# Experimental Design for Machine Learning on Multimedia Data
# Lecture 3

**Dr. Gerald Friedland,**
**fractor@eecs.berkeley.edu**

Website: http://www.icsi.berkeley.edu/~fractor/spring2019/

# Projects

- **10 Questions online, will also be discussed today.**
- **Talk to Jaeyoung Choi**

# Today

- Project Questions
- Recap: Memory Equivalent Capacity: Q&A and Galton Board
- Generalization I

# Project Questions 1-4 (Data & Problem Inspection)

1) What is the variable the machine learner should predict? What is the required accuracy for success? What impact will adversarial examples have?

2) How much data do we have to train the prediction of the variable? Are the classes balanced? How many modalities could be exploited in the data? Is there temporal information? How much noise are we expecting? Do you expect bias?

3) How well is the data annotated (anecdotally)? What is the annotator agreement (measured)?

4) Given questions 1-3: Are we reducing information (pattern matching) or do we need to infer information (statistical machine learner)? As a consequence, what seems the best choice for the type of machine learner per modality?

# Project Questions 5-8 (Training for Generalization)

1) Estimate the memory equivalent capacity needed for the machine learner of your choice. What is the expected generalization? How does the progression look like.

2) Train your machine learner for accuracy at memory equivalent capacity. Can you reach near 100% memorization? If not, why (diagnose)?

3) Train your machine learner for generalization: Plot the accuracy/capacity curve. What is the expected accuracy and generalization ratio at the point you decided to stop? Do you need to try a different machine learner (if so, redo from 5)? Should you extract features (if so, redo from 5)?

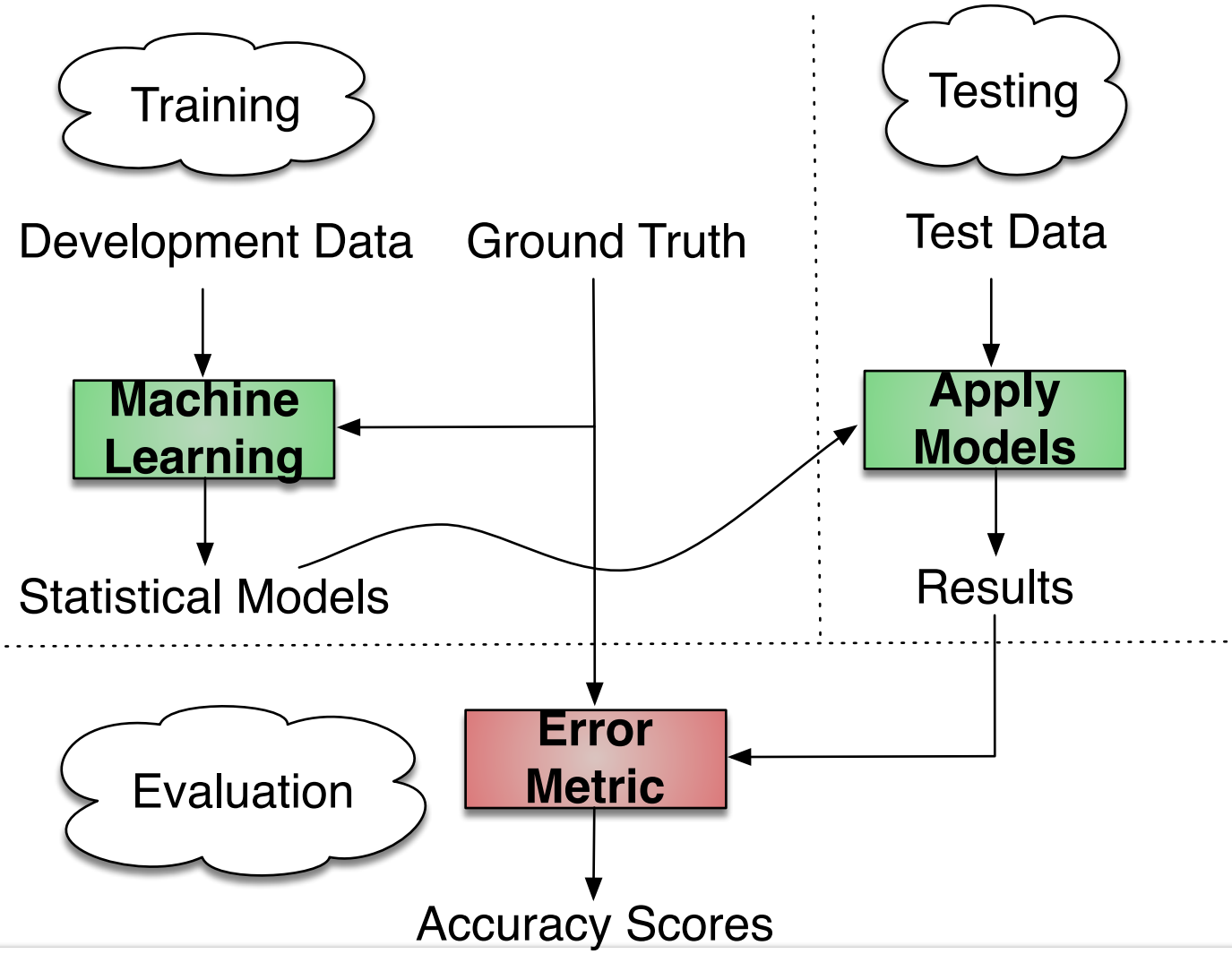
4) How well did your generalization prediction hold on the independent test data? Explain results. How confident are you in the results?

# Project Questions 9-10 (Finishing Touch)

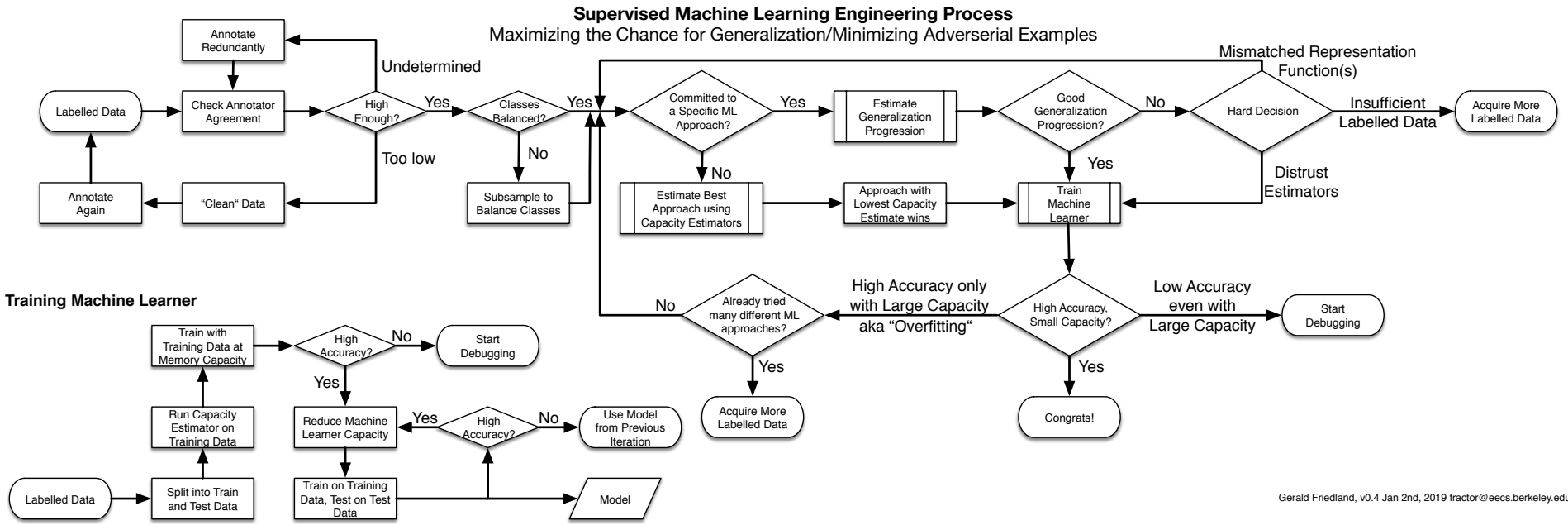
1) How do you combine the models of the modalities? Explain your choice. How confident are you in the combination results (ie., does it make sense to combine)?

2) What are the final combined results of the system? Are the experiments documented and repeatable (if not, please make sure they are, even for bad results)? Are the experiments reproducible (speculate)?

# Generic Project Workflow for Accuracy



# Generic Project Workflow for Generalization




Gerald Friedland, v0.4 Jan 2nd, 2019 fractor@eecs.berkeley.edu

# Project, Last Lecture?

**Your Questions?**

# Generalization

**1) What is generalization (intuitively)?**

**2) Formal definition**

**3) Can we universally generalize?**

# Abstraction

- **Detail removal**

  **"The act of leaving out of consideration one or more properties of a complex object so as to attend to others."**

- **Generalization**

  **"The process of formulating general concepts by abstracting common properties of instances"**

- **Technical terms:**
  **Compression, Quantization, Clustering, Unsupervized Learning**

**Henri Matisse "Naked Blue IV"**

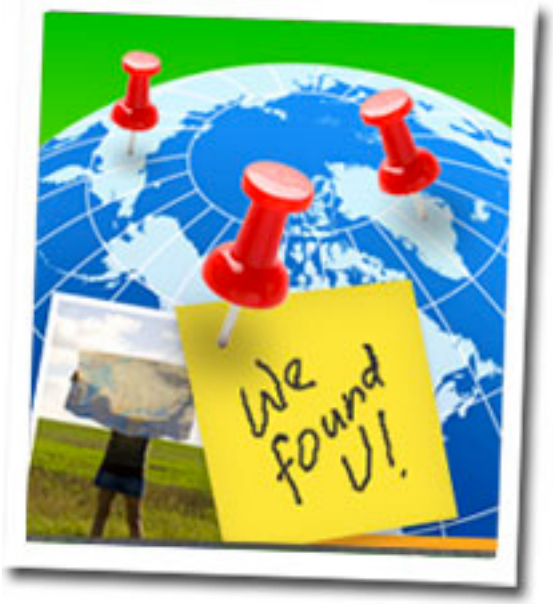
# Experiment

Standard Time Zones of the World
WHERE ARE YOU FROM?
CANADA
UNITED STATES
MEXICO
BRAZIL
ARGENTINA
CHILE
PERU
BOLIVIA
PARAGUAY
URUGUAY
COLOMBIA
VENEZUELA
ECUADOR
RUSSIA
CHINA
INDIA
AUSTRALIA
INDONESIA
PHILIPPINES
JAPAN
MONGOLIA
KAZAKHSTAN
ALGERIA
LIBYA
EGYPT
SUDAN
ETHIOPIA
ZAIRE
NIGERIA
SOUTH AFRICA
NEW ZEALAND
North Atlantic Ocean
South Atlantic Ocean
North Pacific Ocean
South Pacific Ocean
Indian Ocean
Coordinated Universal Time (U.T.C.) formerly Greenwich Mean Time
International Date Line

# Where are you from?

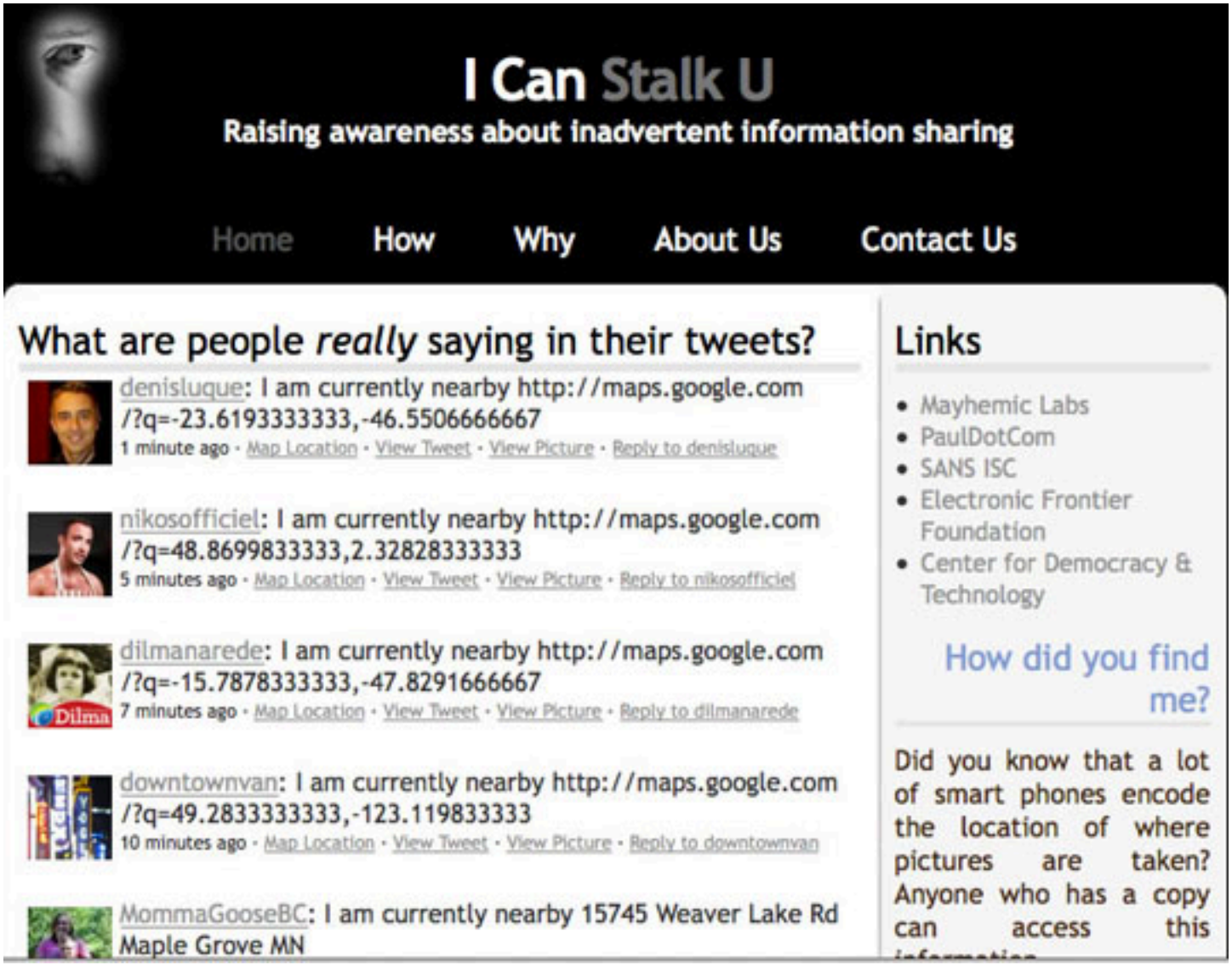
**Possible Answers:**

- China
- California
- The Bay Area
- San Mateo
- 1947 Center Street, Berkeley, CA
- 37.8693° N, 122.2696° W



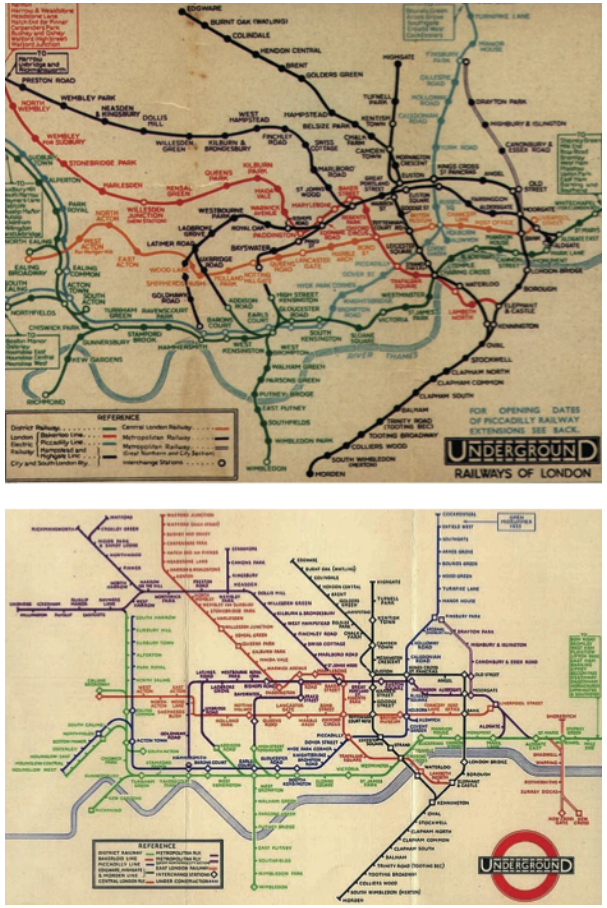

**All correct but different levels of abstraction!**

# Abstraction gone wrong!



# Detail Removal (in Data Science)

- You'll want to look at only the interesting data, leave out the details, zoom in/out…
- Abstraction is the idea that you focus on the essence, the cleanest way to map the messy real world to one you can build
- Experts are often brought in to know what to remove and what to keep!




The London Underground 1928 Map & the 1933 map by Harry Beck.

# The Power of Abstraction, Everywhere!

- **Examples:**
  - **Functions (e.g., sin x)**
  - **Hiring contractors**
  - **Application Programming Interfaces (APIs)**
  - **Technology (e.g., cars)**
- **Amazing things are built when these layer**
  - **And the abstraction layers are getting deeper by the day!**

We only need to worry about the interface, or specification, or contract NOT how (or by whom) it's built

Above the abstraction line

**Abstraction Barrier (Interface)**
(the interface, or specification, or contract)

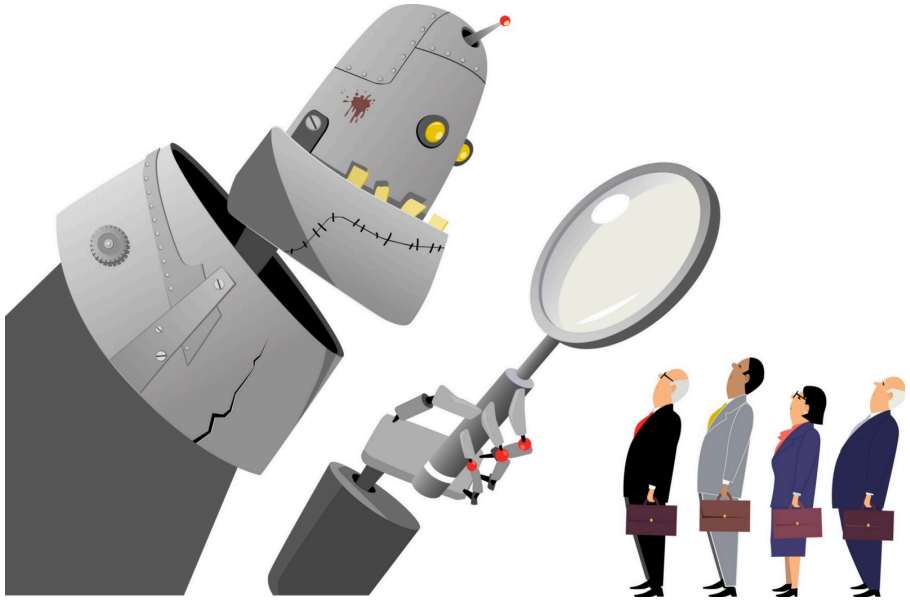
Below the abstraction line

This is where / how / when / by whom it is actually built, which is done according to the interface, specification, or contract.

# Abstraction: Pitfalls

- Abstraction is not universal without loss of information (mathematically provable). This means, in the end, the complexity can only be "moved around"

- Abstraction makes us forget how things actually work and can therefore hide bias. Example: AI and hiring decisions.



- Abstraction makes things special and that creates dependencies. Dependencies grow longer and longer over time and can become unmanageable.

# Formal Definition: Generalization

See whiteboard. Quick recap:

$$\forall x, x' \exists \delta \text{ such that } |x - x'| < \delta \implies f(x) = f(x')$$

$x \in$ Training data

$x' \in$ Test data

$|\circ|$ semi-metric

$f$ machine learner

# That’s it for today.

**Questions?**