



Machine Learning Systems Design

Lecture 3: Training Data

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingeniería Electrónica

Universidad Popular del Cesar

Agenda

1. Mind vs. data
2. Labeling
3. Breakout exercise
4. Sampling
5. Class imbalance

1. Mind vs. Data

WHO WOULD WIN?

Intelligent model architectures
that took researchers their
entire PhDs to design

Terabytes of data
scraped from Reddit in
a week

Poll: who would win?

1. Intelligent design
2. TB of Reddit data

WHO WOULD WIN?

Intelligent model architectures
that took researchers their
entire PhDs to design

1. Who would win?

Intelligent model architectures that took researchers their entire PhDs to design. 15%

Terabytes of data scraped from Reddit in a week. 85%

Terabytes of data
scraped from Reddit in
a week

From last year

Mind

“Data is profoundly dumb.”

Judea Pearl, [Mind over data - The Book of Why](#)

Judea Pearl @yudapearl

ML will not be the same in 3-5 years, and ML folks who continue to follow the current data-centric paradigm will find themselves outdated, if not jobless. Take note.

Mehmet Suzen @memosislant · Sep 26, 2020

Replies to @yudapearl and @mattshomepage

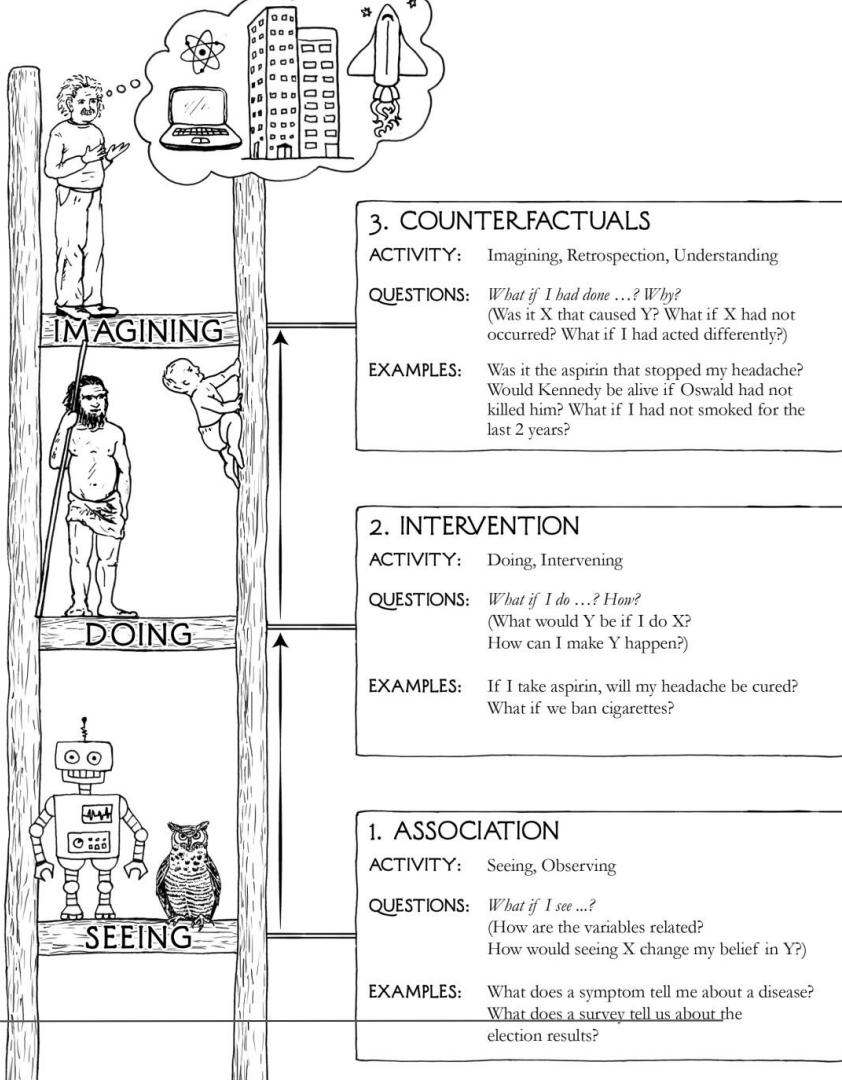
Luckily some high profile ML scientists are investing quite a lot of their research output on causality in ML. Specially arxiv.org/abs/1911.10500 Schölkopf is.mpg.de/~bs

4:32 PM · Sep 27, 2020 · Twitter Web App

119 Retweets 13 Quote Tweets 492 Likes

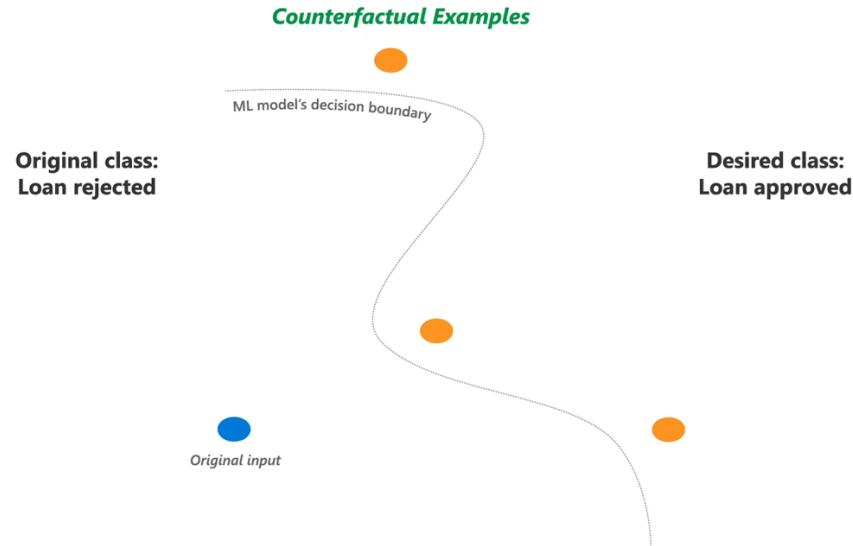
Limitations of data

- Observed X, output Y
- Can learn $P(X, Y)$
- What would happen if didn't do X?



Counterfactuals: example

- ML predicts loan app rejection
- What do I need to change to get my loan approved?



Mind

“Data is profoundly dumb.”

Judea Pearl, [Mind over data - The Book of Why](#)

Judea Pearl (@yudapearl)

ML will not be the same in 3-5 years, and ML folks who continue to follow the current data-centric paradigm will find themselves outdated, if not jobless. Take note.

Mehmet Suzen @memosisland · Sep 26, 2020
Replying to @yudapearl and @mattshomepage

Luckily some high profile ML scientists are investing quite a lot of their research output on causality in ML. Specially arxiv.org/abs/1911.10500 Schölkopf is.mpg.de/~bs

4:32 PM · Sep 27, 2020 · Twitter Web App

119 Retweets 13 Quote Tweets 492 Likes

Data

“General methods that leverage computation are ultimately the most effective, and by a large margin ... Human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation.”

Richard Sutton, [Bitter Lesson](#)

“We don’t have better algorithms. We just have more data.”

Peter Norvig, [The Unreasonable Effectiveness of Data](#)

“Imposing structure requires us to make certain assumptions, which are invariably wrong for at least some portion of the data.”

Yann LeCun, [Deep Learning and Innate Priors](#)

Data is necessary.
The debate is whether *finite** data is sufficient.

* If we had infinite data, we can solve arbitrarily complex problems by just looking up the answers.

Massive data \neq infinite data

THE DATA SCIENCE HIERARCHY OF NEEDS

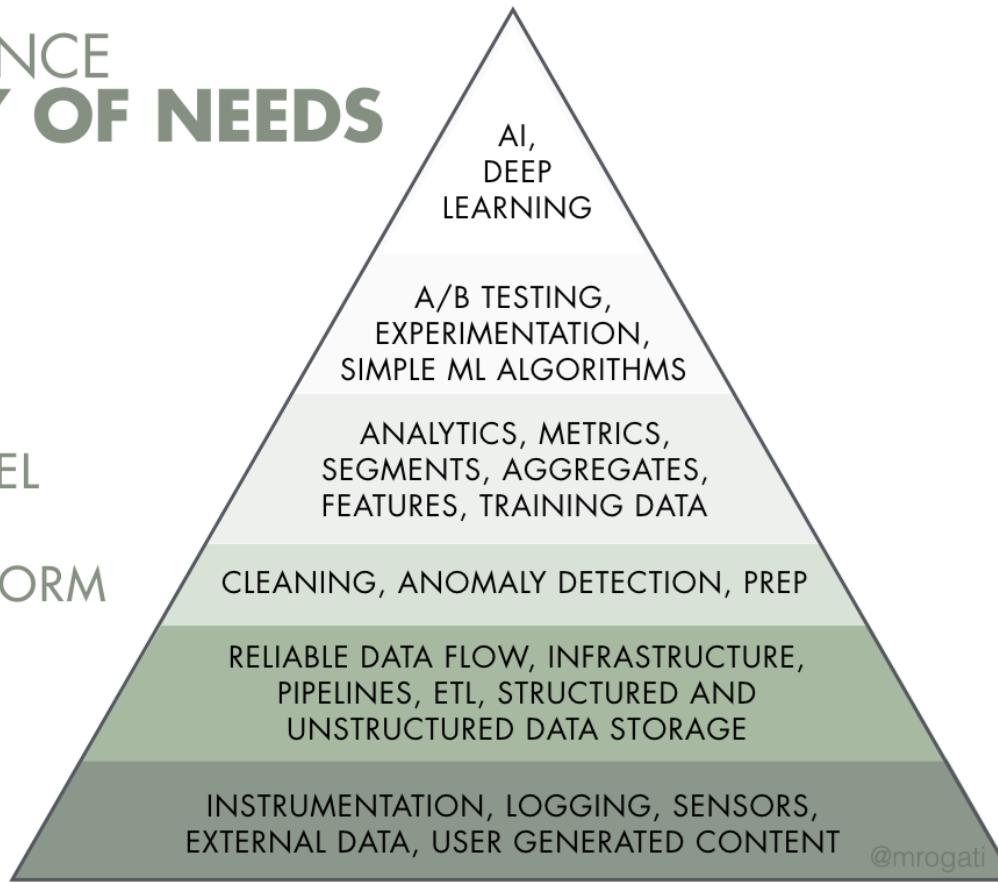
LEARN/OPTIMIZE

AGGREGATE/LABEL

EXPLORE/TRANSFORM

MOVE/STORE

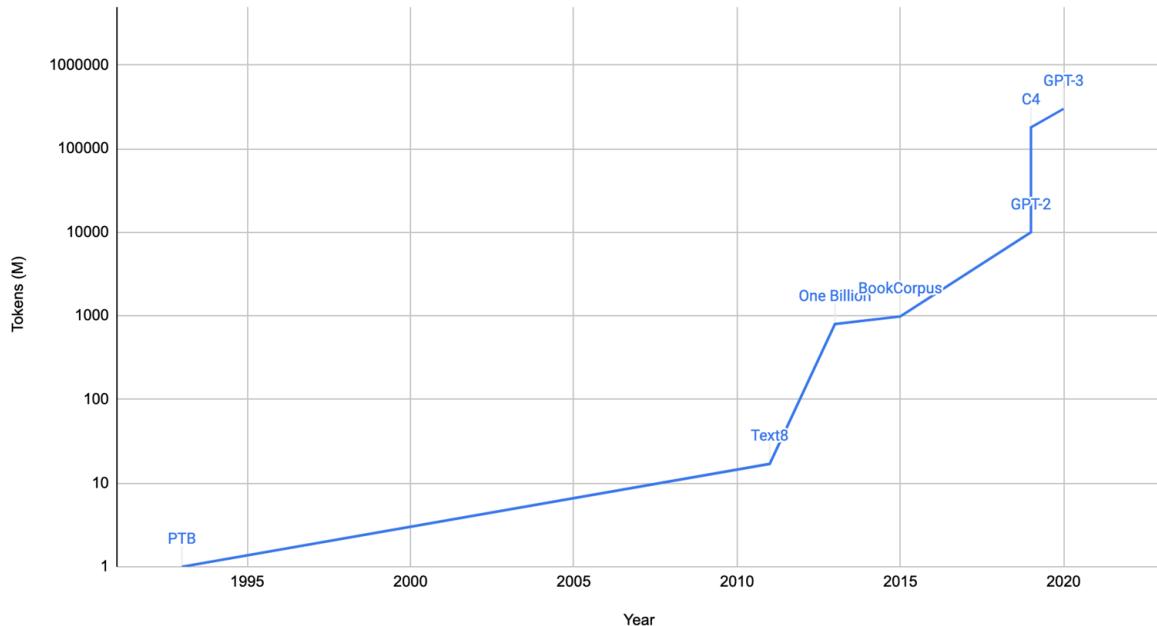
COLLECT



Datasets for language models

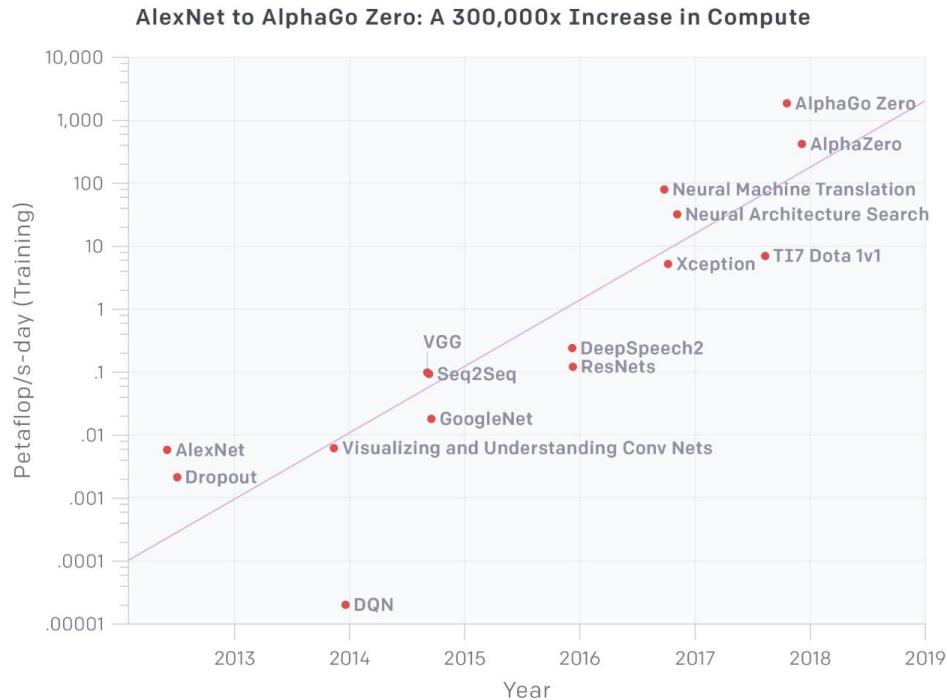
Year	Tokens (M)	Dataset
1993	1	PTB
2011	17	Text8
2013	800	One Billion
2015	985	BookCorpus
2019	10,000	GPT-2
2019	180,000	C4
2020	300,000	GPT-3

Language model datasets over time (log scale)



More data (generally) needs more compute

“amount of compute used in the largest AI training runs has doubled every 3.5 months”



⚠ Data: full of potential for biases ⚠

- sampling/selection biases
- under/over-representation of subgroups
- human biases embedded in historical data
- labeling biases
- ...

Algorithmic biases not covered (yet)!

2. Labeling

Labeling

When I told our recruiters that I wanted an in-house labeling team, they asked how long I'd need this team for. I told them: “How long do we need an engineering team for?”

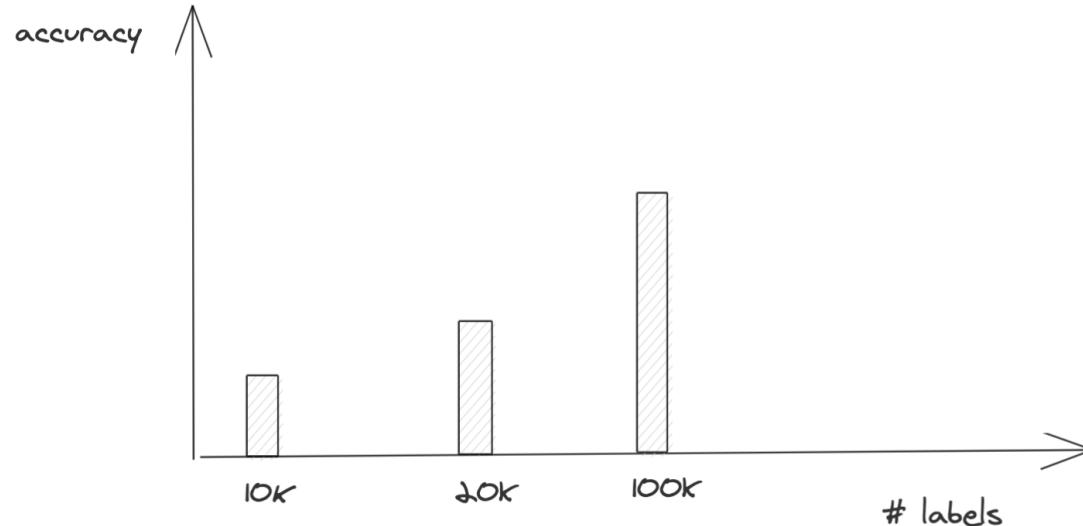


Andrej Karpathy, Director of AI @ Tesla
[CS 329S guest lecture, 2021]

Labeling

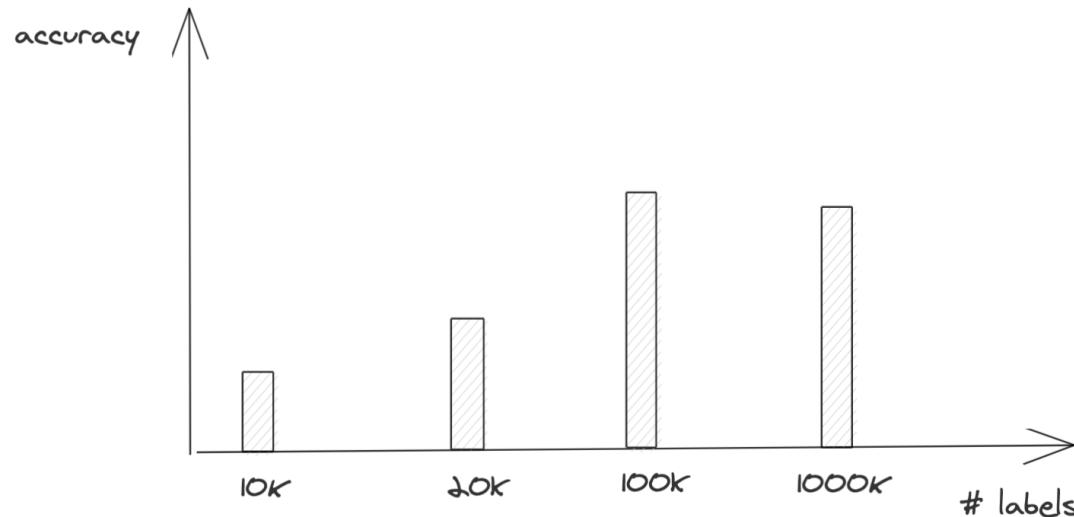
1. Hand-labeling
2. Programmatic labeling
3. Weak supervision, semi supervision, active learning, transfer learning

⚠ More data isn't always better ⚠



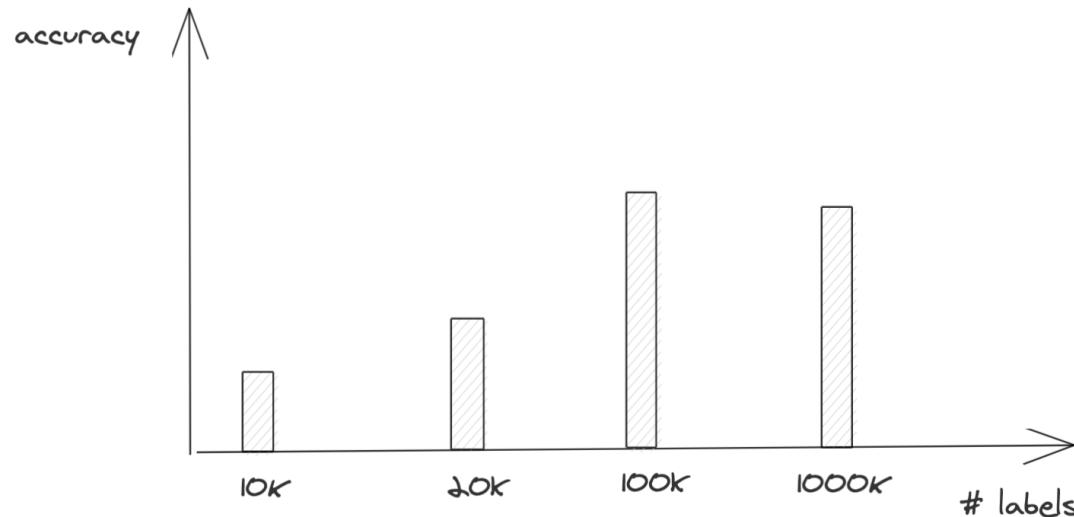
💡 Idea 💡: crowdsource data to get 1 million labels!

⚠ More data isn't always better ⚠



Why is the model getting worse?

⚠ Label sources with varying accuracy ⚠



- 100K labels: internally labeled, high accuracy
- 1M labels: crowdsourced, noisy

Label multiplicity: example

Task: label all entities in the following sentence:

Darth Sidious, known simply as the Emperor, was a Dark Lord of the Sith who reigned over the galaxy as Galactic Emperor of the First Galactic Empire.

Label multiplicity: example

which annotator is correct?

Task: label all entities in the following sentence:

Darth Sidious, known simply as the Emperor, was a Dark Lord of the Sith who reigned over the galaxy as Galactic Emperor of the First Galactic Empire.

Annotator	# entities	Annotation
1	3	[Darth Sidious], known simply as the Emperor, was a [Dark Lord of the Sith] who reigned over the galaxy as [Galactic Emperor of the First Galactic Empire]
2	6	[Darth Sidious], known simply as the [Emperor], was a [Dark Lord] of the [Sith] who reigned over the galaxy as [Galactic Emperor] of the [First Galactic Empire].
3	4	[Darth Sidious], known simply as the [Emperor], was a [Dark Lord of the Sith] who reigned over the galaxy as [Galactic Emperor of the First Galactic Empire].

Label multiplicity

More expertise required (more difficult to label), more room for disagreement!

If experts can't agree on a label, time to rethink human-level performance

Label multiplicity: solution

- Clear problem definition
 - Pick the entity that comprises the longest substring

Annotator	# entities	Annotation
1	3	[Darth Sidious], known simply as the Emperor, was a [Dark Lord of the Sith] who reigned over the galaxy as [Galactic Emperor of the First Galactic Empire]
2	6	[Darth Sidious], known simply as the [Emperor], was a [Dark Lord] of the [Sith] who reigned over the galaxy as [Galactic Emperor] of the [First Galactic Empire].
3	4	[Darth Sidious], known simply as the [Emperor], was a [Dark Lord of the Sith] who reigned over the galaxy as [Galactic Emperor of the First Galactic Empire].

Label multiplicity: solution

- Clear problem definition
- Annotation training
- Data lineage: track where data/labels come from

Label multiplicity: solution

- Clear problem definition
- Annotation training
- Data lineage: track where data/labels come from
- Learning methods with noisy labels
 - [Learning with Noisy Labels](#) (Natarajan et al., 2013)
 - [Loss factorization, weakly supervised learning and label noise robustness](#) (Patrini et al., 2016)
 - [Cost-Sensitive Learning with Noisy Labels](#) (Natarajan et al., 2018)
 - [Confident Learning: Estimating Uncertainty in Dataset Labels](#) (Northcutt et al., 2019)

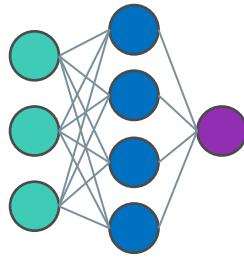
Programmatic labeling

Training data is the bottleneck

Data



Algorithms



ML Model

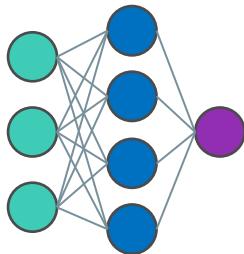
Training data is the bottleneck

Data



Key differentiator

Algorithms



ML Model

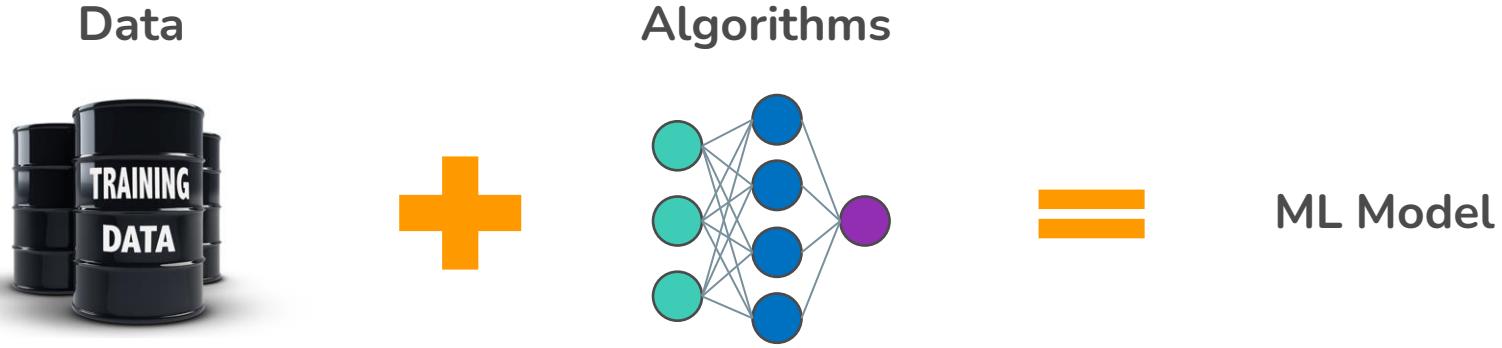
```
from transformers \\\n    import BertModel as model
```

Increasingly
commoditized

“We don’t have better algorithms. We just have more data.”

Peter Norvig, [The Unreasonable Effectiveness of Data](#)

Training data is the bottleneck



- 8 Person-months
- 8-9 pt. differences
- 1-2 days
- <1 pt. differences

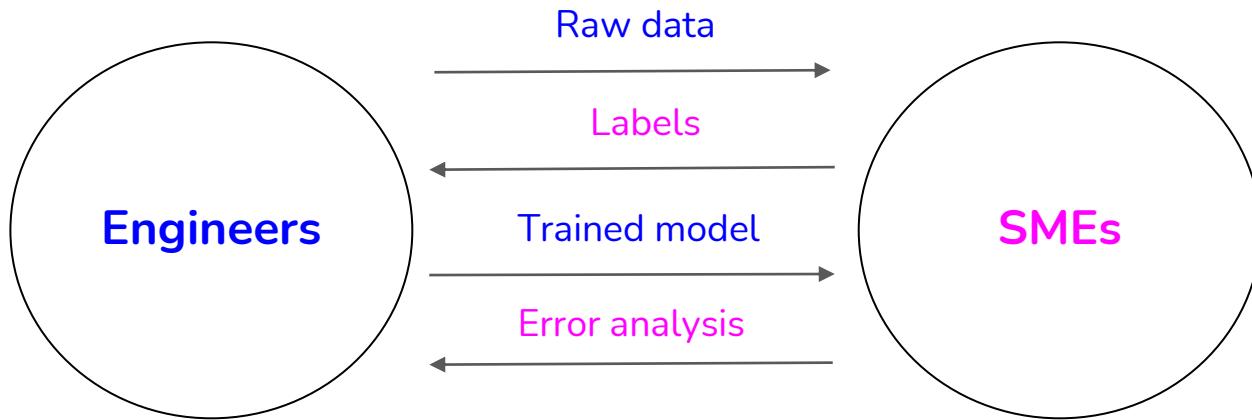
How to get training data in days?

Hand labeling data is ...



- **Expensive:** Esp. when **subject matter expertise** required
- **Non-private:** Need to ship data to human annotators
- **Slow:** Time required scales linearly with # labels needed
- **Non-adaptive:** Every change requires re-labeling the dataset

Cross-functional communication



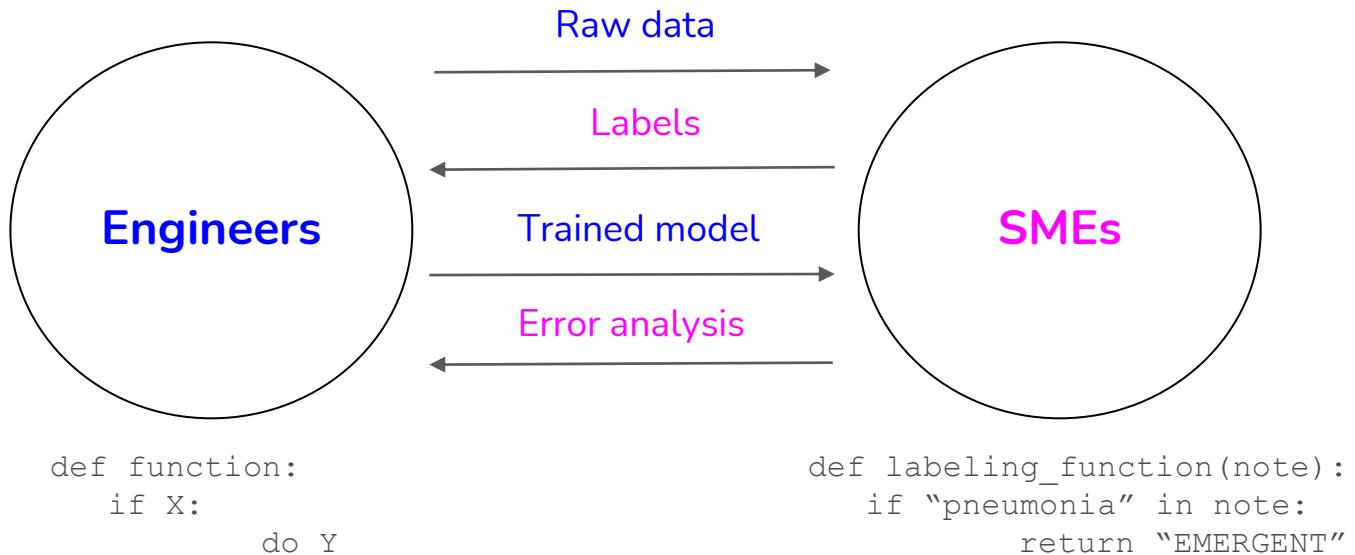
```
def function:  
    if X:  
        do Y
```

Code: version control, reuse,
share

If the nurse's note mentions serious conditions like pneumonia, the patient's case should be given priority consideration.

How to version, share, reuse
expertise?

SME as labeling functions



Labeling functions (LFs): Encode SME heuristics as function and use them to label training data *programmatically*

LFs: can express many different types of heuristics

(*)	Pattern Matching	If a phrase like “send money” is in email
	Boolean Search	If <code>unknown_sender</code> AND <code>foreign_source</code>
	DB Lookup	If sender is in our <code>Blacklist.db</code>
	Heuristics	If <code>SpellChecker</code> finds 3+ spelling errors
	Legacy System	If <code>LegacySystem</code> votes spam
	Third Party Model	If <code>BERT</code> labels an entity “diet”
	Crowd Labels	If <code>Worker #23</code> votes spam

LFs: can express many different types of heuristics



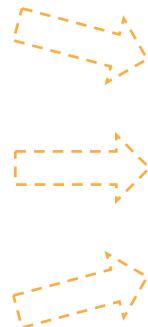
"If nurse's report says 'malignant', likely to be emergent"



"If it matches a list of patient names..."



"If our legacy model thinks it's emergent..."



Labeling functions: Simple, flexible, interpretable, adaptable, fast

LFs: powerful but noisy



```
def LF_contains_money(x):  
    if "money" in x.body.text:  
        return "SPAM"
```



```
def LF_from_grandma(x):  
    if x.sender.name is "Grandma":  
        return "HAM"
```



```
def LF_contains_money(x):  
    if "free money" in x.body.text:  
        return "SPAM"
```



From: Grandma

“Dear handsome grandson,
Since you can't be home for Thanksgiving
dinner this year, I'm sending you some
money so you could enjoy a nice meal ...”

??

“You have been pre-approved for
free **cash** ...”

??

- **Noisy:** Unknown, inaccurate
- **Overlapping:** LFs may be correlated
- **Conflicting:** different LFs give different labels
- **Narrow:** Don't generalize well

LF labels are combined to generate ground truths



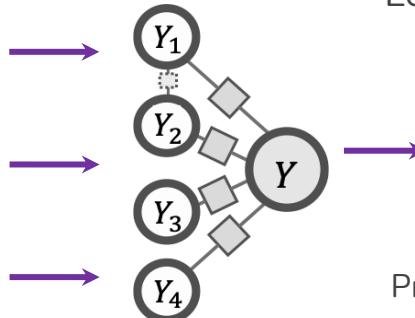
```
def LF_contains_money(x):  
    if "money" in x.body.text:  
        return "SPAM"
```



```
def LF_from_grandma(x):  
    if x.sender.name is "Grandma":  
        return "HAM"
```



```
def LF_contains_money(x):  
    if "free money" in x.body.text:  
        return "SPAM"
```



[Intuition]

Look at agreements & disagreements

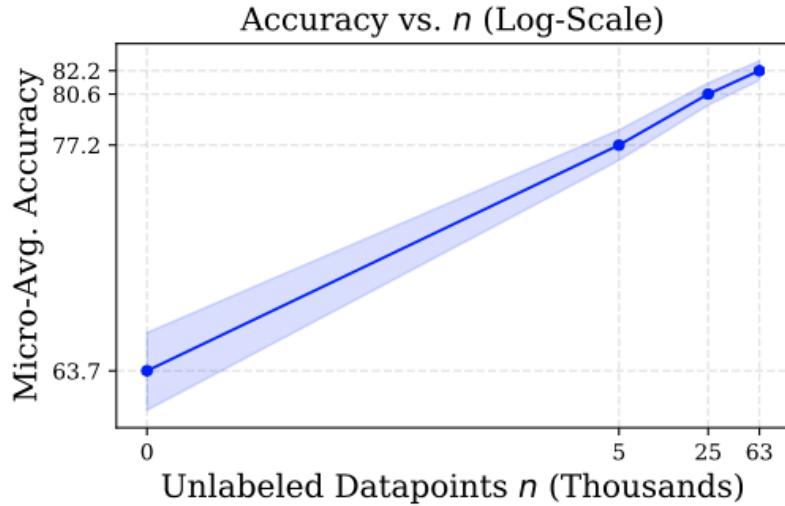
$$(\Sigma^{-1})_O = \Sigma_O^{-1} + zz^T$$

Provably consistent matrix completion-style algorithm over inverse covariance

[Ratner et. al. NeurIPS'16;
Bach et. al. ICML'17;
Ratner et. al. AAAI'19;
Varma et. al. ICML'19;
Sala et. al. NeurIPS'19;
Fu et. al. ICML'20]

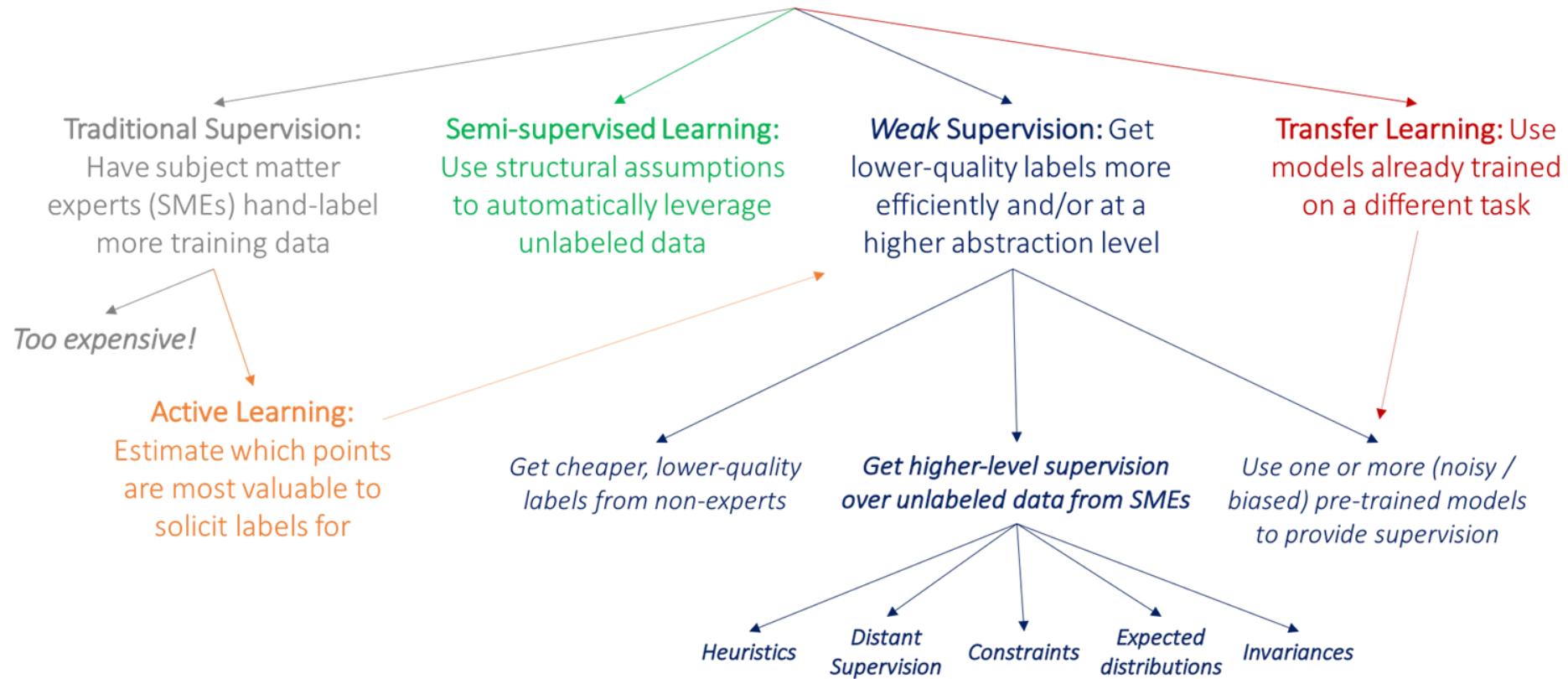
Hand labeling	Programmatic labeling
Expensive: esp. when subject matter expertise required	Cost saving: Expertise can be versioned, shared, reused across organization
Non-private: Need to ship data to human annotators	Privacy: Create LFs using a cleared data subsample then apply LFs to other data without looking at individual samples.
Slow: Time required scales linearly with # labels needed	Fast: Easily scale 1K -> 1M samples
Non-adaptive: Every change requires re-labeling the dataset	Adaptive: When changes happen, just reapply LFs!

Programmatic labeling: Scale with unlabeled data



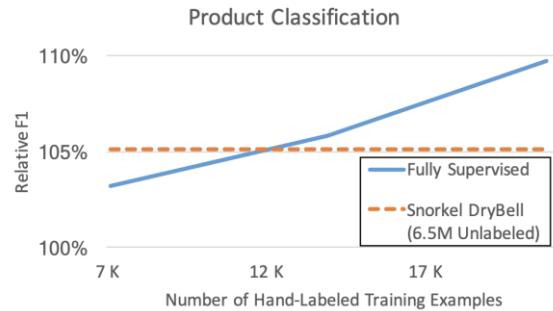
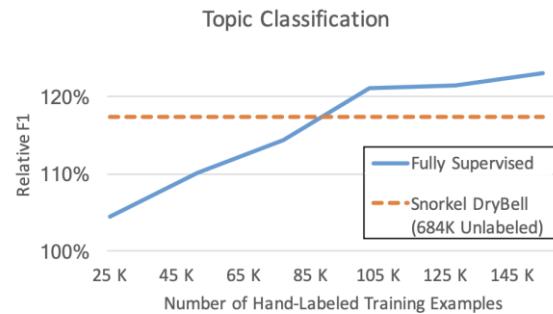
**Weak supervision, semi-supervision,
active learning,
transfer learning**

How to get more labeled training data?



Weak supervision

- Leverage noisy, imprecise sources to create labels
 - e.g. if “money” is in an email it’s probably spam



Semi-supervision

- Use structural assumptions to leverage a large amount of unlabeled data together with a small amount of labeled data
 - Hashtags in the same profile/tweet are probably of similar topics



A screenshot of a Twitter profile card for the account @MIT_CSAIL. The card features the MIT CSAIL logo (a stylized building icon with the text 'CSAIL' below it), the handle '@MIT_CSAIL' with a blue verified checkmark, and the status text 'Follows you'. To the right is a blue 'Follow' button. Below the card, the bio text reads: 'MIT's largest research lab, the Computer Science & Artificial Intelligence Lab. instagram.com/mit_csail/ #AI #ml #bigdata #iot #datascience #nlp #coding'

Semi-supervision

- Use structural assumptions to leverage a large amount of unlabeled data together with a small amount of labeled data
- Might require complex algorithms like clustering to discover similarity

Semi-supervision: self-training

1. Train model on a small set of labeled data
2. Use this model to generate predictions for unlabeled data
3. Use predictions with high raw probabilities as labels
4. Repeat step 1 with new labeled data

Semi-supervision: perturbation-based methods

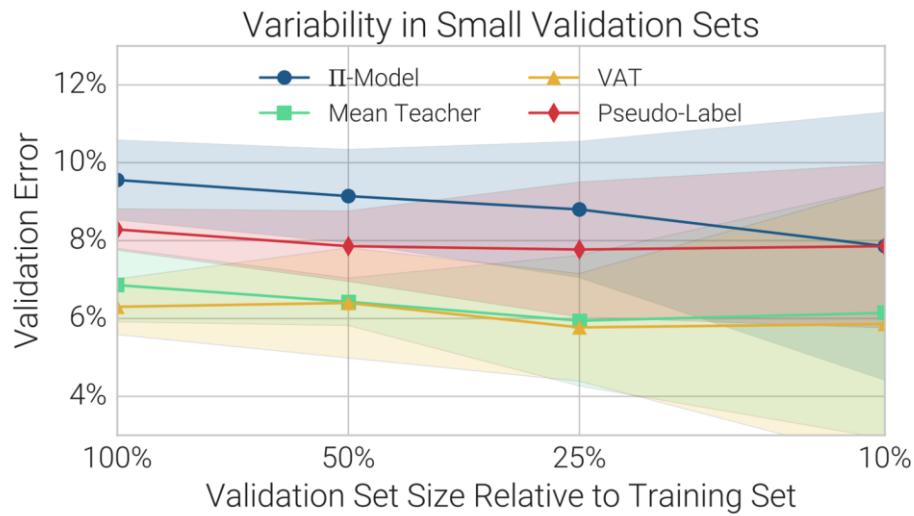
Assumption: small perturbation wouldn't change a sample's label

- Add white noises to images
- Add small values to word embeddings

Also a data augmentation method!

Semi-supervision challenge: valid set's size

- Big valid set: less data for training
- Small valid set: not enough signal to choose the best model



Transfer learning

Apply model trained for one task to another task

1. Fine-tuning
2. Prompt-based

Transfer learning: fine-tuning

- fine-tuning only some layers
- fine-tuning the entire model

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Transfer learning: Prompt-based

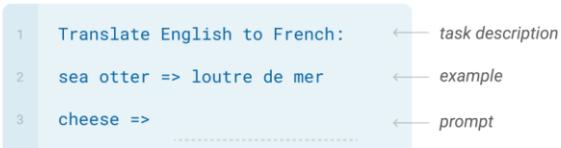
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



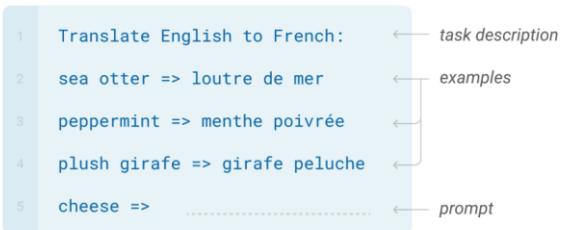
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Demo

Use GPT-3 to generate React apps

<https://twitter.com/sharifshameem/status/1284095222939451393>

Active learning

- **Goal:** Increase the efficiency of labels
- Label samples that are estimated to be most valuable to the model according to some metrics

Active learning metrics

- Uncertainty measurement
 - e.g. label samples with lowest raw probability for the predicted class

Active learning metrics

- Uncertainty measurement
- Candidate models' disagreement
 - Have several candidate models (e.g. models with different hyperparams)
 - Each model makes its own prediction
 - Label samples with most disagreement

Active learning

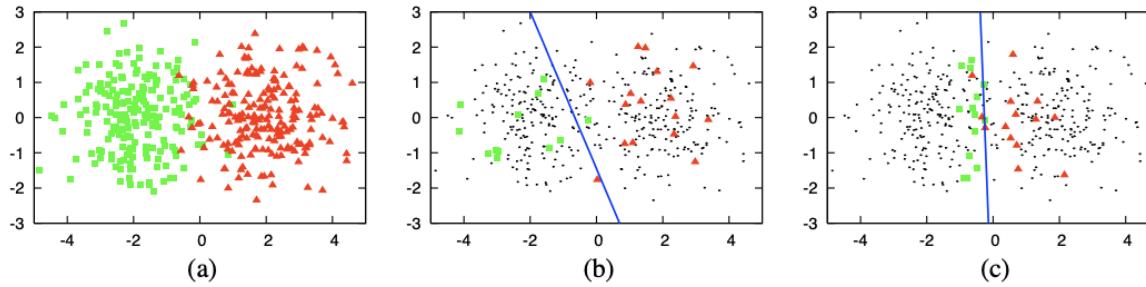


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem domain. The line represents the decision boundary of the classifier (70% accuracy). (c) A logistic regression model trained with 30 actively queried instances using uncertainty sampling (90%).

Method	How	Ground truths required?
Weak supervision	Leverages (often noisy) heuristics to generate labels	No, but a small number of labels is useful to guide the development of heuristics
Semi-supervision	Leverages structural assumptions to generate labels	Yes. A small number of initial labels as seeds to generate more labels
Transfer learning	Leverages models pretrained on another task for your new task	No for zero-shot learning Yes for fine-tuning, though # GTs required is often much less than # GTs required if training from scratch.
Active learning	Labels data samples that are most useful to your model	Yes

3. Breakout exercise

Sampling exercise (group of 5, 10 minutes)

You want to build a model to classify whether a tweet spreads misinformation.

- **10M tweets** from **10K users** over the last **24 months**
- # tweets/user follows a long-tail distribution
- You estimate 1% of tweets are misinformation

Questions

1. How would you sample 100K tweets to label?
2. You get 100K labels from 20 annotators and want to look at some labels to estimate the quality.
 - a. How many labels would you look at?
 - b. How would you sample?
3. Imagine you have a stream of unknown number of tweets coming in, can't fit all in memory. How to sample 10K tweets such that each tweet has an equal chance of being selected?

4. Sampling

Types of sampling

- Non-probability sampling
 - Convenience sampling: selection based on availability
 - Soliciting response
 - Choosing existing datasets
 - Looking at available reviews on Amazon
 - Snowball sampling: future samples are selected based on existing samples
 - E.g. to scrape legit Twitter accounts, start with seed accounts then scrape their following
 - Judgment sampling: experts decide what to include
 - Quota sampling: quotas for certain slices of data (no randomization)
 -

Data used in ML is mostly driven by convenience

- Language models: BookCorpus, CommonCrawl, Wikipedia, Reddit links
- Sentiment analysis: IMDB, Amazon
 - Only users who have access to the Internet and are willing to put reviews online
- Self-driving cars: most data is from the Bay Area (CA) and Phoenix (AZ)
 - Very little data on raining & snowing weather

⚠ Lots of biases in data! ⚠

Types of sampling

- Non-probability sampling
- Random sampling
 - Simple random sampling
 - Stratified sampling
 - Weighted sampling
 - Importance sampling
 - Reservoir sampling
 - ...

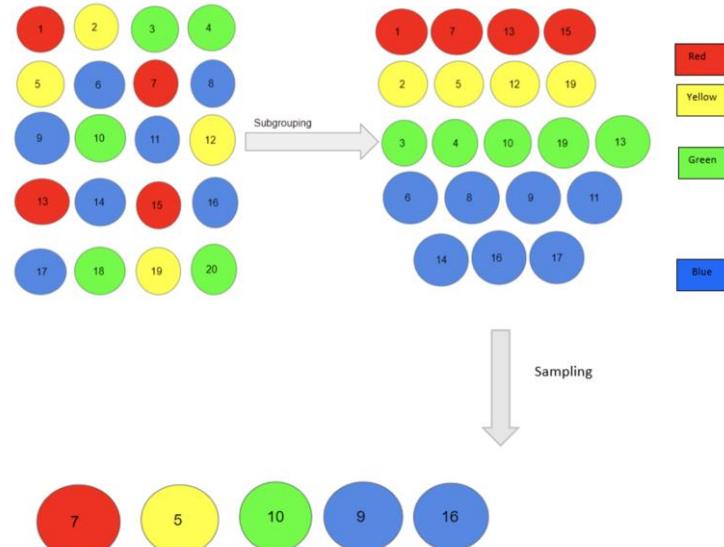
Simple random sampling

- Each sample in **population** has an equal chance of being selected
 - E.g. select 10% of all samples in population

Pros	Cons
<ul style="list-style-type: none">• Simple (easiest type of random sampling)	<ul style="list-style-type: none">• No representation guarantee: might exclude rare classes (black swan!)

Stratified sampling

- Divide population by subgroups
 - Slices of data
 - 20% of each age group: 18-24, 25-34, 35+, etc.
 - Classes
 - 2% of each class



Pros	Cons
Minor groups are represented	<p>Can't be used when:</p> <ul style="list-style-type: none">● samples can't be put into subgroups● samples can belong in multiple subgroups (multilabel)

Weighted sampling

- Each element is given a weight, which determines the probability of being selected.
 - If you want to select a sample 30% of the time, give it 3/10 weight
- Might embed domain knowledge
 - E.g. know distribution of your target population or want to prioritize recent samples

```
random.choices(population=[1, 2, 3, 4, 100, 1000],  
               weights=[0.2, 0.2, 0.2, 0.2, 0.1, 0.1],
```

```
k=2)
```



```
random.choices(population=[1, 1, 2, 2, 3, 3, 4, 4, 100, 1000],  
               k=2)
```

Importance sampling

- Useful when sampling from $P(x)$ is expensive, slow, or infeasible
 - Sample $x \sim Q(x)$ ← easier to sample from
 - Weight by $P(x)/Q(x)$
- Δ Calculating $P(x)/Q(x)$ might be expensive Δ

$$E_{P(x)}[x] = \sum_x P(x) x = \sum_x Q(x) x \frac{P(x)}{Q(x)} = E_{Q(x)}\left[x \frac{P(x)}{Q(x)}\right]$$

Importance sampling

- Useful when sampling from $P(x)$ is expensive, slow, or infeasible
 - Sample $x \sim Q(x)$ ← easier to sample from
 - Weight by $P(x)/Q(x)$
- ⚠ Calculating $P(x)/Q(x)$ might be expensive ⚠
- E.g. essential for RL
 - Too expensive to estimate reward under new policy, so use old policy

$$E_{P(x)}[x] = \sum_x P(x) x = \sum_x Q(x) x \frac{P(x)}{Q(x)} = E_{Q(x)}\left[x \frac{P(x)}{Q(x)}\right]$$

$$\nabla_{\theta} J(\theta) = E_{\tau \sim \bar{\pi}_{\theta}(\tau)} \left[\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta} (\mathbf{a}_t | \mathbf{s}_t) \left(\prod_{t'=1}^t \frac{\pi_{\theta} (\mathbf{a}_{t'} | \mathbf{s}_{t'})}{\bar{\pi}_{\theta}(\mathbf{a}_{t'} | \mathbf{s}_{t'})} \right) \left(\sum_{t'=t}^T r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) \right) \right]$$

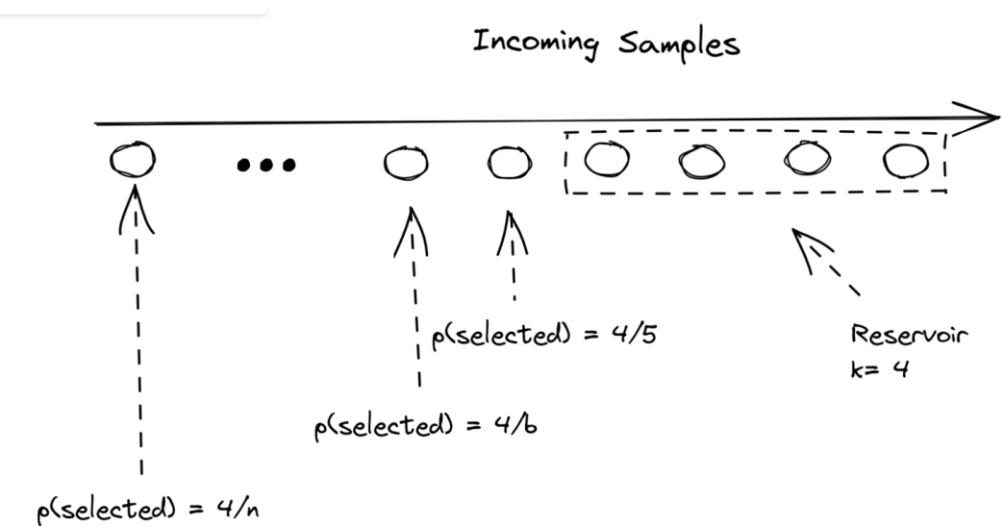
use old policy to sample data old policy

Reservoir sampling: problem

- Need select k samples from a stream of n samples with equal probability
 - n is unknown
 - impossible/inefficient to fit all in memory
- Can stop the stream any moment and get the required samples

Reservoir sampling: solution

1. First k elements are put in reservoir
2. For each incoming i^{th} element, generate a random number j between 1 and i
 - a. If $1 \leq j \leq k$: replace j^{th} in reservoir with i^{th}
3. Each incoming element has k/i chance of being in reservoir!



Also checkout algorithm L (based on geometric distribution)

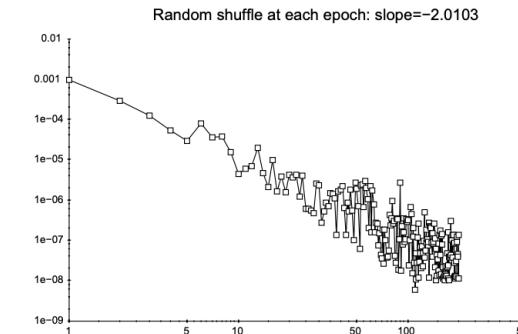
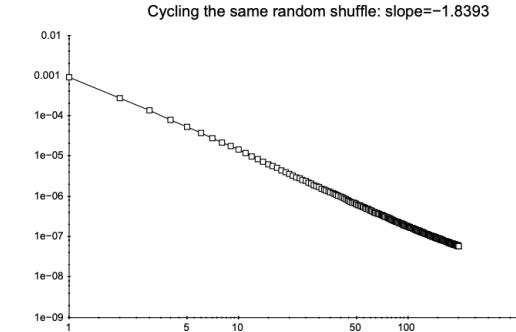
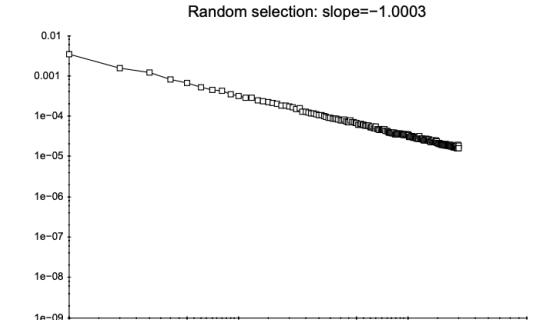
With vs. without replacement

With replacement	Without replacement
Same item can be chosen more than once	Same item can't be chosen more than once
<ul style="list-style-type: none">• No covariance between two chosen samples• Approximate true population distribution	<ul style="list-style-type: none">• Covariance between two chosen samples• Covariance reduced as dataset size becomes large
Bagging	Mini-batch gradient descent

Why do we use epochs instead of just sampling with replacement from the entire dataset?

With vs. without replacements

Because empirically it's converged faster
proven for strongly convex loss functions

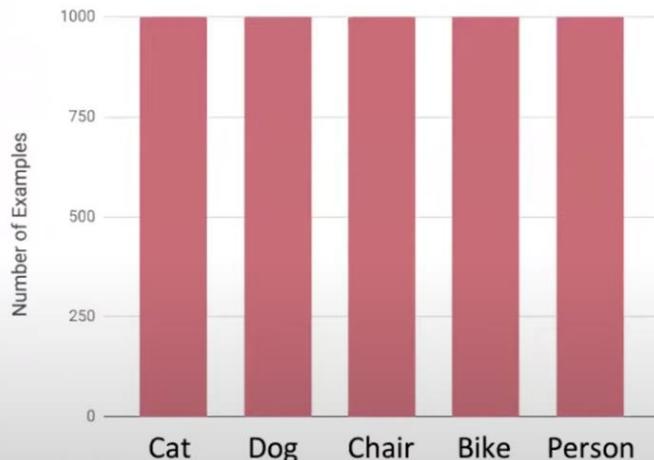


5. Class imbalance

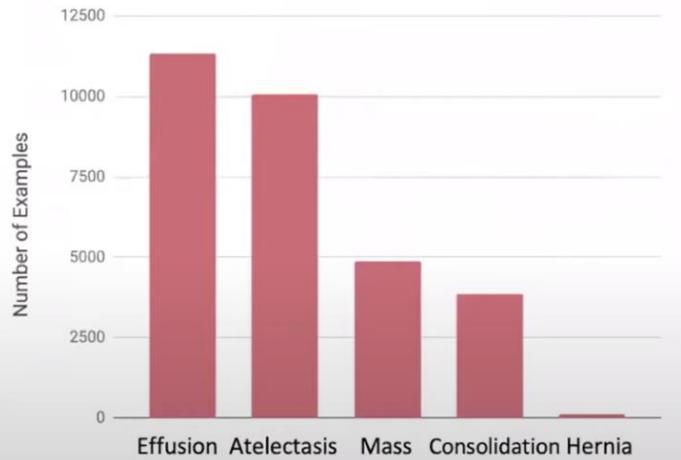
Class imbalance

Small data and rare occurrences

ML works well when the data distribution is this:



Not so well when it is this:



Why is class imbalance hard?

- Not enough signal to learn about rare classes

Why is class imbalance hard?

- Not enough signal to learn about rare classes
- Statistically, predicting majority label has higher chance of being right
 - If a majority class accounts 99% of data, always predicting it gives 99% accuracy

Computer Facts
@computerfact

concerned parent: if all your friends jumped off a bridge would you follow them?
machine learning algorithm: yes.

3:20 PM · Mar 15, 2018 · Twitter Web Client

7.2K Retweets 292 Quote Tweets 14.6K Likes

Why is class imbalance hard?

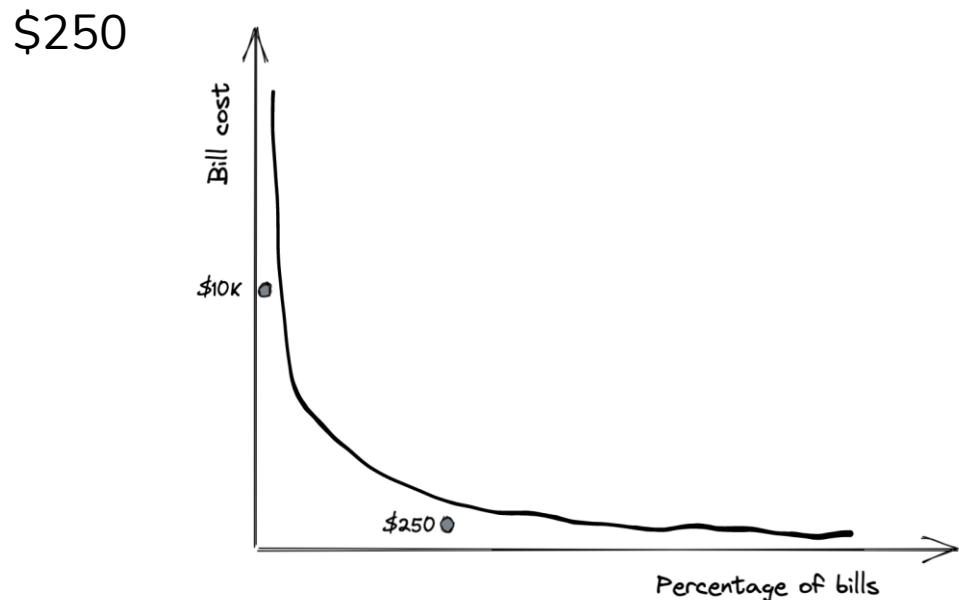
- Not enough signal to learn about rare classes
- Statistically, predicting majority label has higher chance of being right
- Asymmetric cost of errors: different cost of wrong predictions

Why is class imbalance hard?

- Not enough signal to learn about rare classes
- Statistically, predicting majority label has higher chance of being right
- Asymmetric cost of errors: different cost of wrong predictions

Asymmetric cost of errors: regression

- 95th percentile: \$10K
- Median:



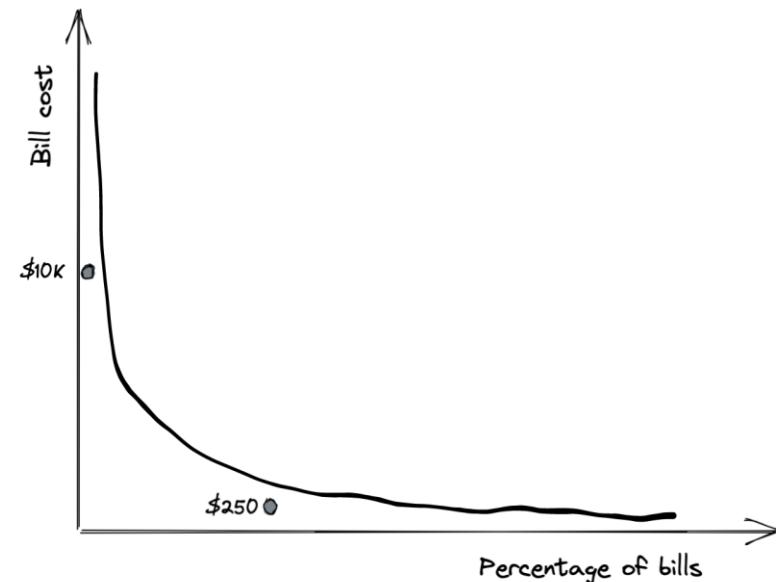
Asymmetric cost of errors: regression

100% error difference

Not OK

- \$10K bill: off by \$10K
- \$250 bill: off by \$250

OK



Machine Learning Systems Design

Next class: Feature Engineering