Machine Learning Systems Design

Lecture 3: Training Data

Stanford University

CS 329S (Chip Huyen, 2022) | cs329s.stanford.edu

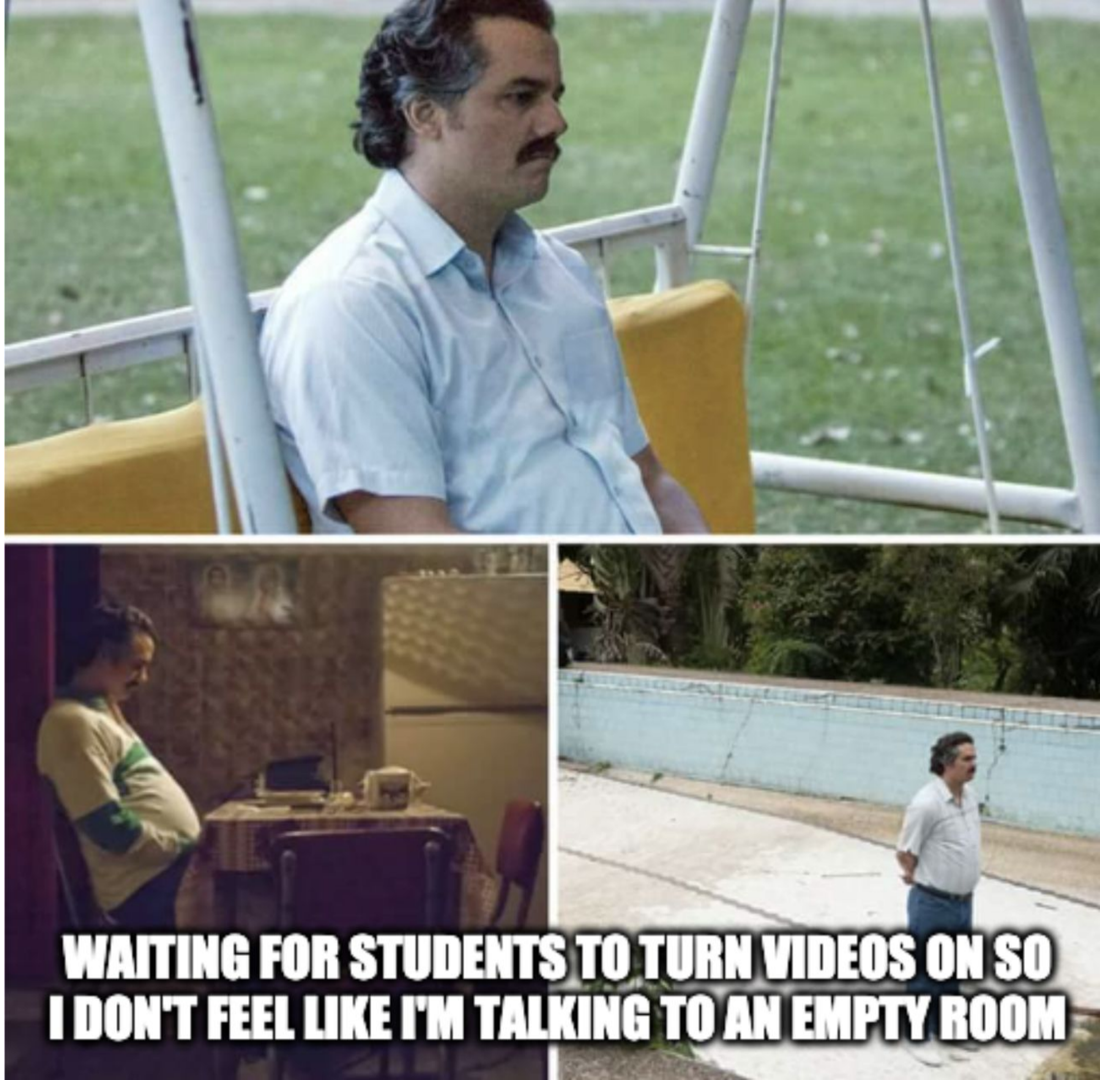
Reply in Zoom chat:

What are your favor movies/shows?
(I need recommendations)

Zoom etiquettes

We appreciate it
if you keep videos on!

- More visual feedback for us
  to adjust materials
- Better learning environment
- Better sense of who you're with
  in class!



Logistics

- OHs started this week
  - Megan: Mon 2 - 2:30pm PST
  - Chloe: Tue 8.30 - 9am PST
  - Chip: Wed 6 - 6:30pm PST
  - Kinbert: Tue 3 - 3:30pm PST
- Final project instruction out
  - Must work in a group of 3

Final project goals

- Have a team by Sun, Jan 16.
  - If you don't have a team by this Friday, let us know!
- Week 3: Meet with course staff for brainstorming
  - Sign-up sheet out this Sun
- Team search is happening

Poll: Do you have a team yet?

1. Yes
2. Yes, but only 2 members and we're looking for one more!
3. No

Agenda

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

Lecture note is on course website / syllabus

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

Terabytes of data scraped from Reddit in a week

1. Who would win?

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

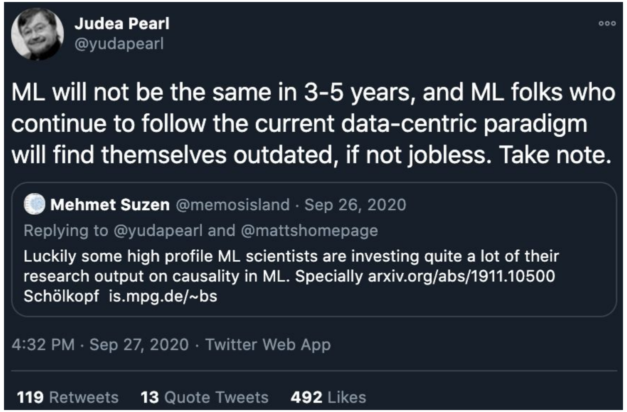
Terabytes of data scraped from Reddit in a week. 85%

From last year

Mind

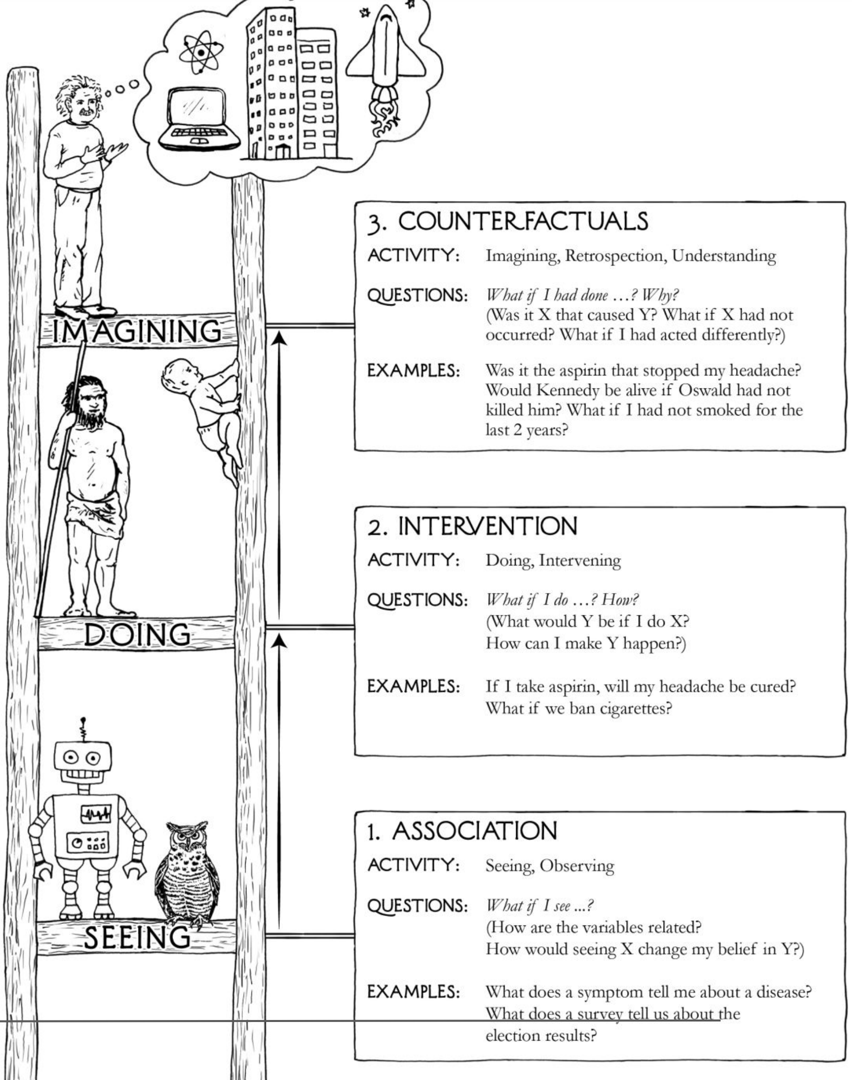
“Data is profoundly dumb.”

Judea Pearl, Mind over data - The Book of Why



Limitations of data

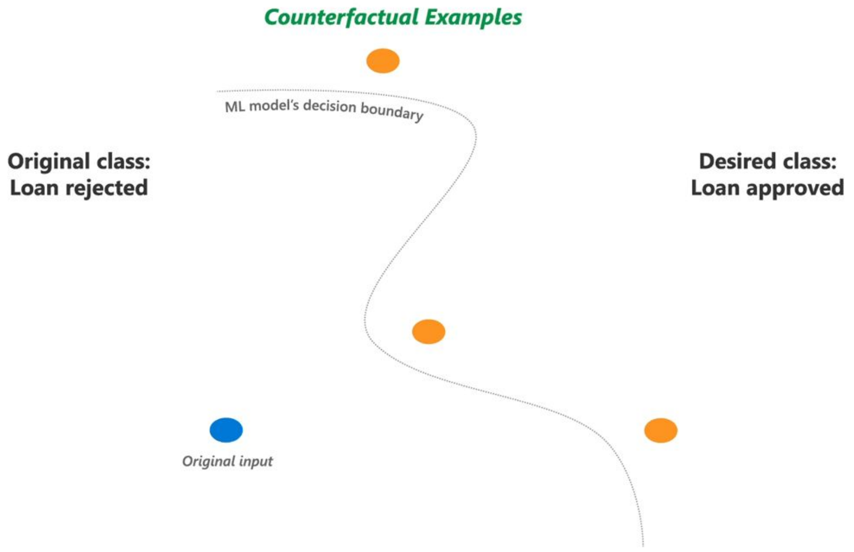
- Observed X, output Y
- Can learn P(X, Y)
- What would happen if didn't do X?



The Book of Why (Pearl and MacKenzie, 2018)

Counterfactuals: example

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

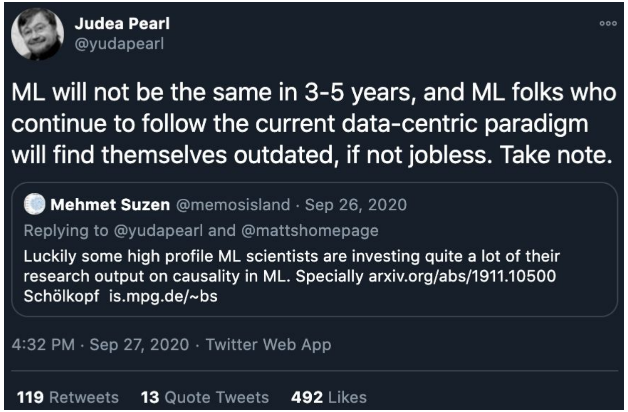


[Open-source library provides explanation for machine learning through diverse counterfactuals](#) (Amit Sharma, Microsoft 2020)

Mind

“Data is profoundly dumb.”

Judea Pearl, Mind over data - The Book of Why



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

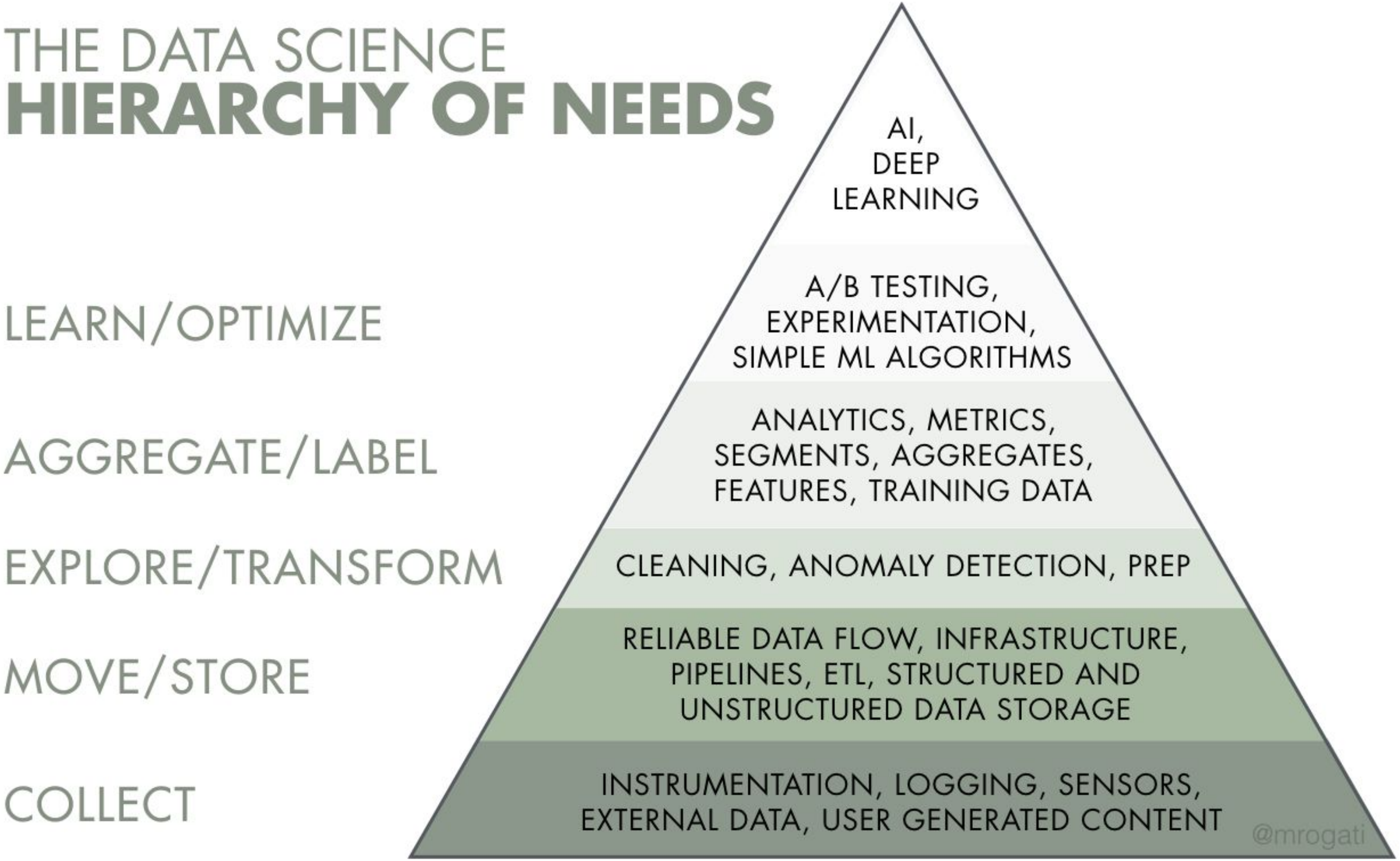
“Imposing structure requires us to make certain assumptions, which are invariably wrong for at least some portion of the data.”

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.

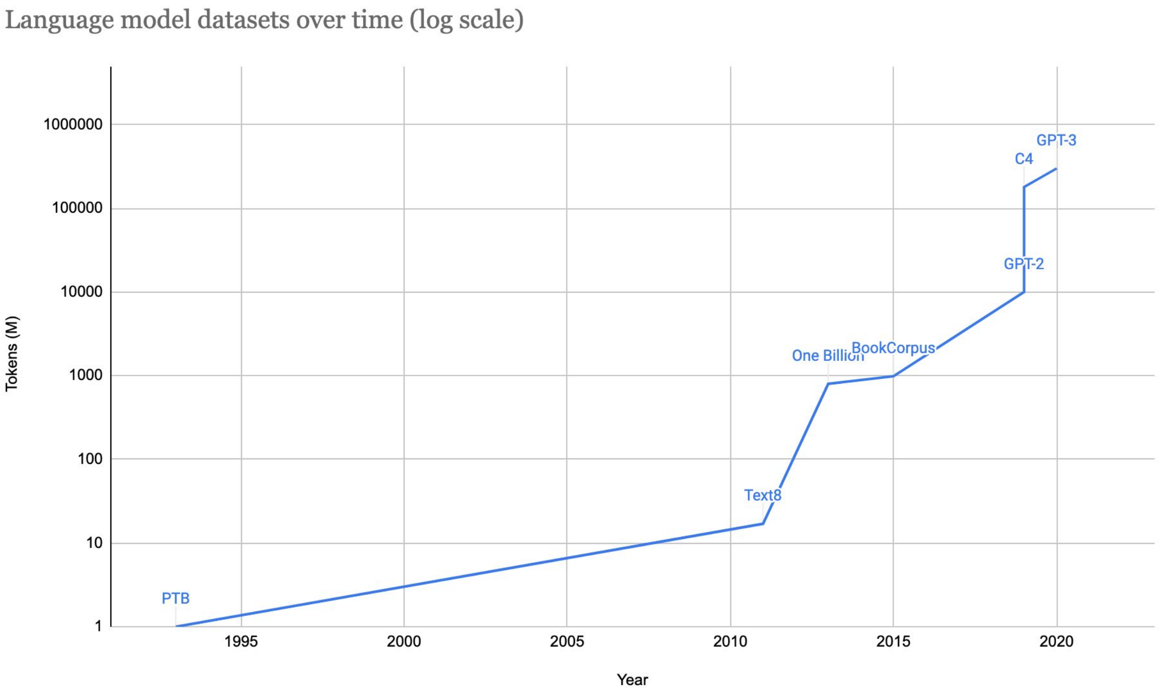
Massive data $\neq$ infinite data



[The AI Hierarchy of Needs](#) (Monica Rogati, 2017)

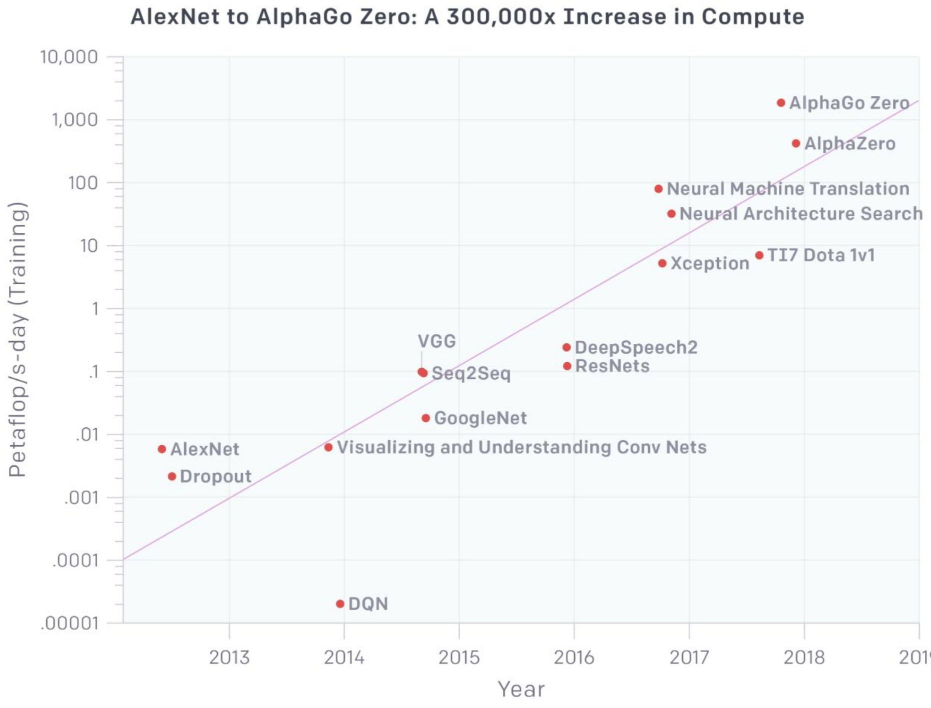
Datasets for language models

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



More data (generally) needs more compute

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



AI and Compute (OpenAI 2018)

⚠️ 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?"

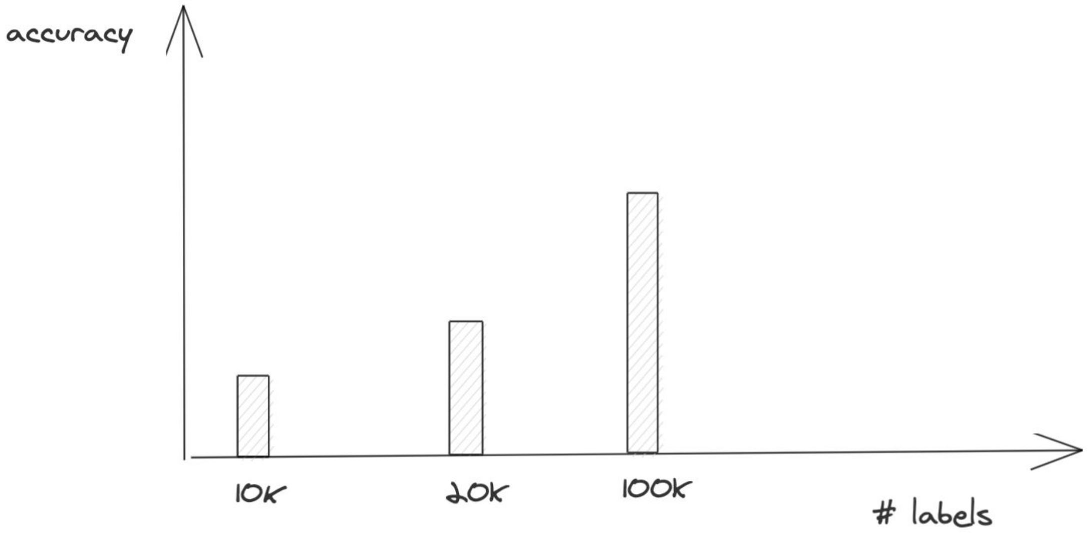
Andrej Karpathy, Director of AI @ Tesla

[CS 329S guest lecture, 2021]

Labeling

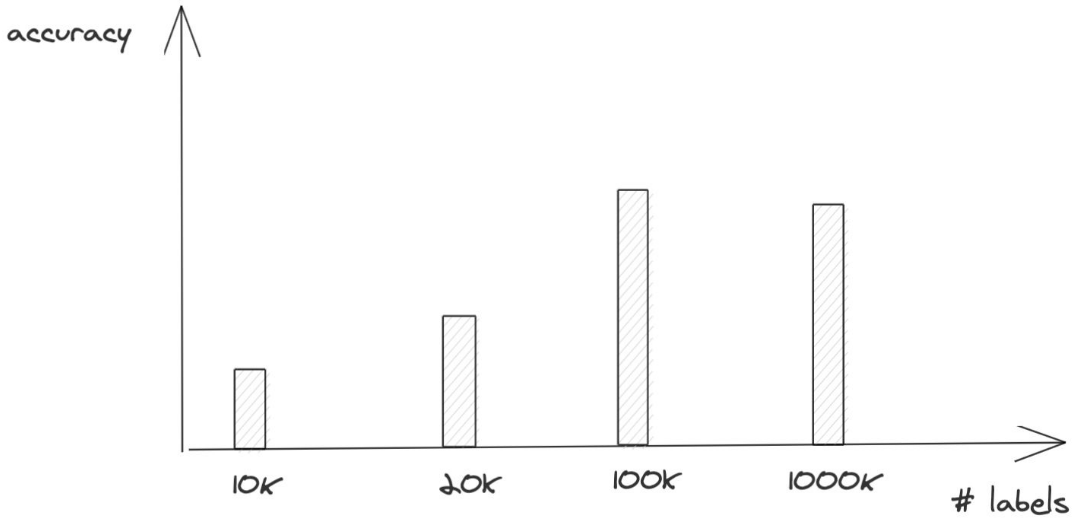
1. Hand-labeling
2. Programmatic labeling
3. Weak supervision, semi supervision, active learning, transfer learning

⚠️ More data isn't always better ⚠️



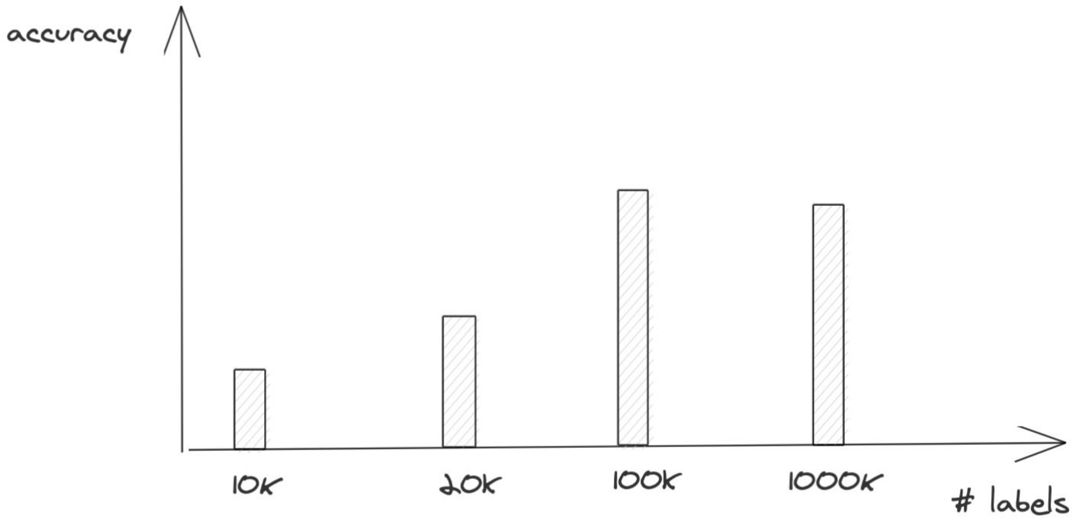
🧠 Idea 🧠: crowdsource data to get 1 million labels!

⚠️ 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

Zoom poll: 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)

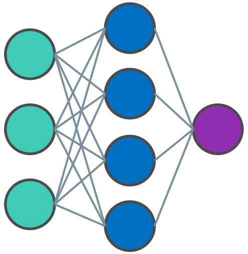
Programmatic labeling

Training data is the bottleneck

Data

+

Algorithms



=

ML Model

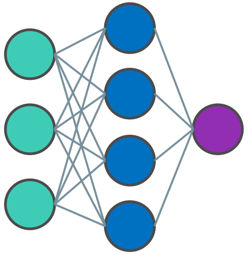
Snorkel

Training data is the bottleneck

Data

Key differentiator

Algorithms



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

Increasingly commoditized

ML Model

"We don't have better algorithms. We just have more data."

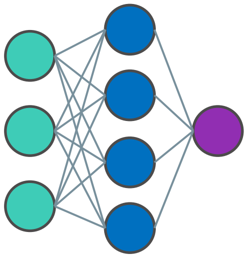
Peter Norvig, [The Unreasonable Effectiveness of Data]

Training data is the bottleneck

Data

- 8 Person-months
- 8-9 pt. differences

Algorithms



- 1-2 days
- <1 pt. differences

ML Model

Stanford MEDICINE

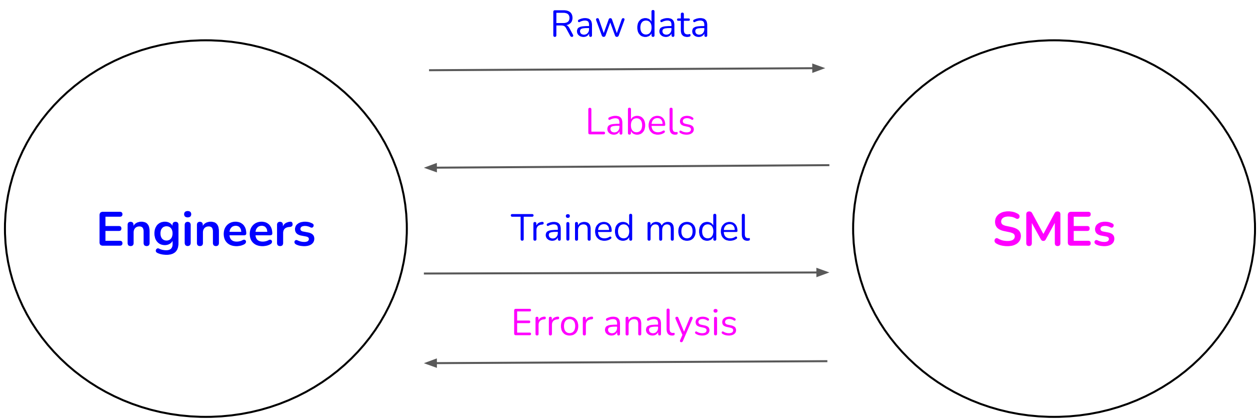
How to get training data in days?

[Cross-Modal Data Programming Enables Rapid Medical Machine Learning]() (Dunnmon et al., 2019)

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



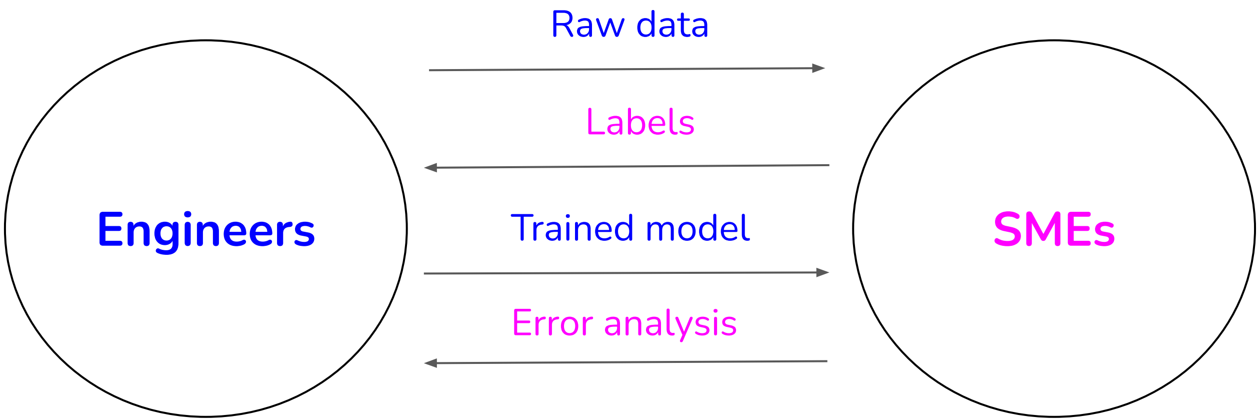

```
def function:
    if X:
       do Y
```

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

Code: version control, reuse, share

How to version, share, reuse **expertise**?

SME as labeling functions




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

```
def labeling_function(note):
    if "pneumonia" in note:
        return "EMERGENT"
```

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

LFs: can express many different types of heuristics

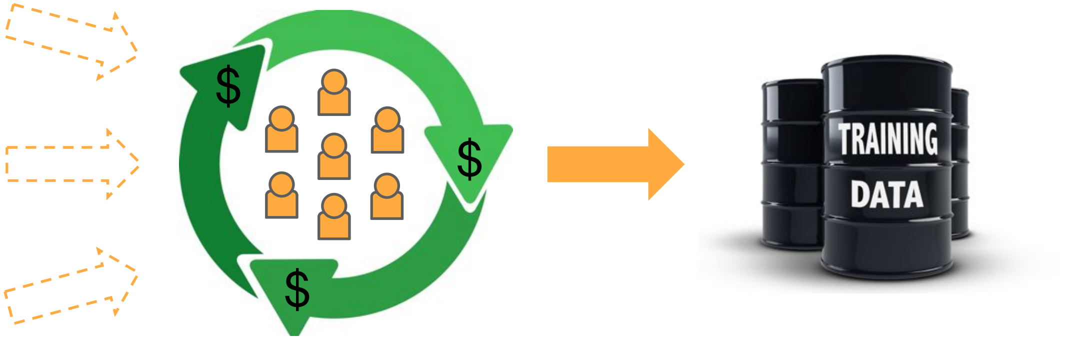
(.*)	Pattern Matching	If a phrase like "send money" is in email
	Boolean Search	If unknown_sender AND foreign_source
	DB Lookup	If sender is in our Blacklist.db
	Heuristics	If SpellChecker finds 3+ spelling errors
	Legacy System	If LegacySystem votes spam
	Third Party Model	If BERT labels an entity "diet"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

??

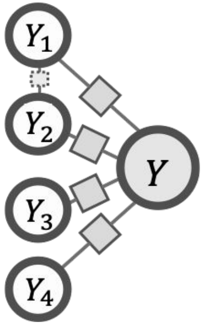
- **Noisy**: Unknown, inaccurate
- **Overlapping**: LFs may be correlated
- **Conflicting**: different LFs give different labels
- **Narrow**: Don't generalize well

LF labels are combined to generate ground truths

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

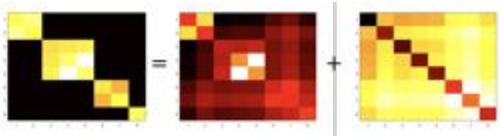
```
def LF_from_grandma(x):
   if x.sender.name is “Grandma”:
      return “HAM”
```

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



[Intuition]

Look at agreements & disagreements



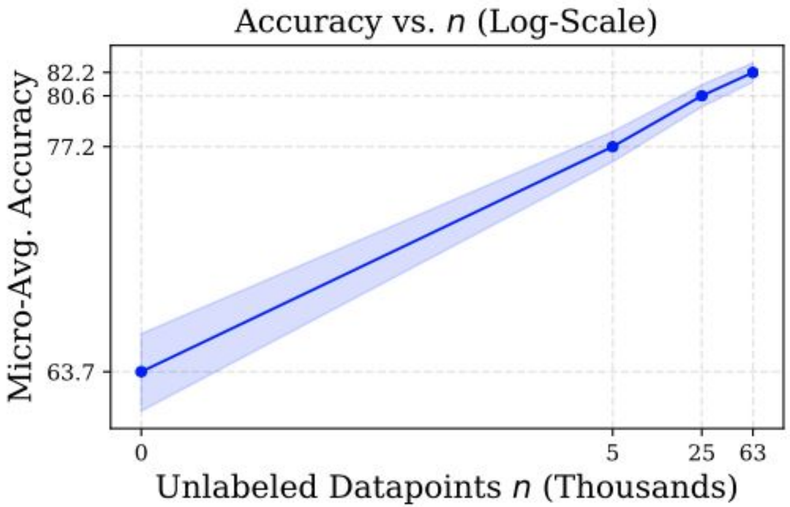
$$(\Sigma^{-1})_O = \Sigma_O^{-1} + zz^T$$

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’19l;
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



Ratner et. al. NeurIPS'16; Ratner et. al. VLDB'18; Ratner et. al. AAAI'19

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

How to get more labeled training data?

Traditional Supervision: Have subject matter experts (SMEs) hand-label more training data

Too expensive!

Active Learning: Estimate which points are most valuable to solicit labels for

Semi-supervised Learning: Use structural assumptions to automatically leverage unlabeled data

***Weak* Supervision:** Get lower-quality labels more efficiently and/or at a higher abstraction level

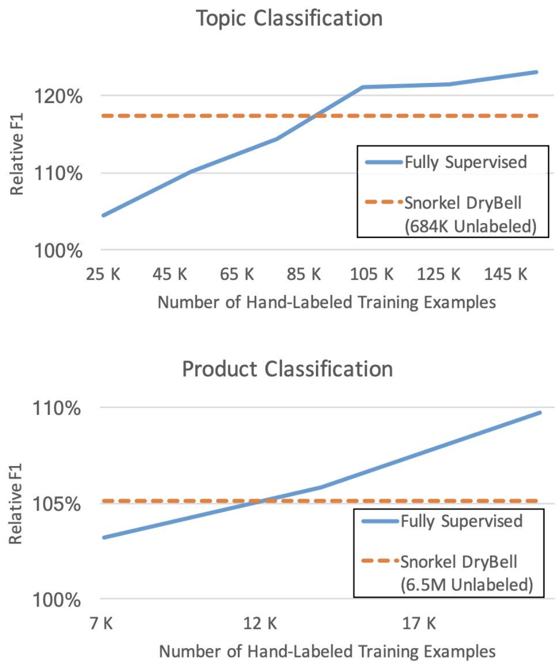
- *Get cheaper, lower-quality labels from non-experts*
- ***Get higher-level supervision over unlabeled data from SMEs***
  - *Heuristics*
  - *Distant Supervision*
  - *Constraints*
  - *Expected distributions*
  - *Invariances*
- *Use one or more (noisy / biased) pre-trained models to provide supervision*

Transfer Learning: Use models already trained on a different task

[Weak Supervision: A New Programming Paradigm for Machine Learning](#) (Ratner et al., 2019)

Weak supervision

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



Harnessing Organizational Knowledge for Machine Learning (Ratner et al., Google AI Blog 2019)

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

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

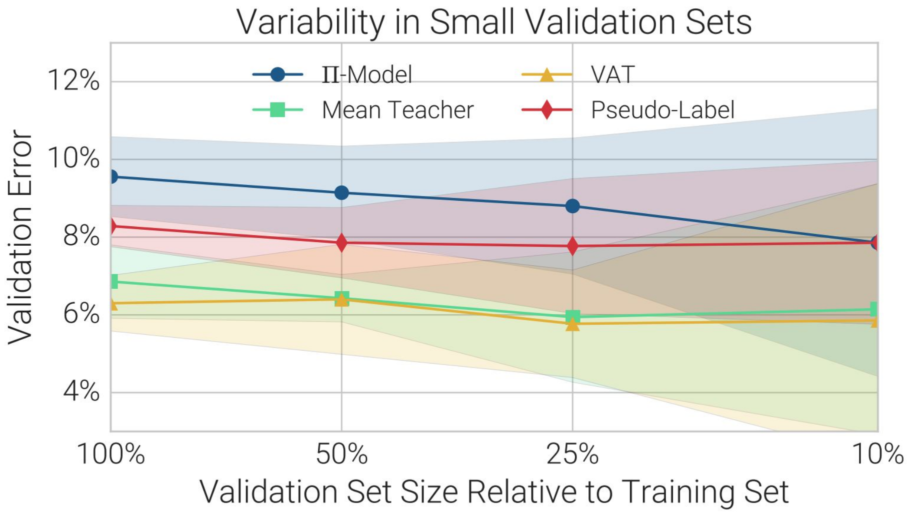
Assumption: small perturbation wouldn’t change a sample’s label

- 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



[Realistic Evaluation of Deep Semi-Supervised Learning Algorithms](#) (Oliver et al., 2018)

Transfer learning

Apply model trained for one task to another task

1. Fine-tuning
2. Prompt-based

Language Models are Few-Shot Learners (OpenAI 2020)

Transfer learning: fine-tuning

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

Language Models are Few-Shot Learners (OpenAI 2020)

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.

```
1  Translate English to French:      ← task description
2  cheese => ...................     ← prompt
```

One-shot

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

```
1  Translate English to French:      ← task description
2  sea otter => loutre de mer        ← example
3  cheese => ...................     ← prompt
```

Few-shot

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

```
1  Translate English to French:      ← task description
2  sea otter => loutre de mer        ← examples
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese => ...................     ← prompt
```

Language Models are Few-Shot Learners (OpenAI 2020)

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 Literature Survey (Burr Settles, 2010)

Active learning metrics

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

Active Learning Literature Survey (Burr Settles, 2010)

Active learning metrics

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

Active Learning Literature Survey (Burr Settles, 2010)

Active learning

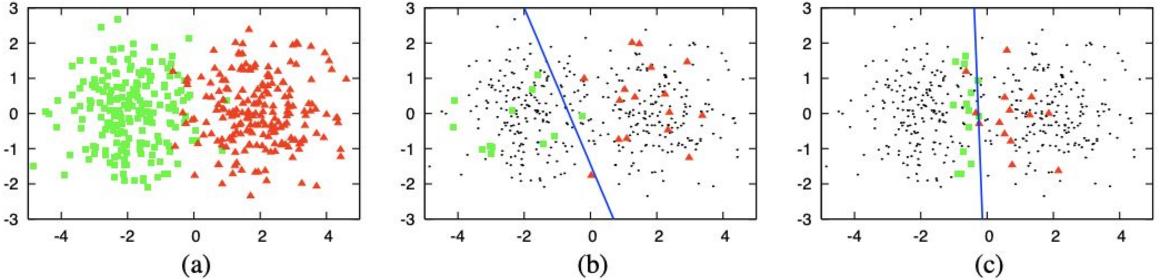


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem 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%).

Active Learning Literature Survey (Burr Settles, 2010)

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 scape 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
  - ...

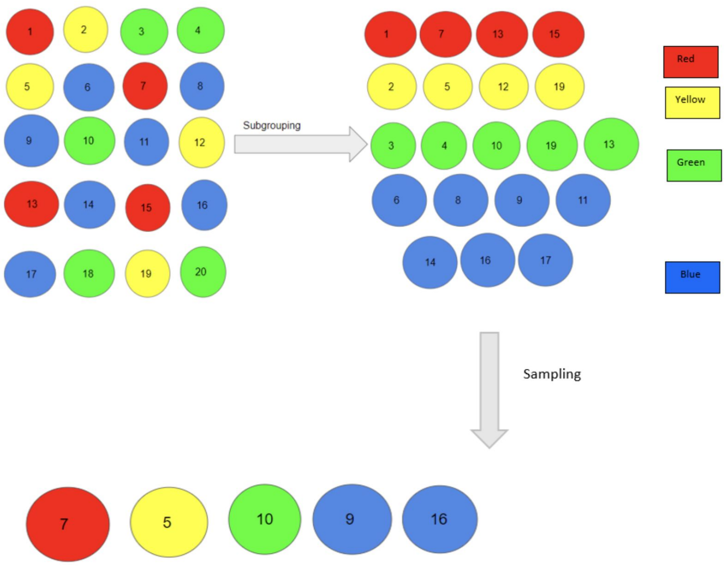
Simple random sampling

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

Pros	Cons
• Simple (easiest type of random sampling)	• 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	Can't be used when: • samples can't be put into subgroups • samples can belong in multiple subgroups (multilabel)

Image from https://www.analyticsvidhya.com/blog/2019/09/data-scientists-guide-8-types-of-sampling-techniques/

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 ~ Q(x) ← easier to sample from
  - Weight by P(x)/Q(x)
- ⚠️ Calculating P(x)/Q(x) might be expensive ⚠️

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

Importance sampling

- Useful when sampling from P(x) is expensive, slow, or infeasible
  - Sample x ~ 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

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

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

use old policy to sample data

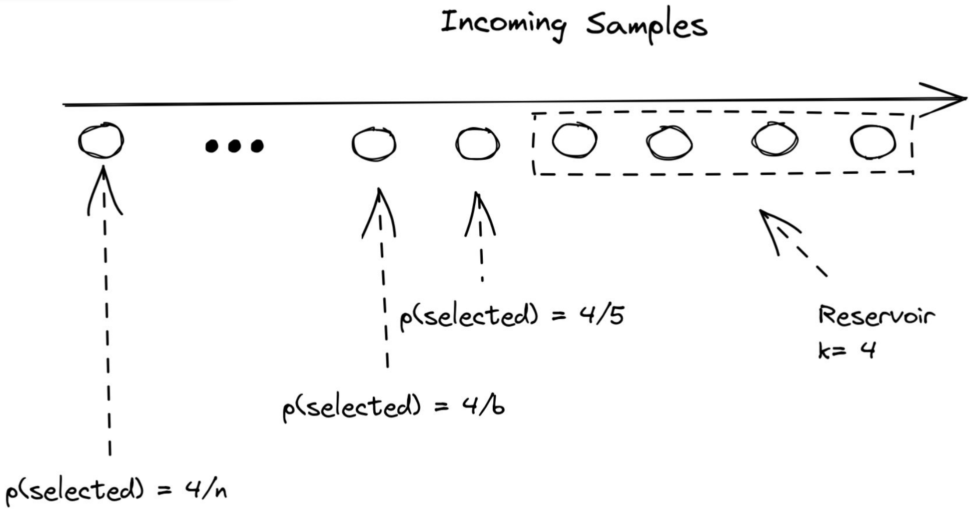
old policy

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 $1 \le j \le k$: 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

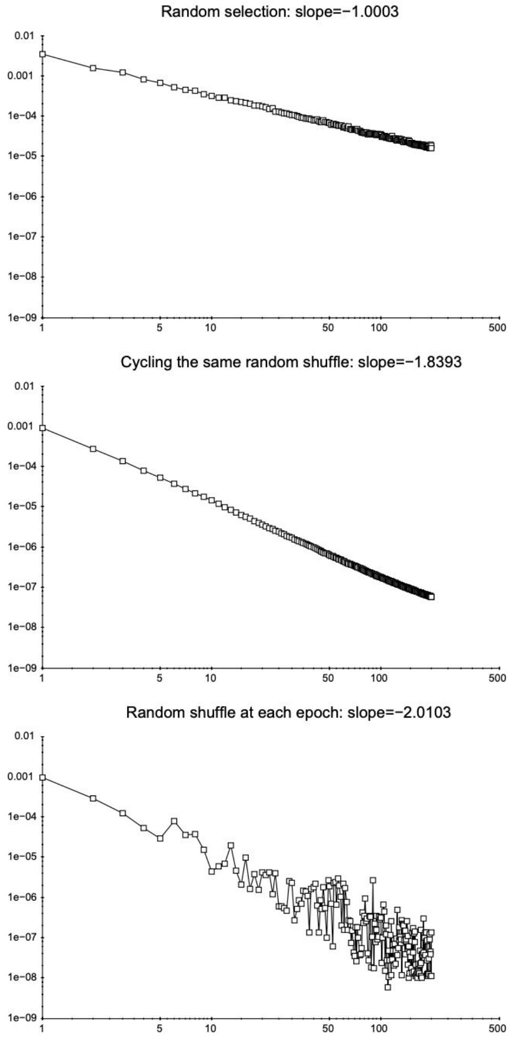
With replacement	Without replacement
Same item can be chosen more than once	Same item can’t be chosen more than once
• No covariance between two chosen samples • Approximate true population distribution	• Covariance between two chosen samples • Covariance reduced as dataset size becomes large
Bagging	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

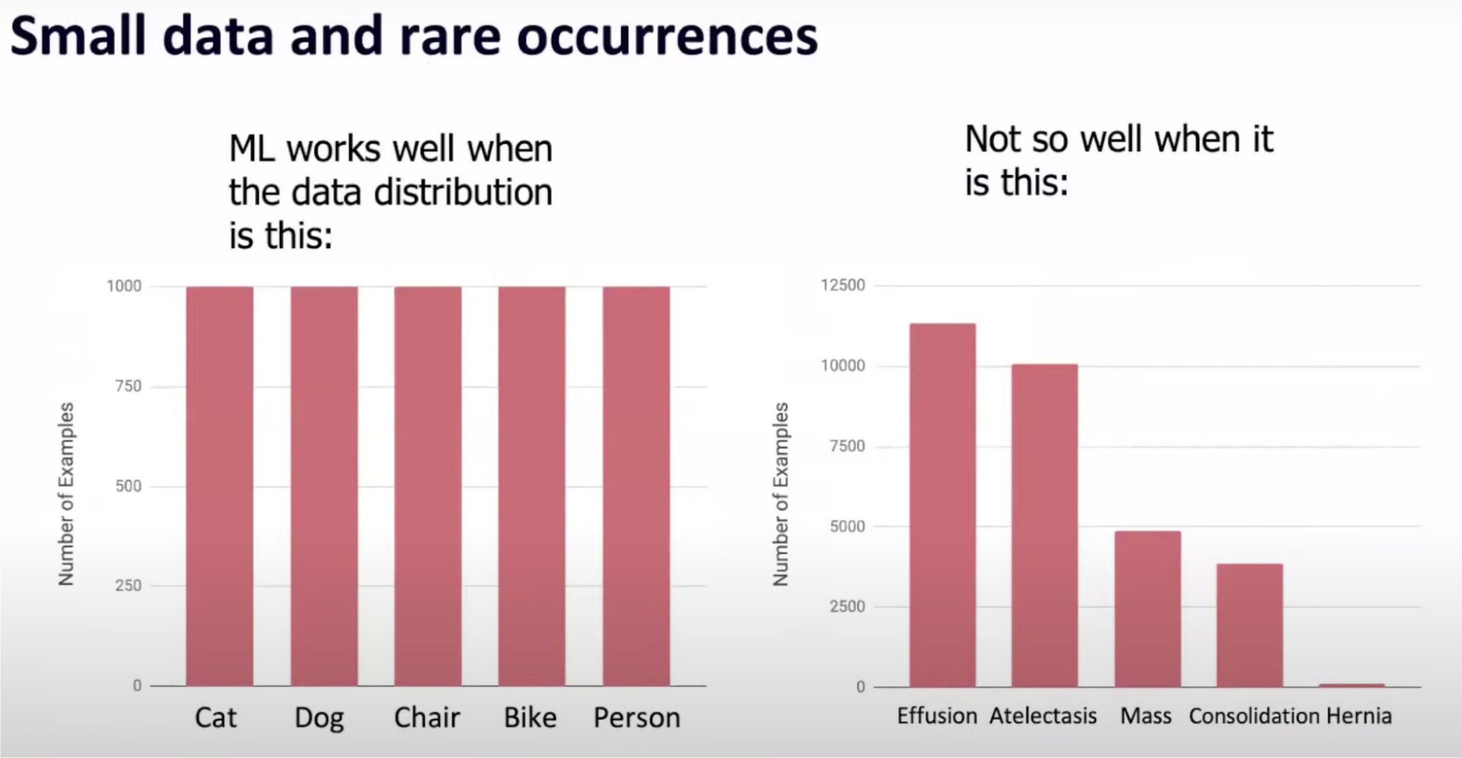
proven for strongly convex loss functions



[Curiously Fast Convergence of some Stochastic Gradient Descent Algorithms]() (Leon Bottou, 2009)
[Why Random Reshuffling Beats Stochastic Gradient Descent]() (Gurbuzbalaba et al., 2015)

5. Class imbalance

Class imbalance



Andrew Ng: Bridging AI's Proof-of-Concept to Production Gap (2020)

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

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: $250

Thanks Eugene Yan for this example!

Asymmetric cost of errors: regression

100% error difference

Not OK

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

OK

Thanks Eugene Yan for this example!

Class imbalance is the norm

- Fraud detection
- Spam detection
- Disease screening
- Churn prediction
- Resume screening
  - E.g. 2% of resumes pass screening
- Object detection
  - Most bounding boxes don't contain any object

People are more interested in unusual/potentially catastrophic events

Image from PyImageSearch

Sources of class imbalance

- Sampling biases
  - Narrow geographical areas (self-driving cars)
  - Selection biases
- Domain specific
  - Costly, slow, or infeasible to collect data of certain classes
- Labeling errors

How to deal with class imbalance

1. Choose the right metrics
2. Data-level methods
3. Algorithm-level methods

1. Choose the right metrics

Model A vs. Model B confusion matrices

Zoom poll:
Which model would you choose?

Model A	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	10	10
Predicted NORMAL	90	890

Model B	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	90	90
Predicted NORMAL	10	810

Choose the right metrics

Model A vs. Model B confusion matrices

Model B has a better chance of telling if you have cancer

Model A	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	10	10
Predicted NORMAL	90	890

Model B	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	90	90
Predicted NORMAL	10	810

Both have the same accuracy: 90%

Symmetric metrics vs. asymmetric metrics

Symmetric metrics	Asymmetric metrics
Treat all classes the same	Measures a model's performance w.r.t to a class
Accuracy	F1, recall, precision, ROC

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

$$F_1\text{–score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

- TP: True positives
- TN: True negatives
- FP: False positives
- FN: False negatives

Class imbalance: asymmetric metrics

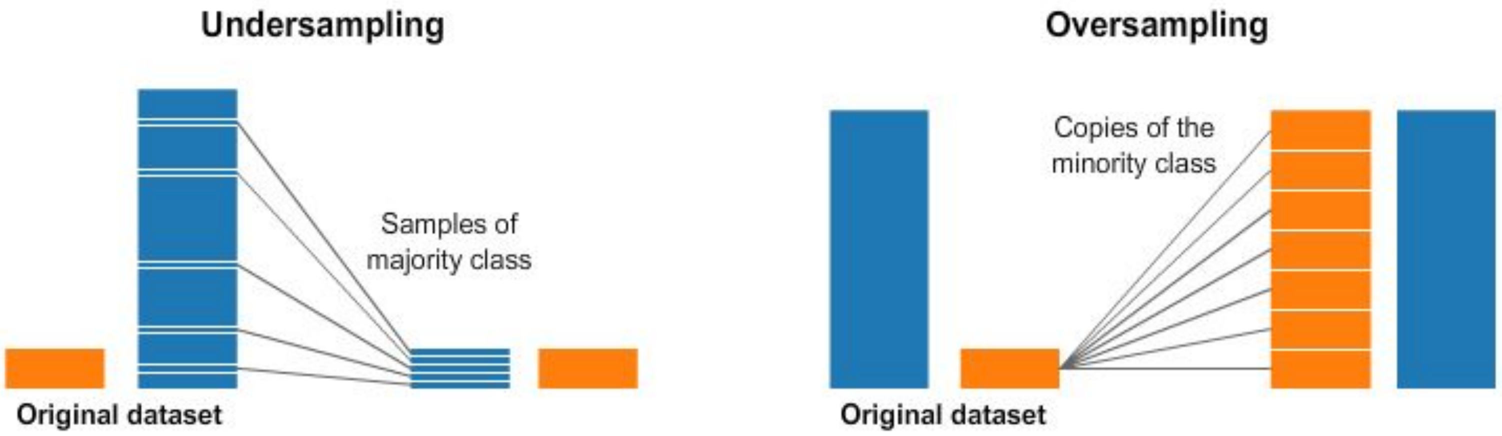
- Your model's performance w.r.t to a class

	CANCER (1)	NORMAL (0)	Accuracy	Precision	Recall	F1
Model A	10/100	890/900	0.9	0.5	0.1	0.17
Model B	90/100	810/900	0.9	0.5	0.9	0.64

⚠️ F1 score for CANCER as 1 is different from F1 score for NORMAL as 1 ⚠️

2. Data-level methods: Resampling

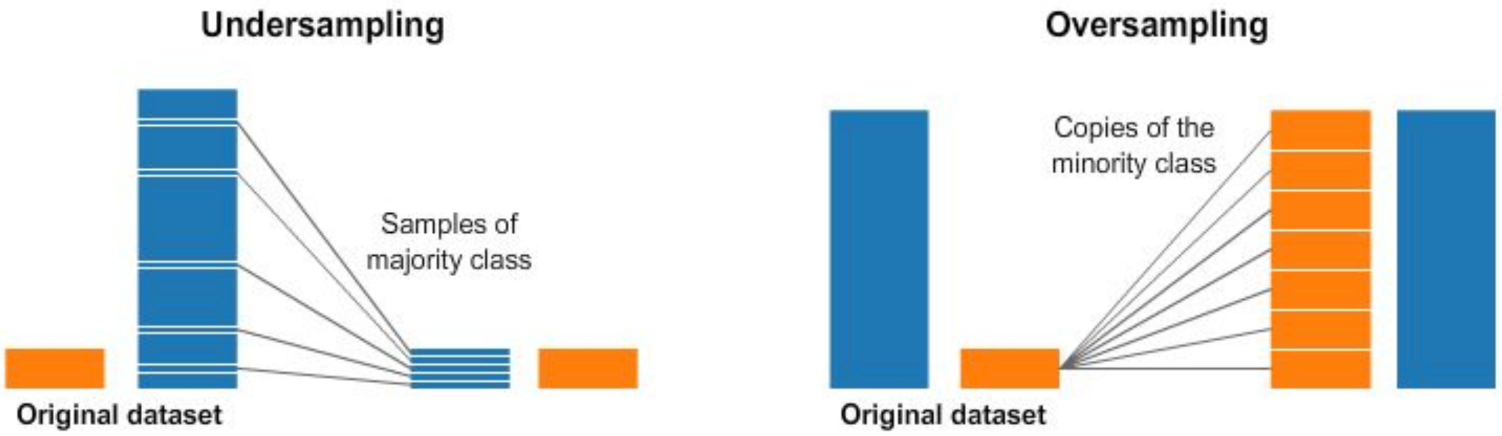
Undersampling	Oversampling
Remove samples from the majority class	Add more examples to the minority class



https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets#t1

2. Data-level methods: Resampling

Undersampling	Oversampling
Remove samples from the majority class	Add more examples to the minority class
Can cause overfitting	Can cause loss of information



https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets#t1

Undersampling: Tomek Links

- Find pairs of close samples of opposite classes
- Remove the sample of majority class in each pair
  - Pros: Make decision boundary more clear
  - Cons: Make model less robust

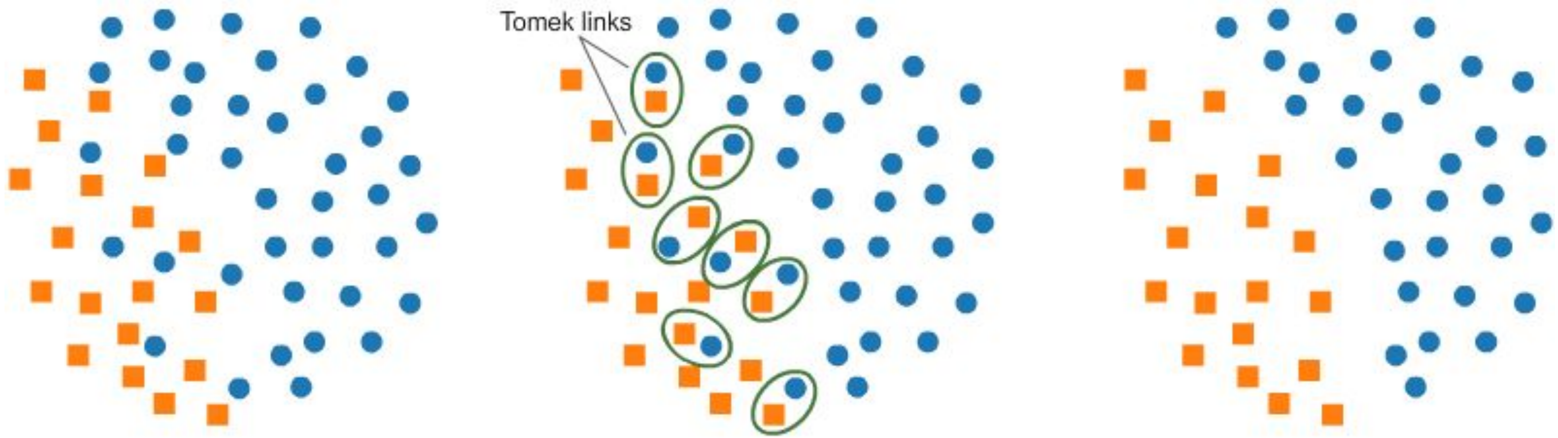


Image from https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets

Oversampling: SMOTE

- Synthesize samples of minority class as convex (~linear) combinations of existing points and their nearest neighbors of same class.

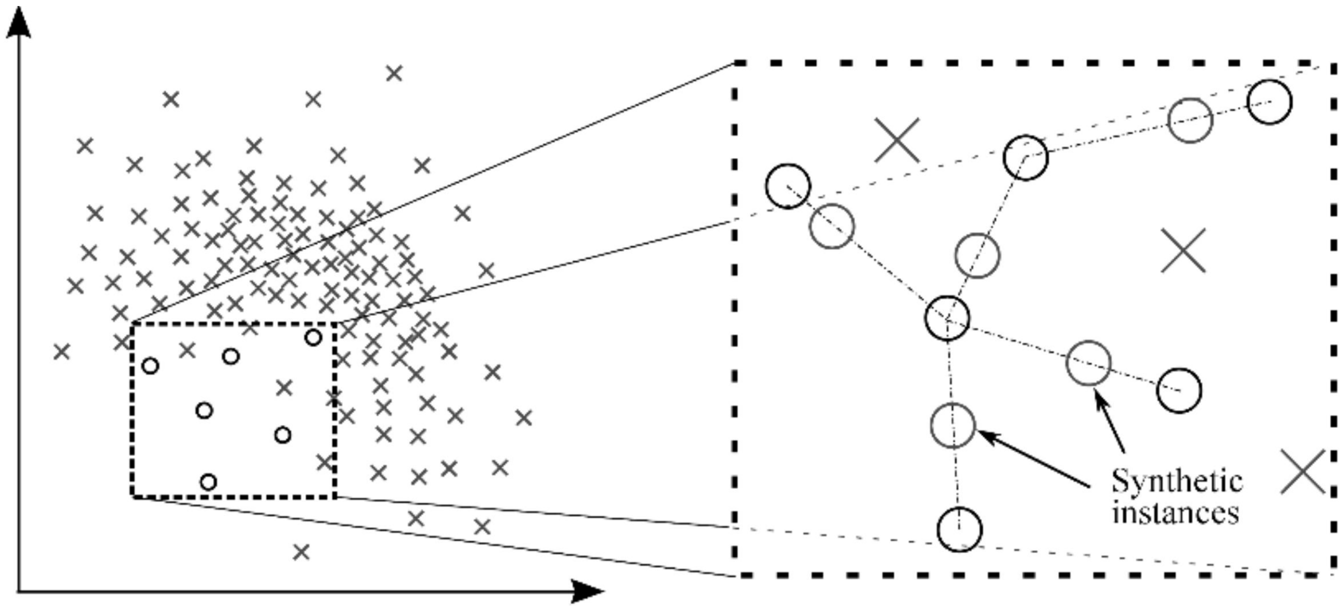


Image from [Analytics Vidhya]

Oversampling: SMOTE

Both SMOTE and Tomek links only work on low-dimensional data!

- Synthesize samples of minority class as convex (~linear) combinations of existing points and their nearest neighbors of same class.

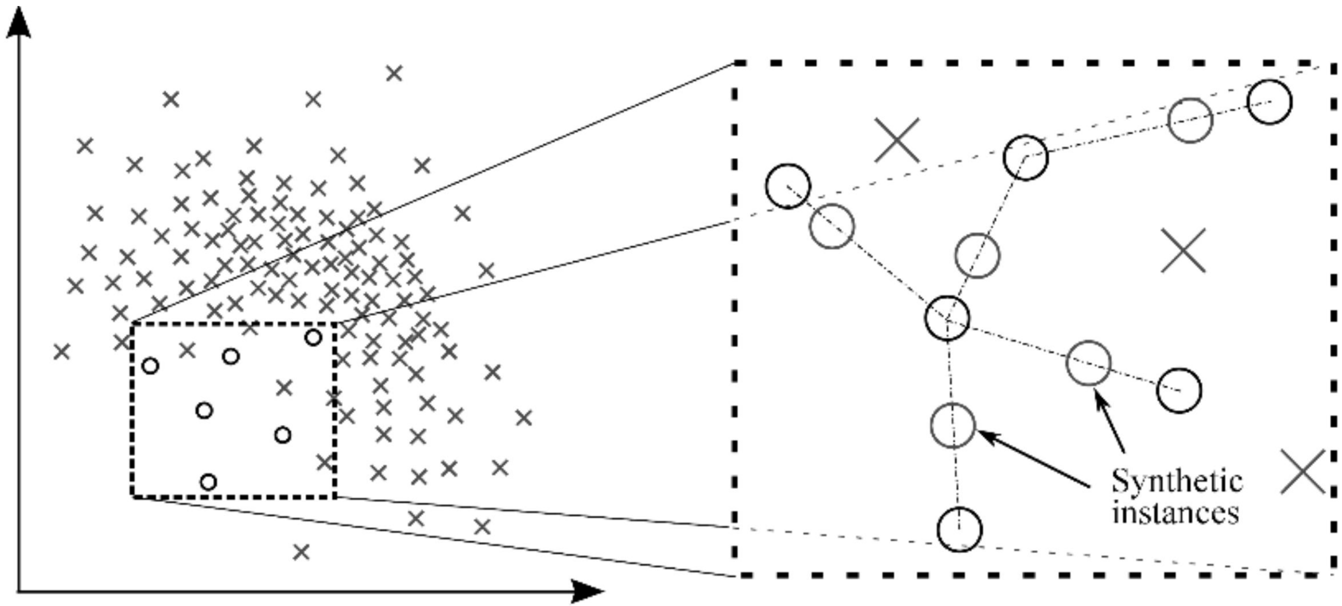


Image from [Analytics Vidhya]()

3. Algorithm-level methods

- Naive loss: all samples contribute equally to the loss
- Idea: training samples we care about should contribute more to the loss

$$L(X;\ \theta) = \sum_{x} L(x\ ;\theta)$$

3. Algorithm-level methods

- Cost-sensitive learning
- Class-balanced loss
- Focal loss

Cost-sensitive learning

- C_{ij}: the cost if class i is classified as class j

	Actual NEGATIVE	Actual POSITIVE
Predicted NEGATIVE	$C(0,\ 0)\ =\ C_{00}$	$C(1,\ 0)\ =\ C_{10}$
Predicted POSITIVE	$C(0,\ 1)\ =\ C_{01}$	$C(1,\ 1)\ =\ C_{11}$

- The loss caused by instance x of class i will become the weighted average of all possible classifications of instance x.

$$L(x\ ;\theta)\ =\ \sum_{j} C_{ij}P(j\mid x;\ \theta)$$

Class-balance loss

- Give more weight to rare classes

Non-weighted loss

$$L(X;\ \theta) = \sum_i L(x_i; \theta)$$

Weighted loss

$$L(X;\ \theta) = \sum_i W_{y_i} L(x_i; \theta)$$

$$W_c = \frac{N}{number\ of\ samples\ of\ class\ C}$$

```
model.fit(features, labels, epochs=10, batch_size=32, class_weight={"fraud": 0.9, "normal":
0.1})
```

Focal loss

- Give more weight to difficult samples:
    - downweighs well-classified samples

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

$$\mathrm{CE}(p_t) = -\log(p_t)$$

$$\mathrm{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t)$$

Focal Loss for Dense Object Detection (Lin et al., 2017)

Machine Learning Systems Design

Next class: Feature Engineering