**
  1. Class label (aka "label", denoted by y)
  2. Features (aka "attributes")

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three principal components**
  1. Class label (aka "label", denoted by y)
  2. Features (aka "attributes")
  3. Feature values (aka "attribute values", denoted by x)

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three principal components**
  1. Class label (aka "label", denoted by *y*)
  2. Features (aka "attributes")
  3. Feature values (aka "attribute values", denoted by *x*)
     - ➡ Feature values can be binary, nominal or continuous

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three principal components**
  1. Class label (aka "label", denoted by *y*)
  2. Features (aka "attributes")
  3. Feature values (aka "attribute values", denoted by *x*)
     - ➡ Feature values can be binary, nominal or continuous
- **A *labeled* dataset is a collection of *(x, y)* pairs**

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Task:**

Class	Outlook	Temperature	Windy?
???	Sunny	Low	No

- **Predict the class of this "test" example**

Data ("weather" prediction)

- **Example dataset:**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Task:**

Class	Outlook	Temperature	Windy?
???	Sunny	Low	No

- **Predict the class of this "test" example**
- Requires us to **generalize** from the training data

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

Data (face recognition)

Class	Image	Class	Image
Avrim		Tom	
Avrim		Tom	
Avrim		Tom	
Avrim		Tom	

- What is a good *representation* for images?

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

- What is a good *representation* for images? Pixel values?

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

- What is a good *representation* for images? Pixel values? Edges?

Ingredients for classification

Ingredients for classification

- **Whole idea:** Inject *your* knowledge into a learning system

Ingredients for classification

- **Whole idea:** Inject *your* knowledge into a learning system
- **Sources of knowledge:**
  1. Feature representation
  2. Training data: labeled examples
  3. Model

Ingredients for classification

- **Whole idea:** Inject *your* knowledge into a learning system
- **Sources of knowledge:**
  1. Feature representation
     - ➡ Not typically a focus of machine learning
     - ➡ Typically seen as "problem specific"
     - ➡ However, it's hard to learn from bad representations
  2. Training data: labeled examples
  3. Model

Ingredients for classification

- **Whole idea:** Inject *your* knowledge into a learning system
- **Sources of knowledge:**
  1. Feature representation
     - ➡ Not typically a focus of machine learning
     - ➡ Typically seen as “problem specific”
     - ➡ However, it’s hard to learn from bad representations
  2. Training data: labeled examples
     - ➡ Often expensive to label lots of data
     - ➡ Sometimes data is available for “free”
  3. Model

Ingredients for classification

- **Whole idea:** Inject *your* knowledge into a learning system
- **Sources of knowledge:**
  1. Feature representation
     - ➡ Not typically a focus of machine learning
     - ➡ Typically seen as “problem specific”
     - ➡ However, it’s hard to learn from bad representations
  2. Training data: labeled examples
     - ➡ Often expensive to label lots of data
     - ➡ Sometimes data is available for “free”
  3. Model
     - ➡ No single learning algorithm is always good (“no free lunch”)
     - ➡ Different learning algorithms work with different ways of representing the learned classifier

Regression

Regression

- **Regression** is like **classification** except the **labels are real valued**

Regression

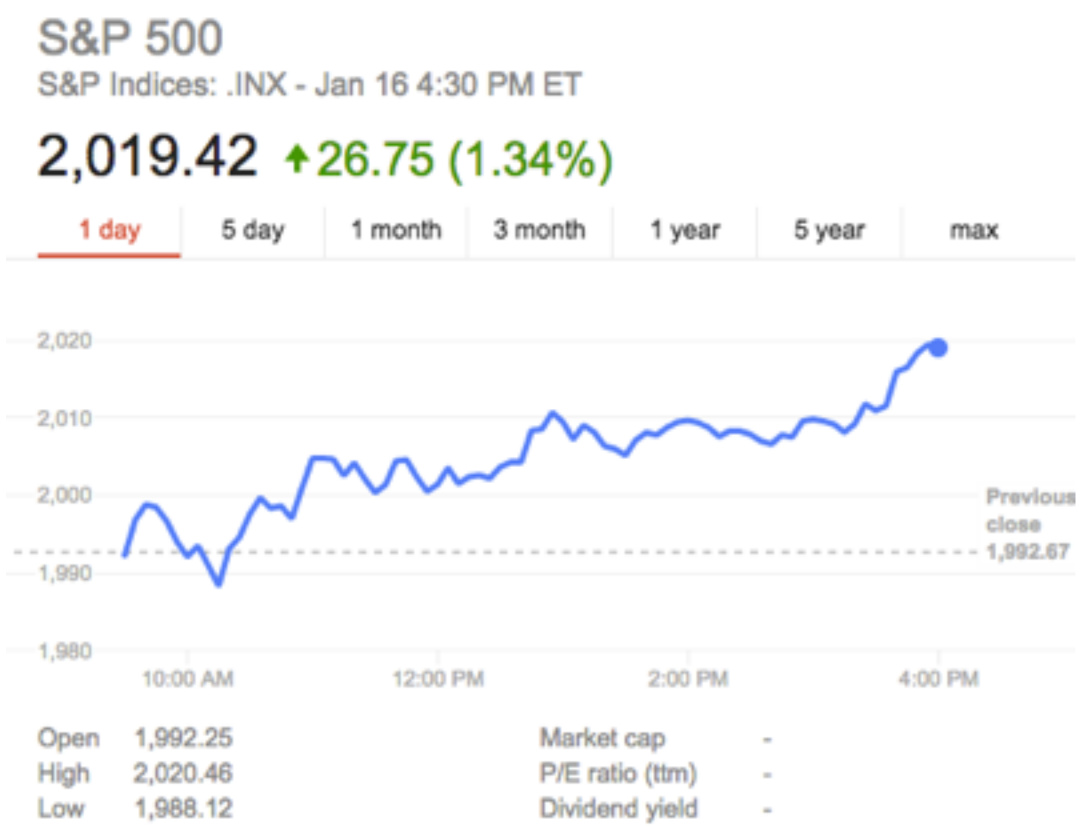
- **Regression** is like **classification** except the **labels are real valued**
- **Example applications:**

Regression

- **Regression** is like **classification** except the **labels are real valued**
- **Example applications:**
  - Stock value prediction

Regression

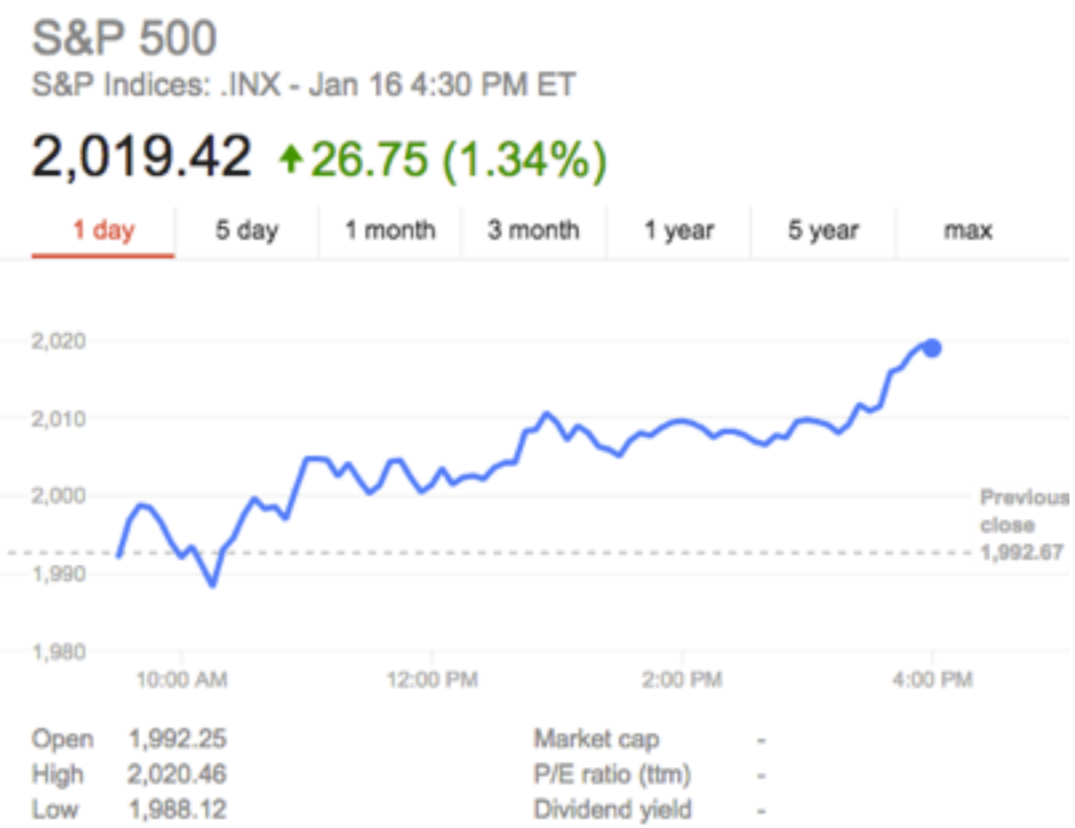
- **Regression** is like **classification** except the **labels are real valued**
- **Example applications:**
  - Stock value prediction
  - Income prediction



Regression

- **Regression** is like **classification** except the **labels are real valued**

- **Example applications:**
  - Stock value prediction
  - Income prediction
  - CPU power consumption

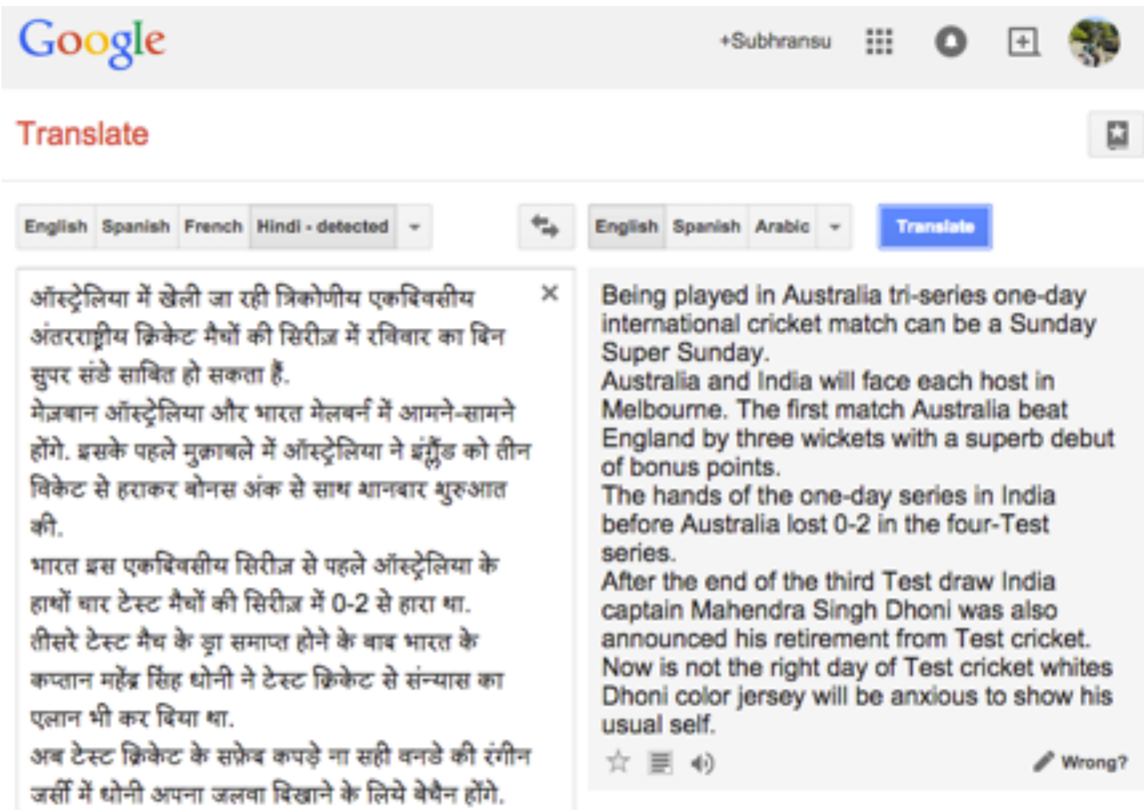


Regression

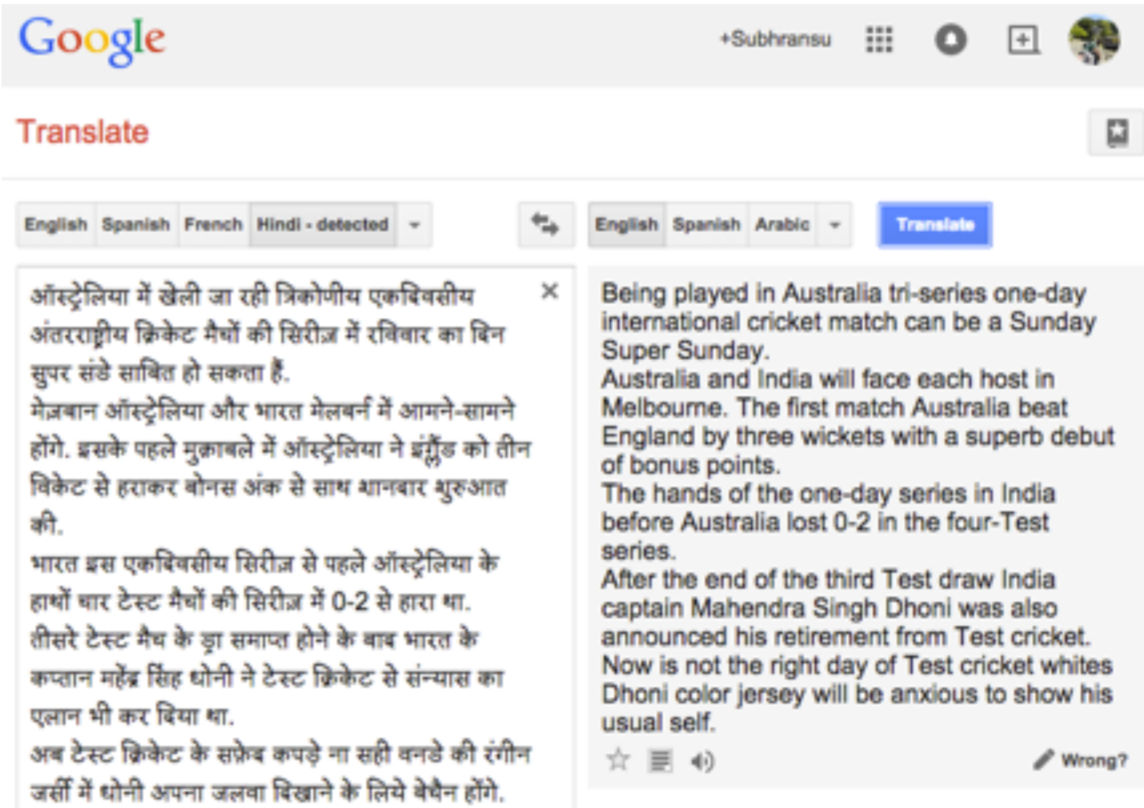
- **Regression** is like **classification** except the **labels are real valued**
- **Example applications:**
  - Stock value prediction
  - Income prediction
  - CPU power consumption
  - Your grade in CMPSCI 689

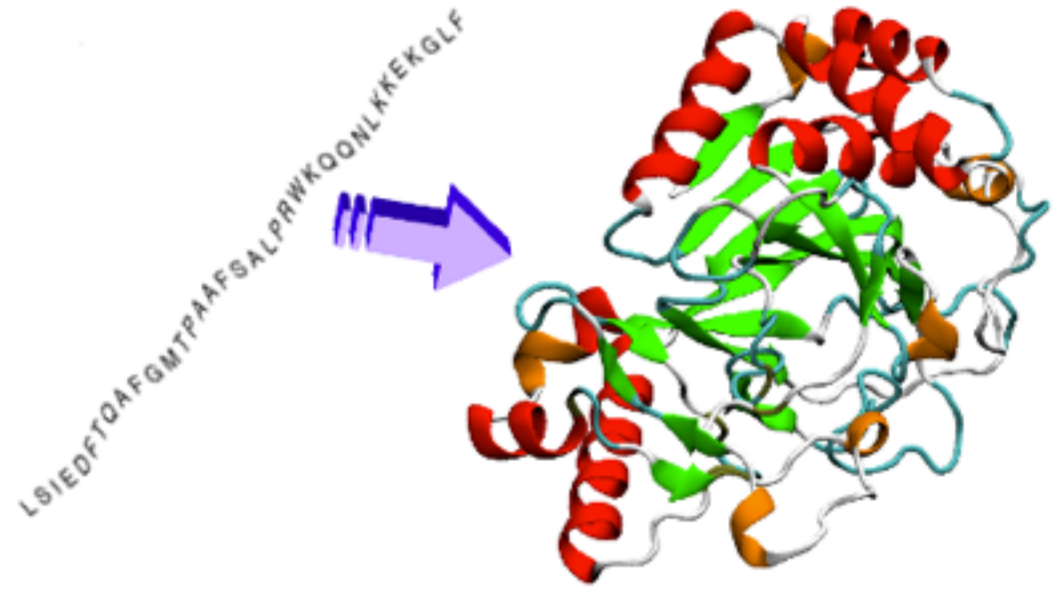
Structured prediction

Structured prediction

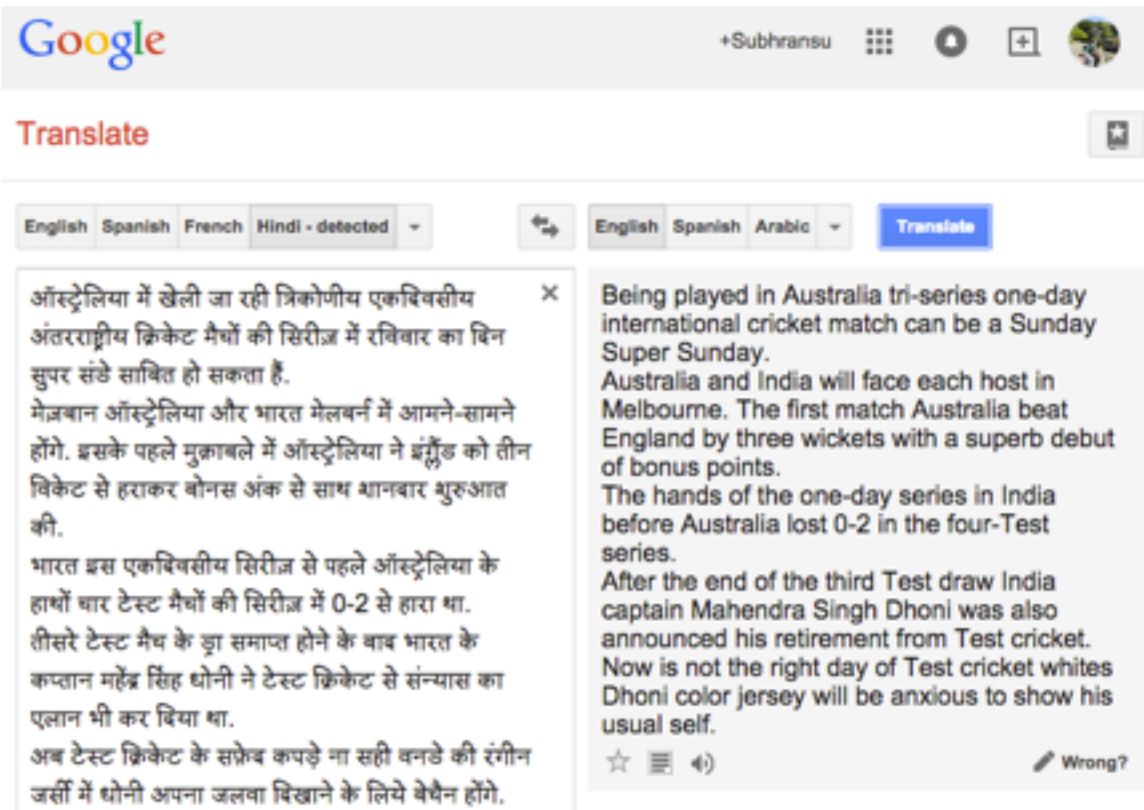


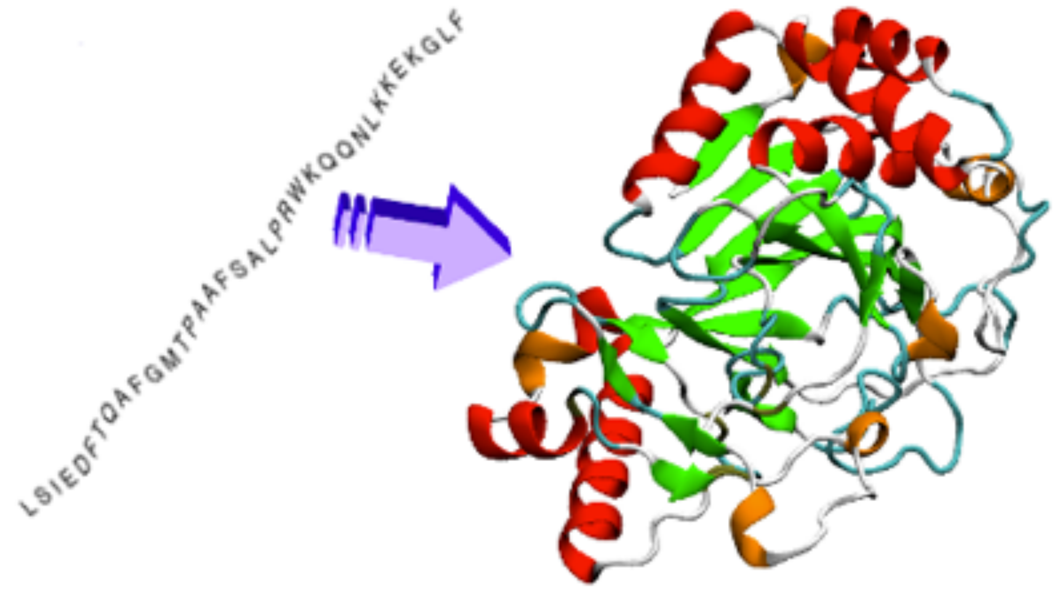
Structured prediction

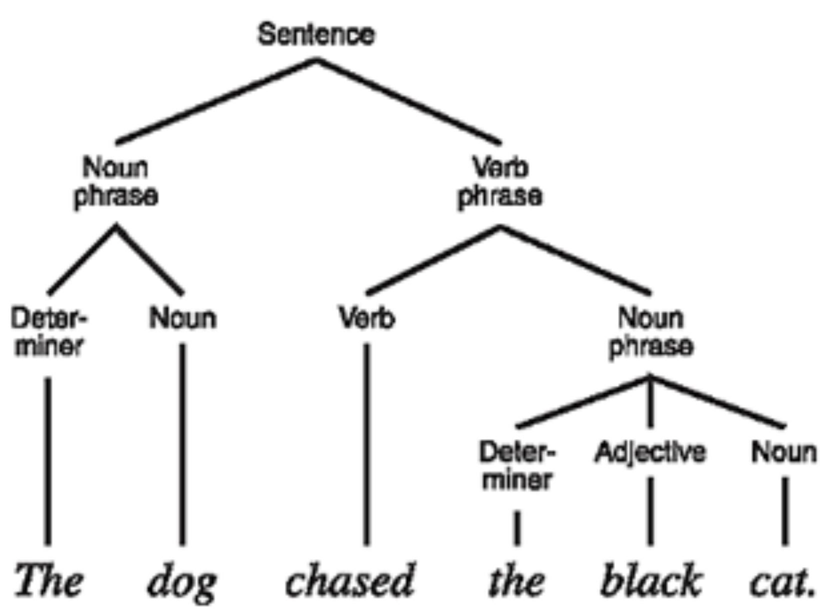




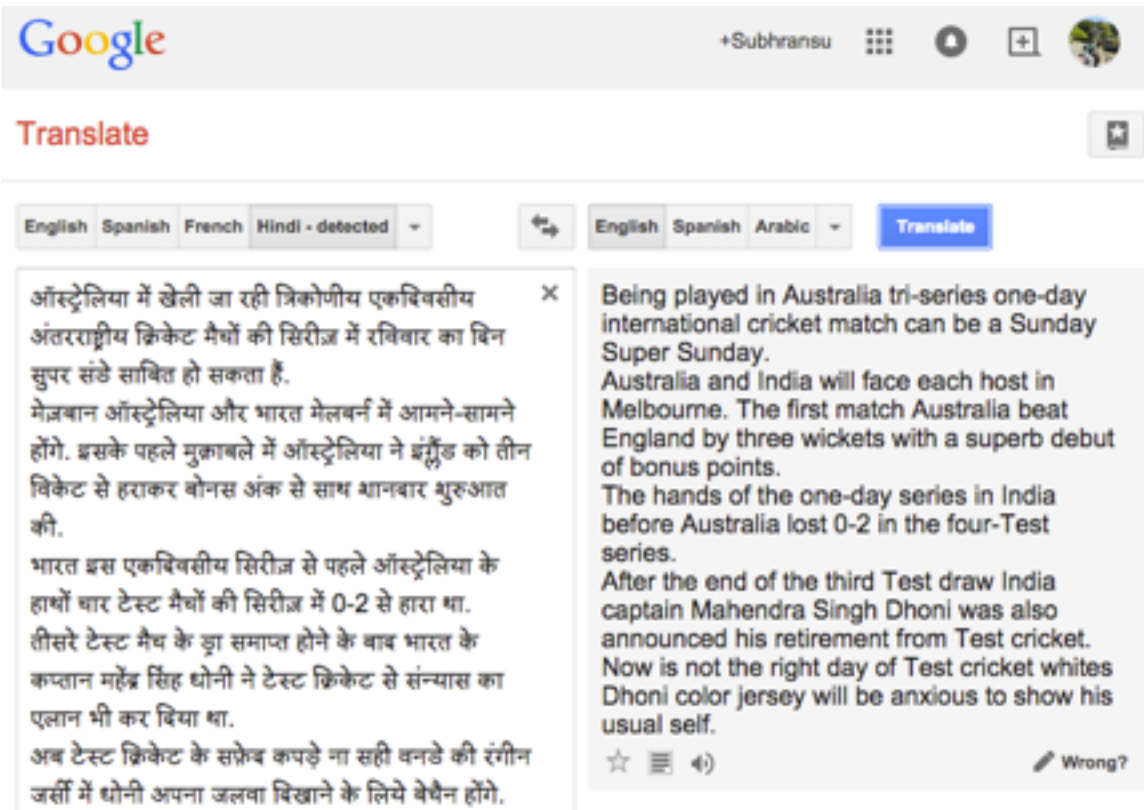
Structured prediction

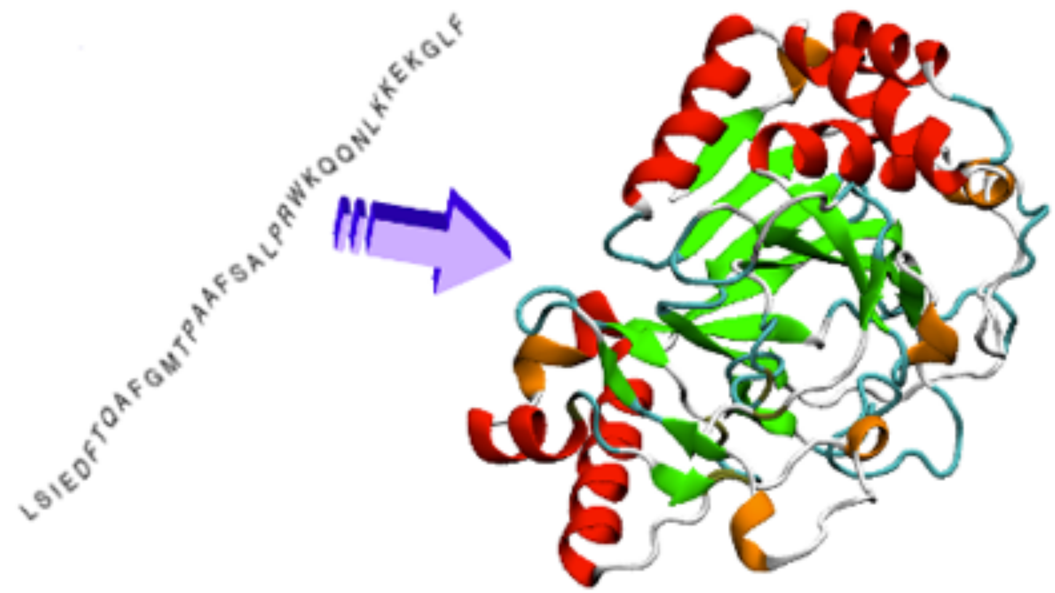


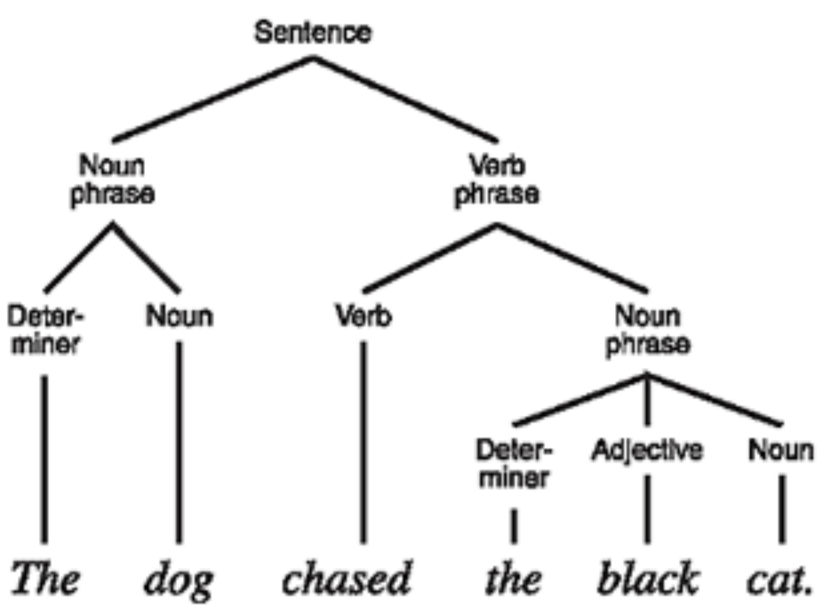


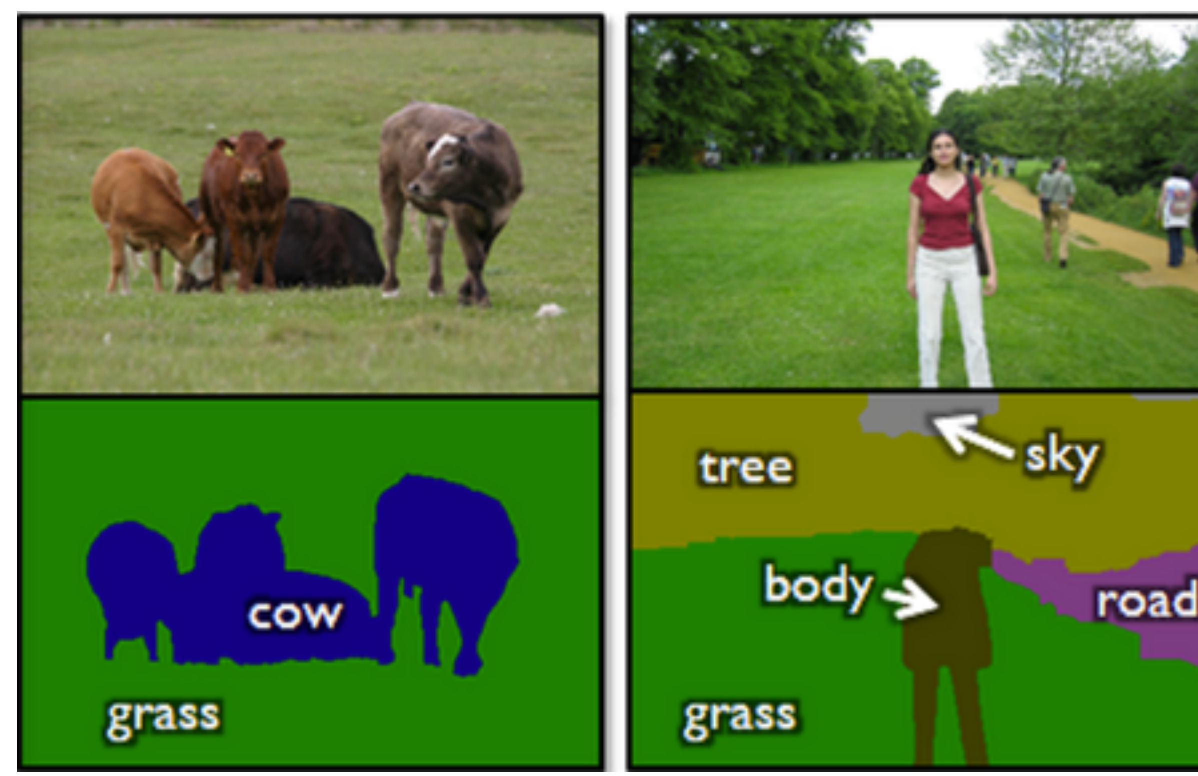


Structured prediction









Unsupervised learning: Clustering

- **Two types of clustering**
  1. Clustering into distinct components

  2. Hierarchical clustering

Unsupervised learning: Clustering

- **Two types of clustering**
  1. Clustering into distinct components

  2. Hierarchical clustering

Unsupervised learning: Clustering

- **Two types of clustering**
  1. Clustering into distinct components

  2. Hierarchical clustering

Unsupervised learning: Clustering

- **Two types of clustering**
  1. Clustering into distinct components

  ➡ How many clusters are there?

  2. Hierarchical clustering

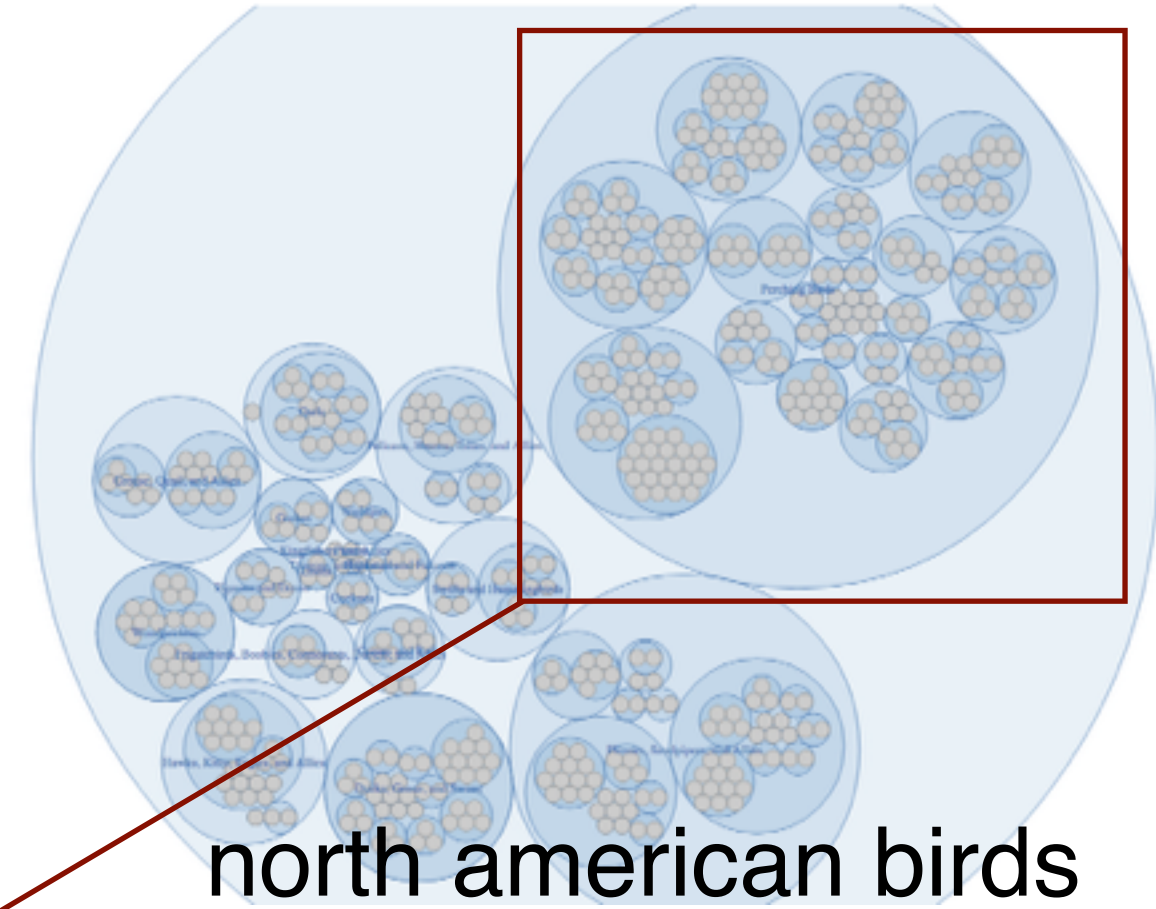
Unsupervised learning: Clustering

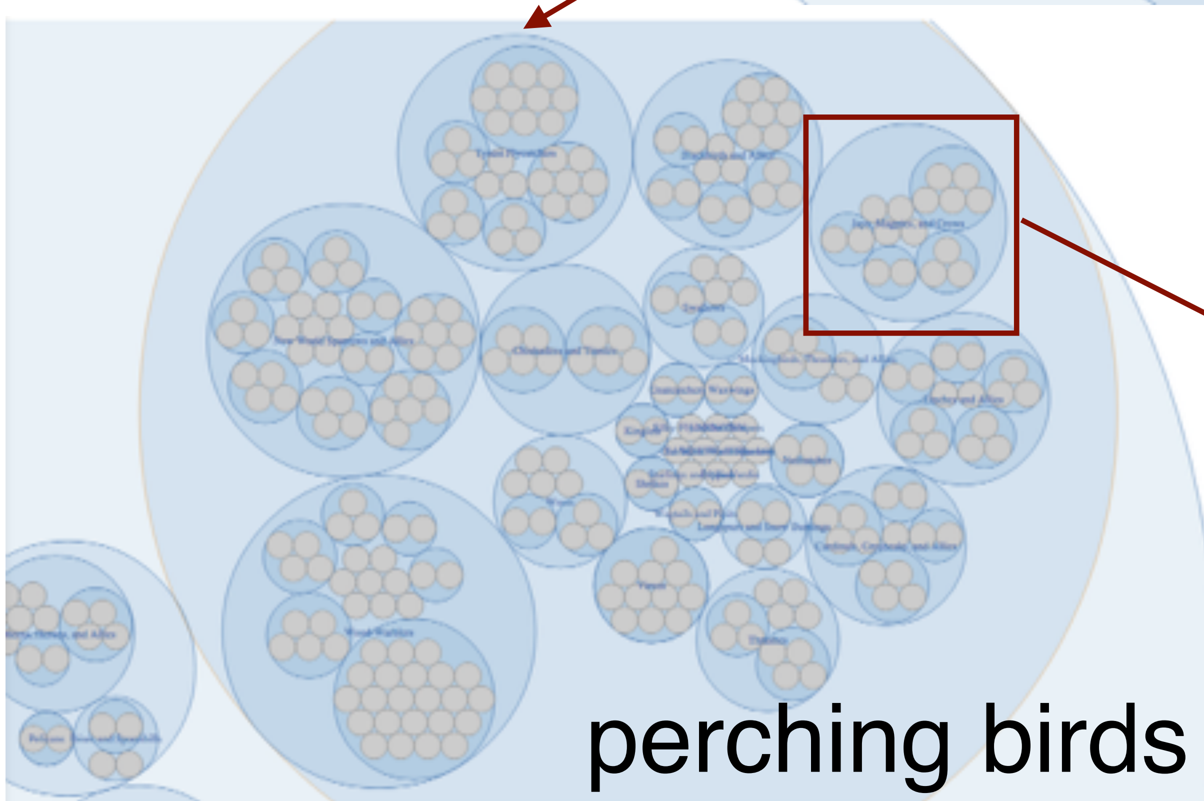
- **Two types of clustering**
  1. Clustering into distinct components

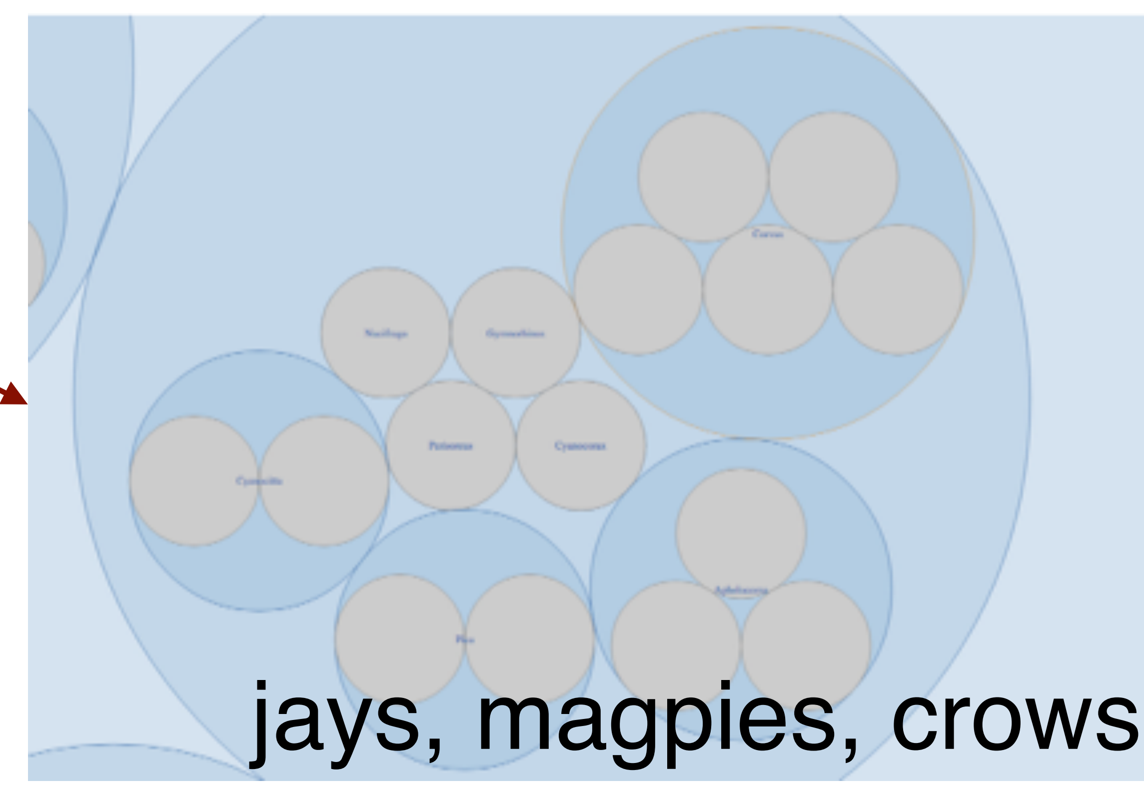
     - ➡ How many clusters are there?
     - ➡ What is important? Person? Expression? Lighting?
  2. Hierarchical clustering

Unsupervised learning: Clustering

http://vision.ucsd.edu/~gvanhorn/

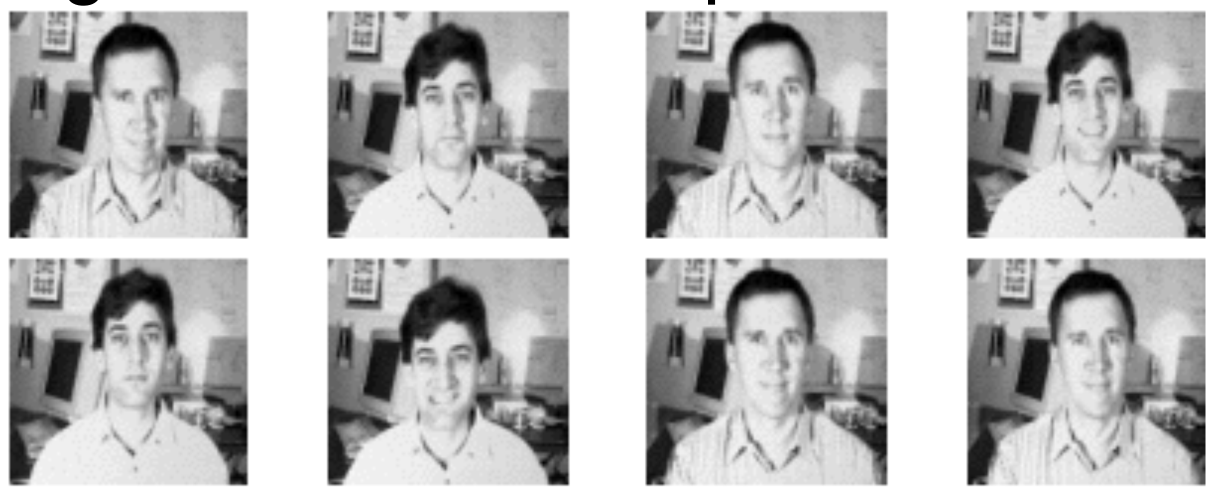




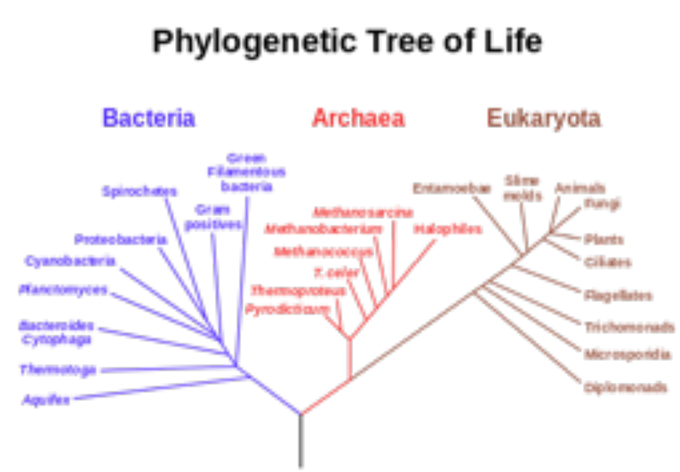


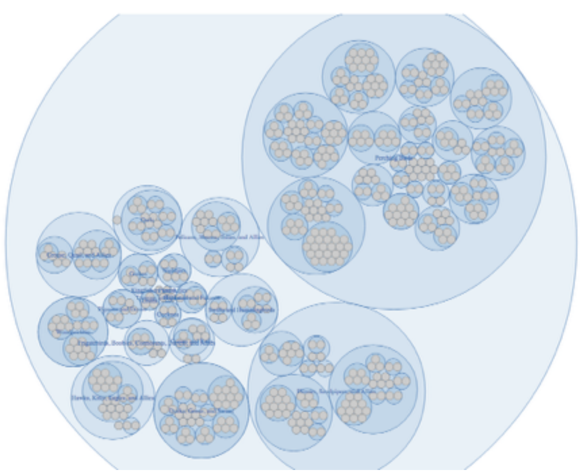
Unsupervised learning: Clustering

- **Two types of clustering**
  1. Clustering into distinct components



     - ➡ How many clusters are there?
     - ➡ What is important? Person? Expression? Lighting?
  2. Hierarchical clustering
     - ➡ What is important?
     - ➡ How will we use this?

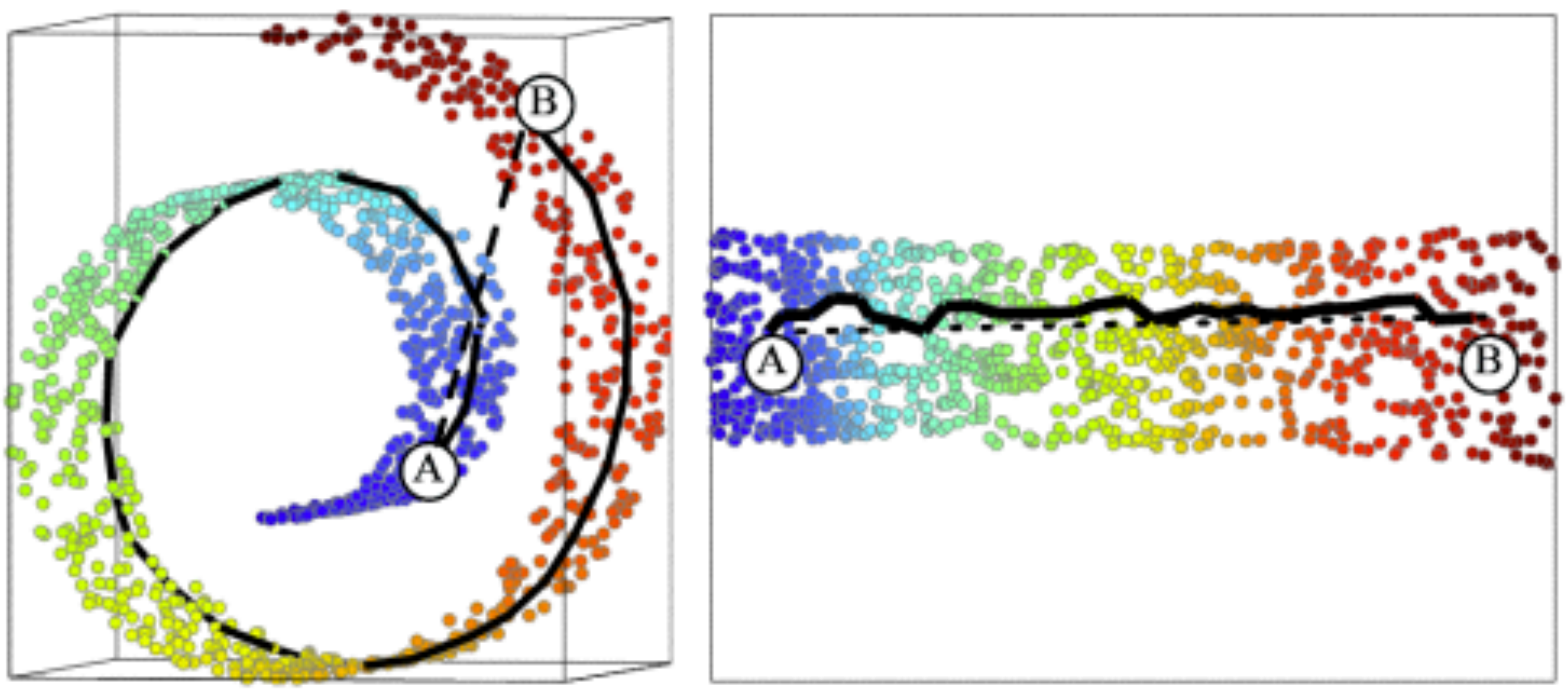




Unsupervised learning: Manifold learning

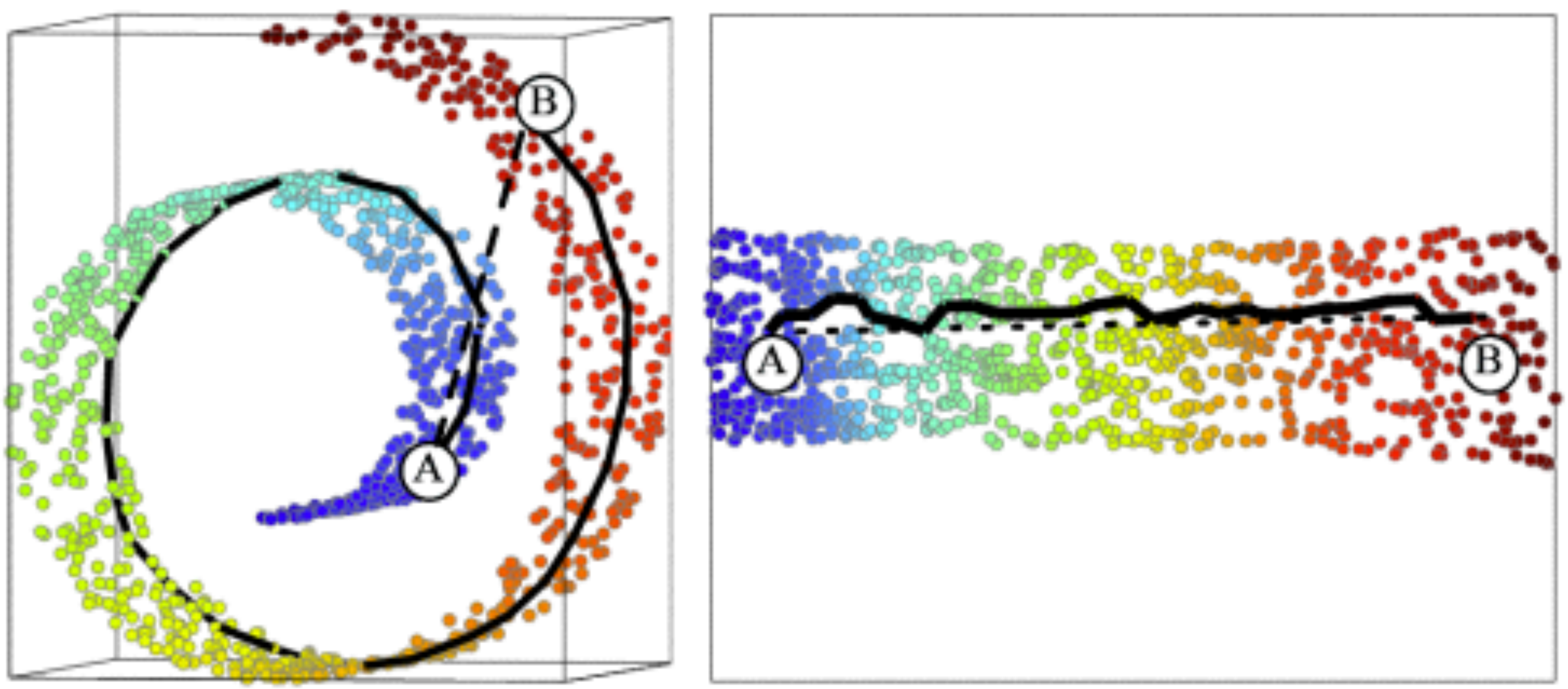
Unsupervised learning: Manifold learning

- Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped
- Task is to recover the true geometry of the underlying data



Unsupervised learning: Manifold learning

- Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped
- Task is to recover the true geometry of the underlying data



  - Usually, replace "two" with d and "three" with D for $d << D$
  - Useful for visualization (when d = 2 or 3)
  - Also useful for finding good representations for input to classifiers

Reinforcement learning

- Unlike classification, regression and unsupervised learning, RL does not receive examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

- Example problems:
  1. Chess
  2. Robot control
  3. Taxi driving
- Key trade-off is exploration versus exploitation

Reinforcement learning: general setting

Reinforcement learning: general setting

- A (simple) reinforcement learning problem is defined by:
  - A state space that defines the *world* that our *agent* inhabits
  - A set of actions that an agent can take in any state
  - The reward the agent gets for reaching some particular state

Reinforcement learning: general setting

- A (simple) reinforcement learning problem is defined by:
  - A state space that defines the *world* that our *agent* inhabits
  - A set of actions that an agent can take in any state
  - The reward the agent gets for reaching some particular state

- The *goal* of the agent is to take a sequence of actions so as to maximize the sum of rewards
- The agent's output is a *policy* that maps states to actions

Reinforcement learning: general setting

- A (simple) reinforcement learning problem is defined by:
  - A state space that defines the *world* that our *agent* inhabits
  - A set of actions that an agent can take in any state
  - The reward the agent gets for reaching some particular state

- The *goal* of the agent is to take a sequence of actions so as to maximize the sum of rewards
- The agent's output is a *policy* that maps states to actions

- **Trivia:** UMass played a significant role in advancing RL

Why do we care about math?!

- Calculus and linear algebra
  - Techniques for finding maxima/minima of functions
  - Convenient language for high dimensional data analysis
- Probability
  - The study of the outcomes of repeated experiments
  - The study of the plausibility of some event
- Statistics:
  - The analysis and interpretation of data

- Statistics makes heavy use of probability theory

Why do we care about probability & statistics?

- Recall, statistics is the analysis and interpretation of data

- In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data

- How is machine learning different from statistics?
  1. Stats care about the model, we care about predictions
  2. Stats care about model fit, we care about generalization
  3. Stats tries to explain the world, we try to predict the future

Why do we care about probability & statistics?

- Recall, statistics is the analysis and interpretation of data
- In machine learning, we attempt to generalize from one "training" data set to general "rules" that can be applied to "test" data
- How is machine learning different from statistics?
  1. Stats care about the model, we care about predictions
  2. Stats care about model fit, we care about generalization
  3. Stats tries to explain the world, we try to predict the future

And it all started with a lady drinking tea …

History of ML?

- Initial attempts at object recognition [Rosenblatt, 1958]
- Learning to play checker [Samuel, 1959, 1963]
- Rosenblatt can't learn XOR [Minsky & Pappert, 1969]
- Symbolic learning [Winston, 1975; Buchanan 1971]
- Backpropagation for neural nets [Werbos, 1974; Rummelhart, 1986]
- PAC model of learning theory [Valiant, 1984]
- Optimization enters machine learning [Bennett & Mangasarian, 1993]
- Kernel methods for non-linearity [Cortes & Vapnik, 1995]
- Machine learning behind day-to-day tasks [2005ish]
- Machine learning takes over the world [2010ish]

Slide credits

- These content of slides are adapted from the machine learning course taught by Hal Daume's at University of Maryland, College Park
- The figure on the hierarchical clustering of birds is from Grant Van Horn's webpage http://vision.ucsd.edu/~gvanhorn/ (Take a look at the page for an interactive version)