Experimental Design for Machine Learning on Multimedia Data
Lecture 5

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

Website: http://www.icsi.berkeley.edu/~fractor/fall2019/

Discuss Homework

Project proposal deadline: October 11th, 2019.

Email project proposals to me and Rishi.

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: Is there enough data?

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)?

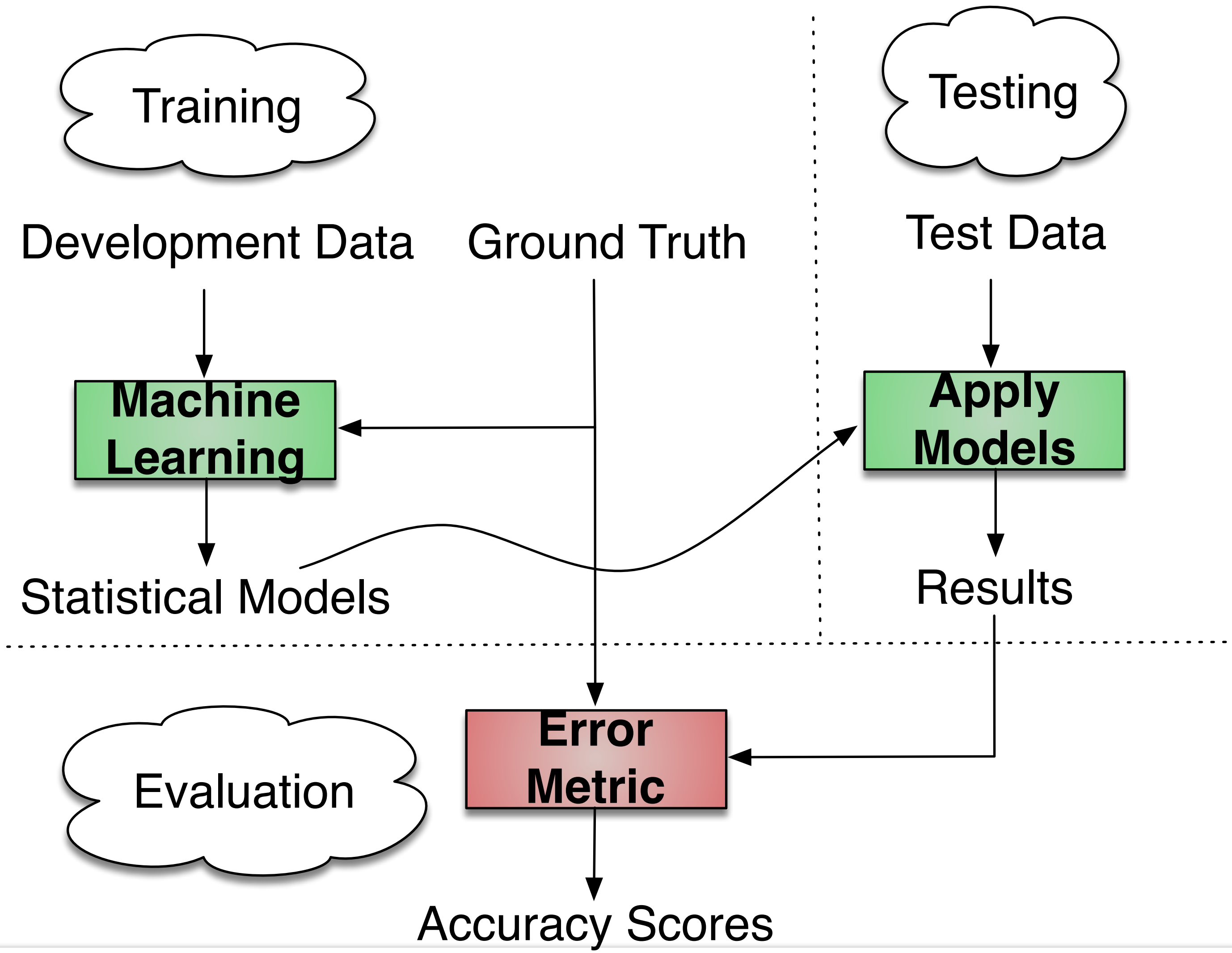
4) How well did your generalization prediction hold on the independent test data? Explain results. How confident are you in the results?

Project Questions 9-10 (Finishing Touch)

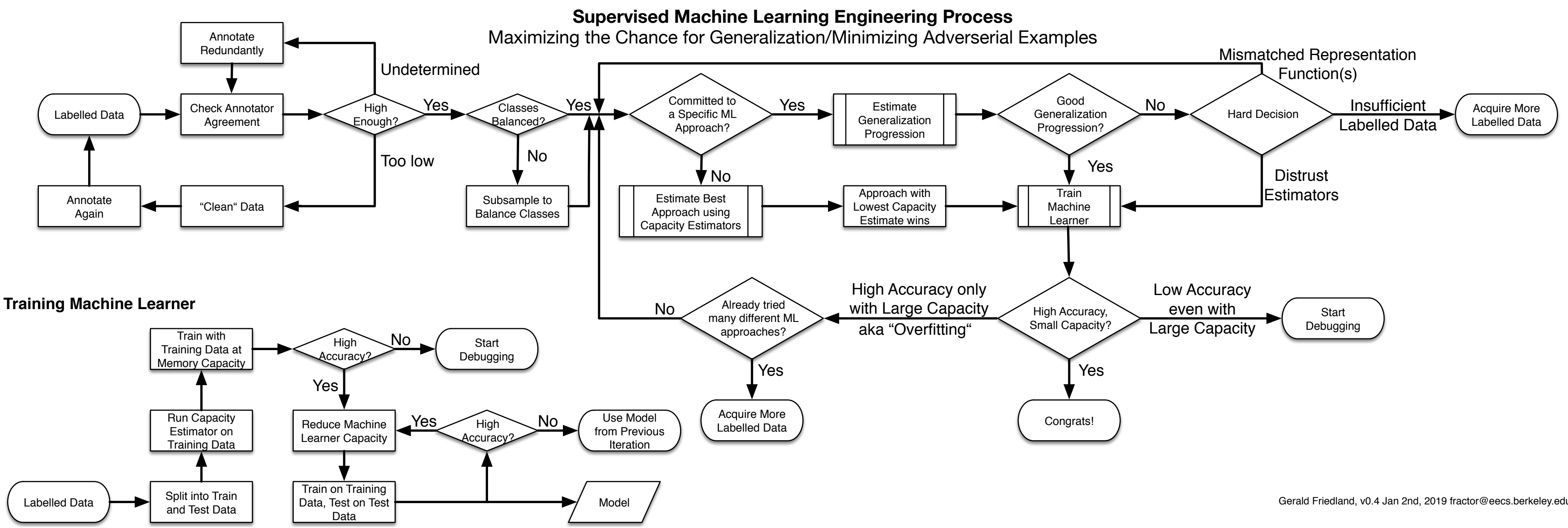
1) How do you combine the models of the modalities? Explain your choice. How confident are you in the combination results (ie., does it make sense to combine)?

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



Conclusions so far

- The lower limit of generalization is memorization. This is, the upper limit for the size of a machine learner is it’s memory capacity.
- The memory capacity is measurable in bits.
- Using a machine learner that is over capacity is a waste of resources and increases the risk of failure!
- Alchemy converted into chemistry by measuring: It’s time to convert guessing and checking in Machine Learning into science! Let’s call it data science?

- **Todi=o:**
  - Non-Binary classifiers, regression
  - Convolutional networks, other machine learners
  - Re-thinking training
  - Explainable adversarial examples

Predicting Capacity Requirements

Given data and labels: How much actual capacity do I need to memorize the function?

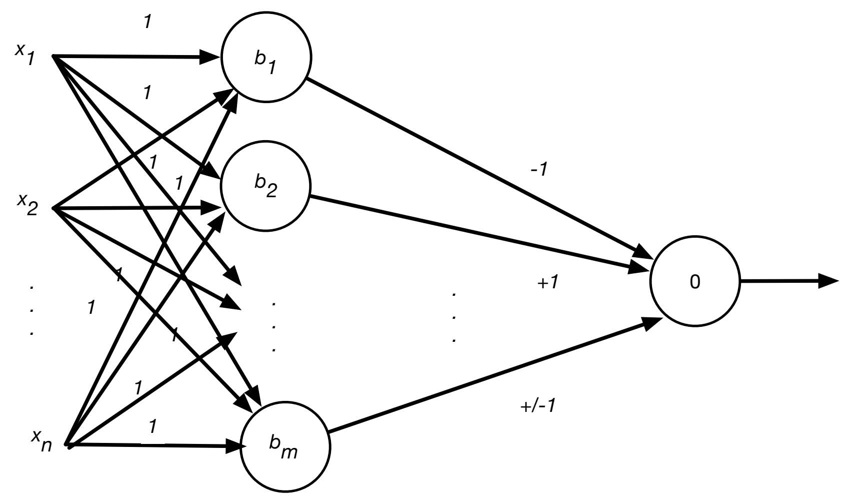
Theoretical answer: What is the minimum description length of the table representing the function f (this is, Shannon Entropy).

Practical Answer:

1) Worst case: Let's build a neural network where only the biases are trained
2) Expected case: How much parameter reduction can (exponential) training buy us?

Predicting Maximum Memory Capacity

data: array of length i containing vectors x with dimensionality d
labels: a column containing 0 or 1
$MaxCapReq(data, labels)$
$thresholds \leftarrow 0$
loop over i: $table[i] \leftarrow (\sum x[i][d], label[i])$
$sortedtable \leftarrow sort(table, key = column\ 0)$
$class \leftarrow 0$
loop over i: **if** *not* $sortedtable[i][1] == class$ **then**
    $class \leftarrow sortedtable[i][1]$
    $thresholds \leftarrow thresholds + 1$
end
$maxcapreq \leftarrow thresholds * d + thresholds + 1$
$expcapreq \leftarrow \log_2(thresholds + 1) * d$
print "Max: "$+maxcapreq+$" bits"
print "Exp: "$+expcapreq+$" bits"



"Dumb" Network

Runtime: *O(n log n)*

Predicting Memory Capacity

Dumb Network:

- Highly inefficient.
- Potentially not 100% accurate (hash collisions).
- We can assume training weights (and biases) gets 100% accuracy while reducing parameters.

Expected Reduction: Exponential!

n thresholds should be able to be represented with $log_2\, n$ weights and biases (search tree!).

Empirical Results

Dataset	Max Capacity Requirement	Expected Capacity Requirement	Validation (% accuracy)
AND, 2 variables	4 bits	2 bits	2 bits (100%)
XOR, 2 variables	8 bits	4 bits	7 bits (100%)
Separated Gaussians (100 samples)	4 bits	2 bits	3 bits (100%)
2 Circles (100 samples)	224 bits	12 bits	12 bits (100%)
Checker pattern (100 samples)	144 bits	12 bits	12 bits (100%)
Spiral pattern (100 samples)	324 bits	14 bits	24 bits (98%)
ImageNet: 2000 images in 2 classes	906984 bits	10240 bits	10253 bits (98.2 %)

All results repeatable at: https://github.com/fractor/nntailoring

From Memorization to Generalization

Memorization is worst-case generalization.

Good news:

- Real-world data is not random.
- The information capacity of a perceptron is usually >1bit per parameter (Cover, MacKay).

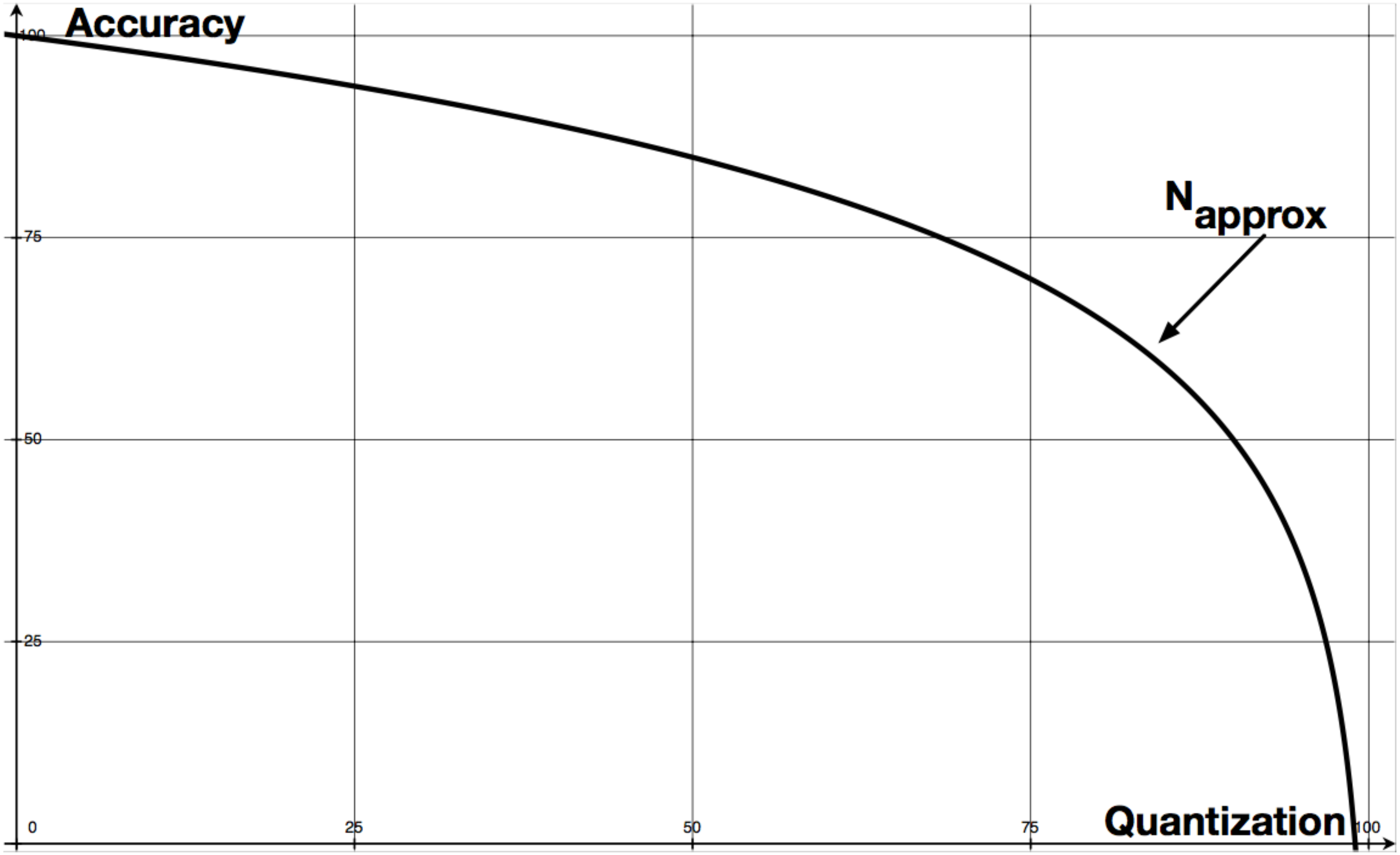
This means, we should be able to use less parameters than predicted by memory capacity calculations.

Suggested Engineering Process for Generalization

- Start at approximate expected capacity.
- Train to >98% accuracy. If impossible, increase parameters.
- Retrain iteratively with decreased capacity while testing against validation set.
  Should see: decrease in training accuracy with increase in validation set accuracy
- Stop at minimum capacity for best held-out set accuracy.

Best case scenario: As parameters are reduced, neural network fails to memorize only the insignificant (noise) bits.

Generalization Process: Expected Curve



Overcapacity Machine Learning: Issues

- Waste of money, energy, and time. Bad for environment.

- Unseen (redundant bits) are a necessary condition for adversarial examples. See:
  B. Li, G. Friedland, J. Wang, R. Jia, C. Spanos, D. Song: "*One Bit Matters: Explaining Adversarial Examples as the Abuse of Redundancy*", submitted to ICLR 2019.

- Less parameters give a higher chance for explainability (Occam's Razor). See:
  G. Friedland, A. Metere: "*Machine Learning for Science*", UQ SciML Workshop, Los Angeles, June 2018.

Reminder: Occam’s Razor

Among competing hypotheses, the one with the fewest assumptions should be selected.

For each accepted explanation of a phenomenon, there may be an extremely large, perhaps even incomprehensible, number of possible and more complex alternatives, because one can always burden failing explanations with ad hoc hypotheses to prevent them from being falsified; therefore, simpler theories are preferable to more complex ones because they are more testable.

(Wikipedia, Sep. 2017)

Demo time!

- Intro to tools on github
  - T(n,k) calculation
  - Capacity estimation given table
  - Capacity progression