Machine Learning Systems Design

Data Lifecycle

Lecture 6: Data Preparation

CE 40959 Spring 2023
Ali Zarezade
SharifMLSD.github.io

Agenda

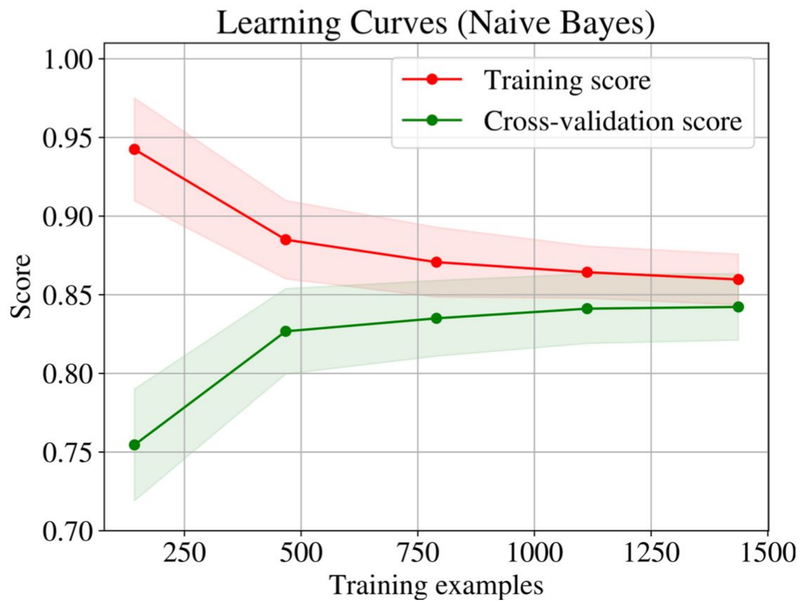
1. Questions About Data

Questions about data

- Is the data accessible?

Questions about data

- Is the data accessible?
- Is there enough data?



Questions about data

- Is the data accessible?
- Is there enough data?

Some rule of thumbs:

- 10 times the amount of features
- 100 or 1000 times the number of classes
- 10 times the number of trainable parameters

Questions about data

- Is the data accessible?
- Is there enough data?
- Is the data useable?

Some causes for unusability of data:

- Expired data or significantly not up to date
- incomplete or unrepresentative of the phenomenon
- There is data leakage
- Data is not tidy[1]

Wickham, Hadley. “Tidy data.” Journal of Statistical Software 59.10 (2014): 1-23

Questions about data

- Is the data accessible?
- Is there enough data?
- Is the data useable?
- Is the data reliable?

Some causes for unreliability of data:

- Measuring device errors
- Majority voted crowd workers label
- Delayed label
- Indirect label

2. Data Quality

Data quality

Data quality has two components:

- Raw data quality
- Labeling quality

Data quality

Some common problems with raw data are

- noise
- bias
- low predictive power
- outdated examples

Noise in data

Noise in data is a corruption of examples:

- Images can be blurry or incomplete.
- Text can lose formatting, which makes some words concatenated or split.
- Audio data can have noise in the background.
- Poll answers can be incomplete or have missing attributes

Noise in data

Is noise in data always bad?

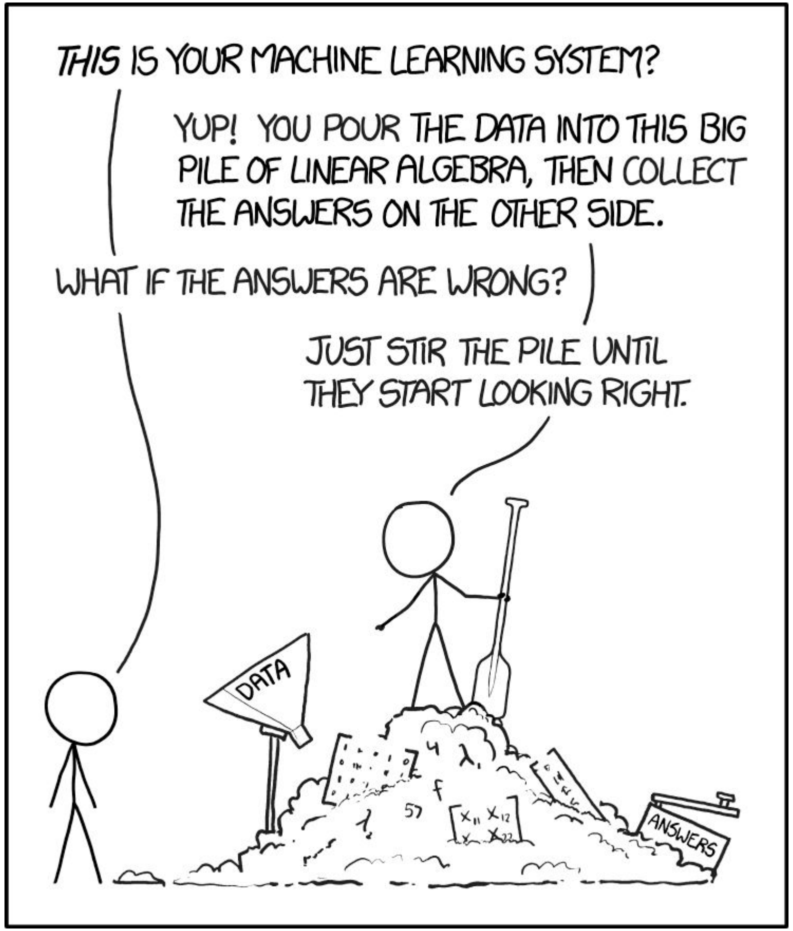
Noise in data

Is noise in data always bad?

- In **small** datasets noise can lead to **overfitting**
- In **large** datasets random noise is typically *averaged out*, and can have **regularization** effect.

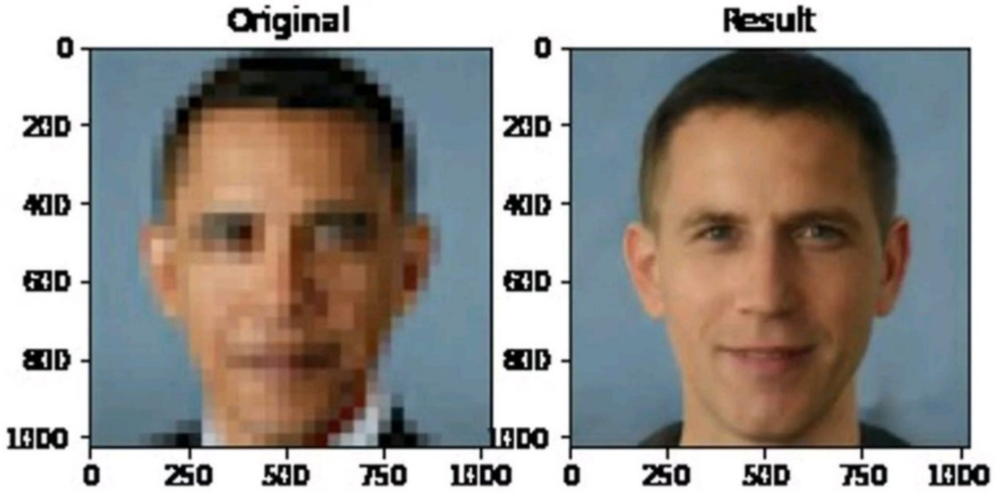
Bias in data

Bias in data is an inconsistency with the phenomenon that data represents.


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Selection bias

Tendency to skew your choice of data sources to those that are easily available, convenient, and/or cost-effective.



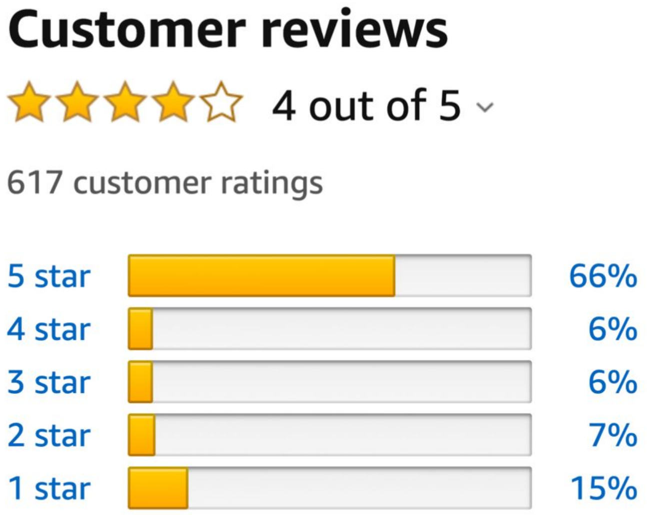
Selection bias

Can be avoided by systematically questioning the reason why a specific data source was chosen.

E.g., training the model on current customers' data only is unwise as they are more brand loyal than random potential customer.

Self-selection bias

Is a form of selection bias where you get the data from sources that “volunteered” to provide it.



Customer reviews

4 out of 5

617 customer ratings

5 star	66%
4 star	6%
3 star	6%
2 star	7%
1 star	15%

Self-selection bias

It is common in surveys. Longer surveys tend to receive lower quality responses due to decreased attention. To minimize bias and increase quality responses, *keep surveys short and provide incentives*.

Sampling bias

When the distribution of examples used for training doesn't reflect the distribution of the inputs the model will receive in production.

You are working on a system that classifies documents. You gather a dataset with equal amount of documents on each topic. You observe 5% error. But, after deployment, you see 30% error. Why did this happen?

Prejudice or stereotype bias

Often observed in data obtained from sources like books or photo archives, or from online activity such as social media, online forums, and comments to online publications.

```
king - man + woman ≈ queen

programmer - man + woman ≈ homemaker
```

Prejudice or stereotype bias

It can be reduced by exposing the learning algorithm to a more even-handed distribution of examples.

E.g., a data analyst could choose to under-sample the number of women indoors, or oversample the number of men at home.

Other biases

- Omitted variable bias
- Experimenter bias
- Labeling bias
- Systematic value distortion
- Sponsorship or funding bias

Low predictive power

Does the model underperform because it is not expressive enough? Does the data not contain enough information from which to learn?

Spotify song/playlist recommendation

Good data properties

- it contains enough information that can be used for modeling,
- it has good coverage of what you want to do with the model,
- it reflects real inputs that the model will see in production,
- it is as unbiased as possible,
- it is not a result of the model itself,
- it has consistent labels, and
- it is big enough to allow generalization.

3. Data Sampling

Types of sampling

- Non-probability sampling
- Random 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

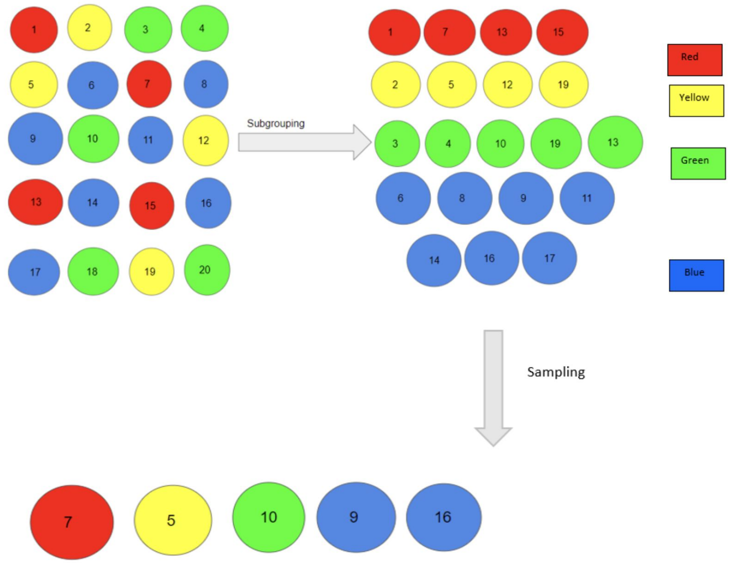
Simple random sampling

- 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) $\longleftarrow$ 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) $\longleftarrow$ 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

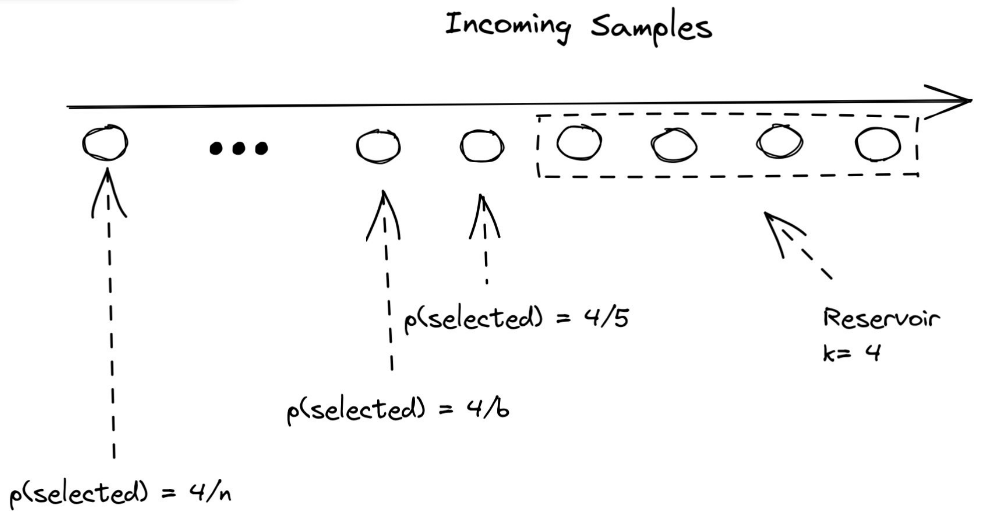
old policy

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 ≤ j ≤ 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

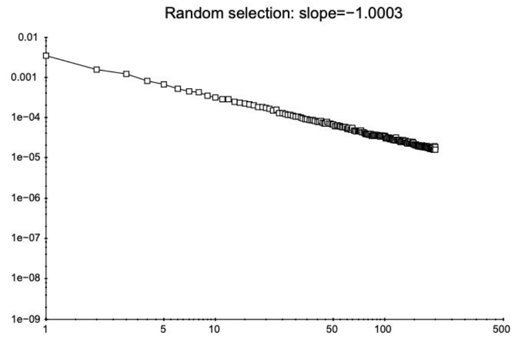
With replacement	Without replacement
Same item can be chosen more than once	Same item can't be chosen more than once
● No covariance between two chosen samples ● Approximate true population distribution	● Covariance between two chosen samples ● Covariance reduced as dataset size becomes large
Bagging	Mini-batch gradient descent

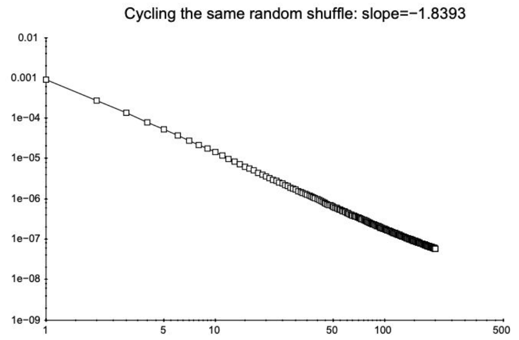
Why do we use epochs instead of just sampling with replacement from the entire dataset?

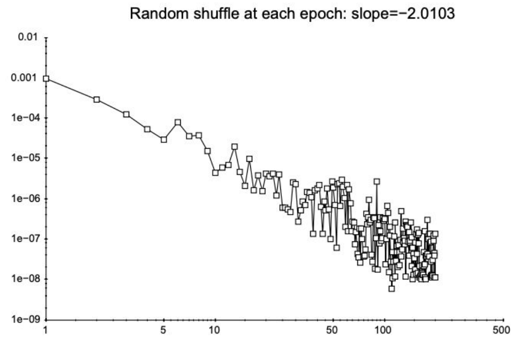
With vs. without replacements

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)

4. Data Labeling

Labeling

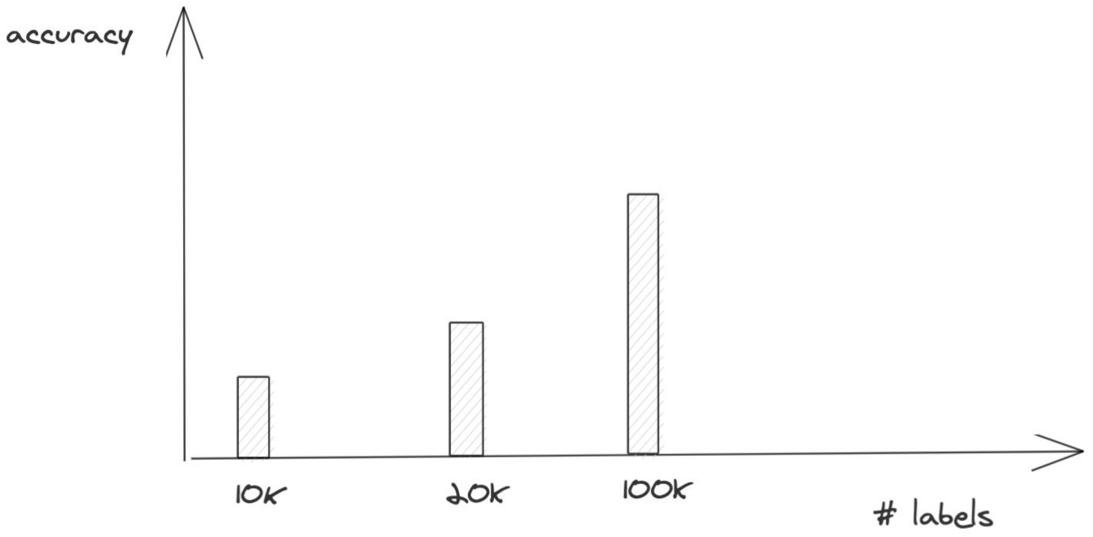
When I told our recruiters that I wanted an in-house labeling team, they asked how long I'd need this team for. I told them: "How long do we need an engineering team for?"

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

Labeling

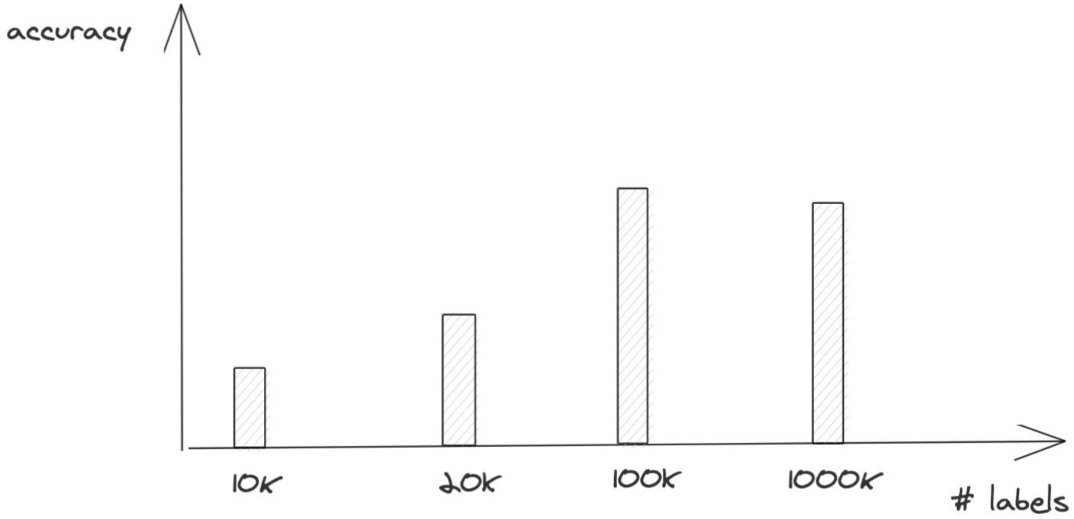
1. Hand-labeling
2. Programmatic labeling
3. Weak supervision, semi supervision, active learning, transfer learning

⚠️ More data isn’t always better ⚠️



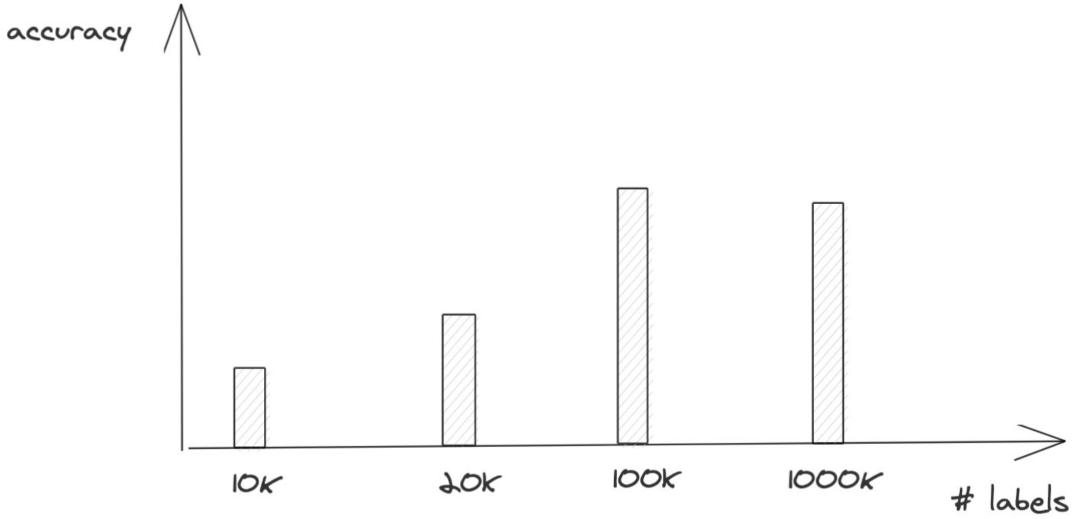
🧠 Idea 🧠: crowdsource data to get 1 million labels!

⚠️ 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/ambiguity: 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 about labeling rule or even 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

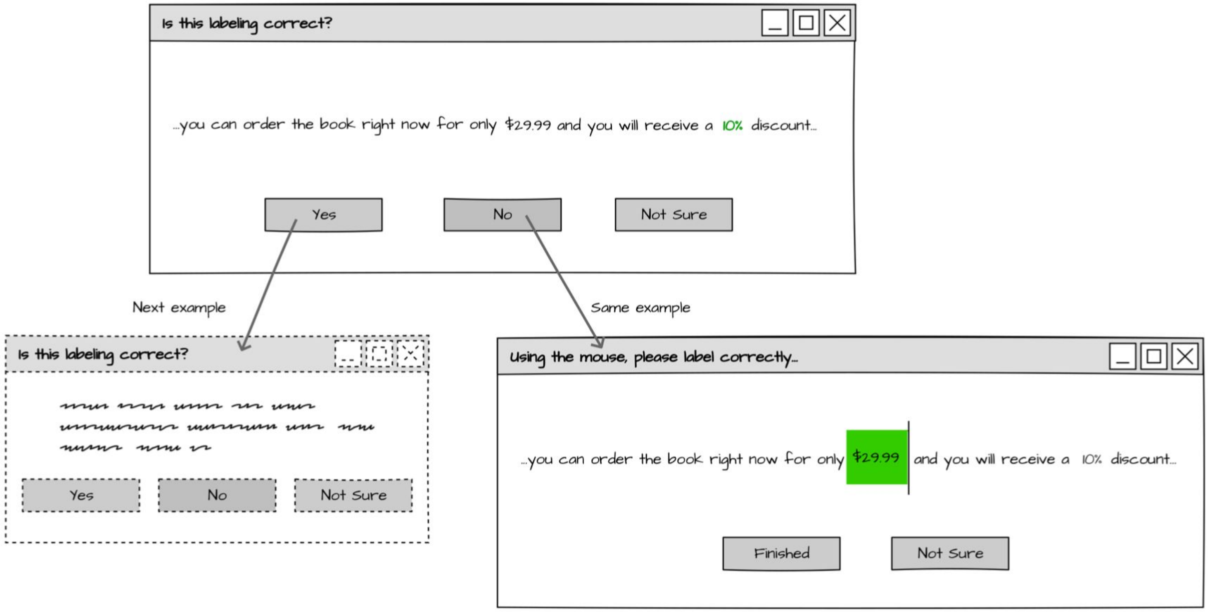
- Clear problem definition
- Annotation training
- Data lineage: track where data/labels come from

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)

Noisy pre-labeling

- Pre-labeling the example using the current best model


Is this labeling correct?
...you can order the book right now for only $29.99 and you will receive a 10% discount...
Yes
No
Not Sure
Next example
Same example
Is this labeling correct?
Yes
No
Not Sure
Using the mouse, please label correctly...
...you can order the book right now for only $29.99 and you will receive a 10% discount...
Finished
Not Sure

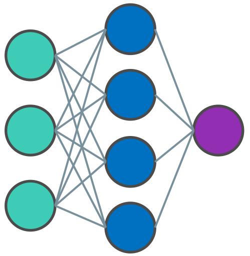
Programmatic labeling

Training data is the bottleneck

Data

+

Algorithms



=

ML Model

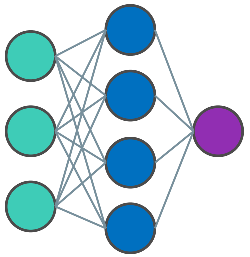
Snorkel

Training data is the bottleneck

Data

+

Algorithms



=

ML Model

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

↑

Key differentiator

$\underbrace{\hphantom{xxxxxxxxxxxx}}$

Increasingly commoditized

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

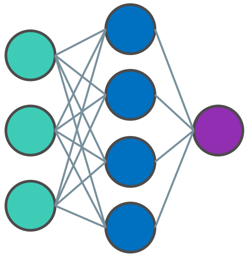
Peter Norvig, [The Unreasonable Effectiveness of Data]

Snorkel

Training data is the bottleneck

Data

Algorithms



ML Model

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

- 1-2 days
- <1 pt. differences

Stanford MEDICINE

How to get training data in days?

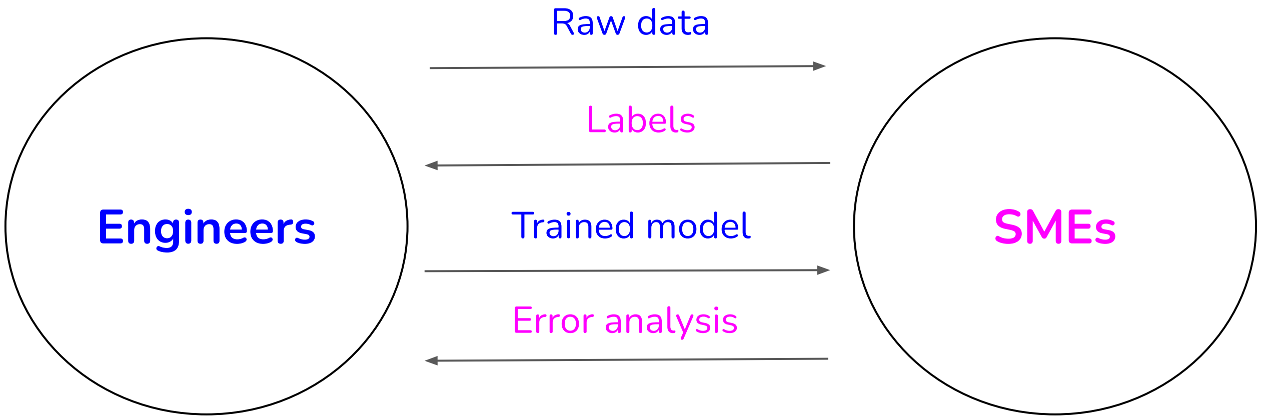
Cross-Modal Data Programming Enables Rapid Medical Machine Learning (Dunnmon et al., 2019)

Snorkel

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



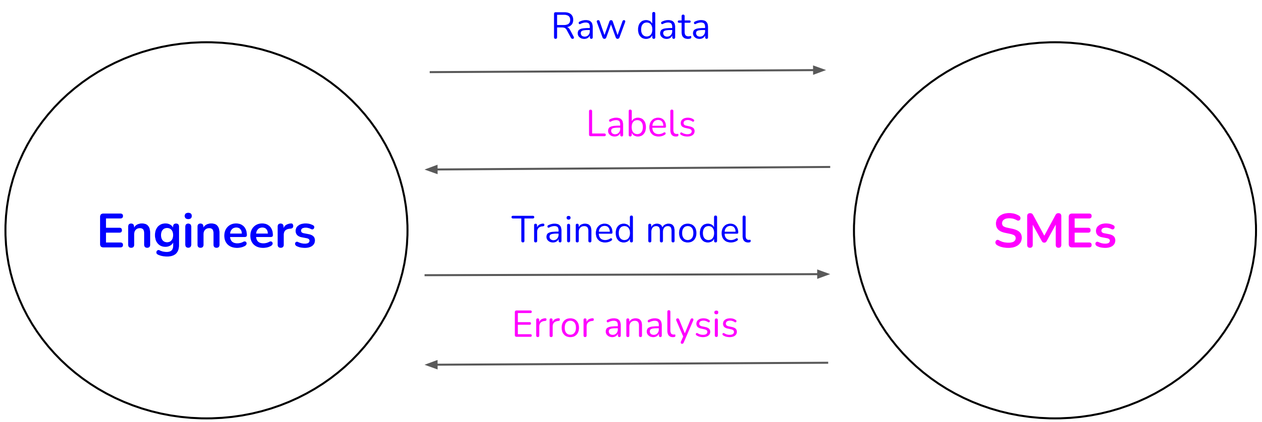
```
def function:
   if X:
      do Y
```

```
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

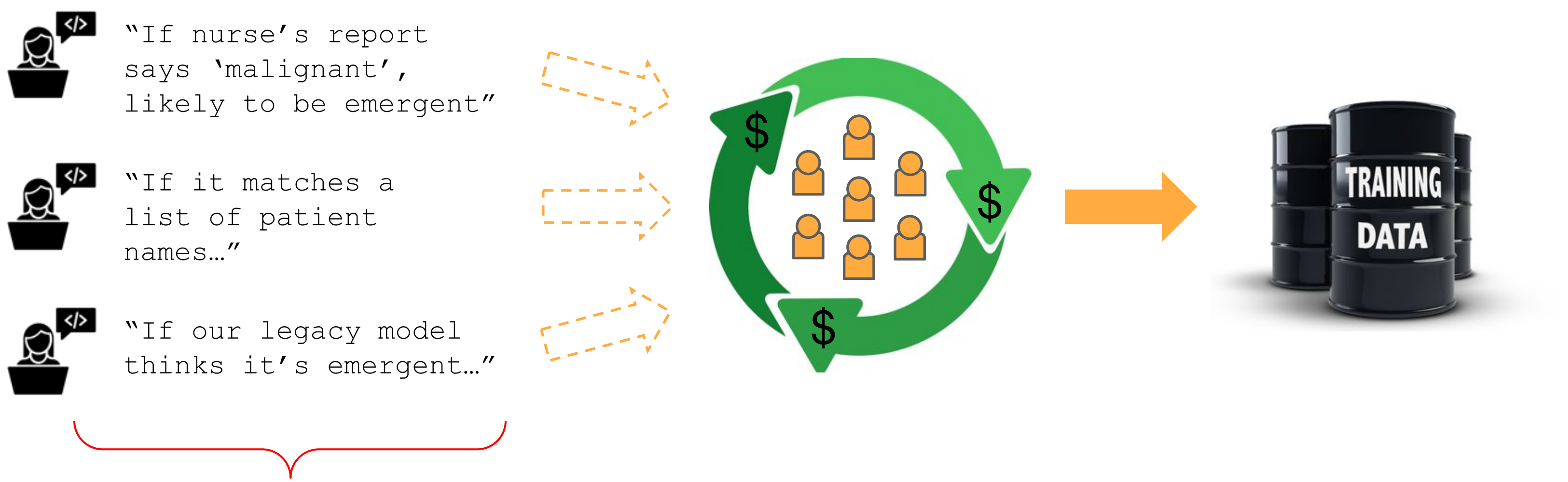
(.*)	Pattern Matching	If a phrase like “send money” is in email
	Boolean Search	If unknown_sender AND foreign_source
	DB Lookup	If sender is in our Blacklist.db
	Heuristics	If SpellChecker finds 3+ spelling errors
	Legacy System	If LegacySystem votes spam
	Third Party Model	If BERT labels an entity “diet”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

??

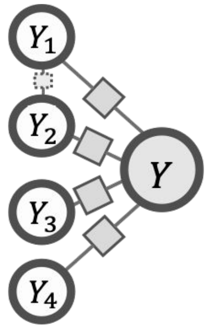
- **Noisy**: Unknown, inaccurate
- **Overlapping**: LFs may be correlated
- **Conflicting**: different LFs give different labels
- **Narrow**: Don’t generalize well

LF labels are combined to generate ground truths

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

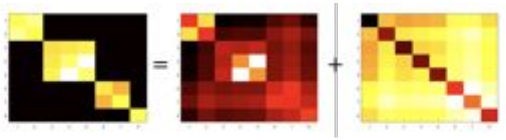
```
def LF_from_grandma(x):
   if x.sender.name is “Grandma”:
      return “HAM”
```

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



[Intuition]

Look at agreements & disagreements



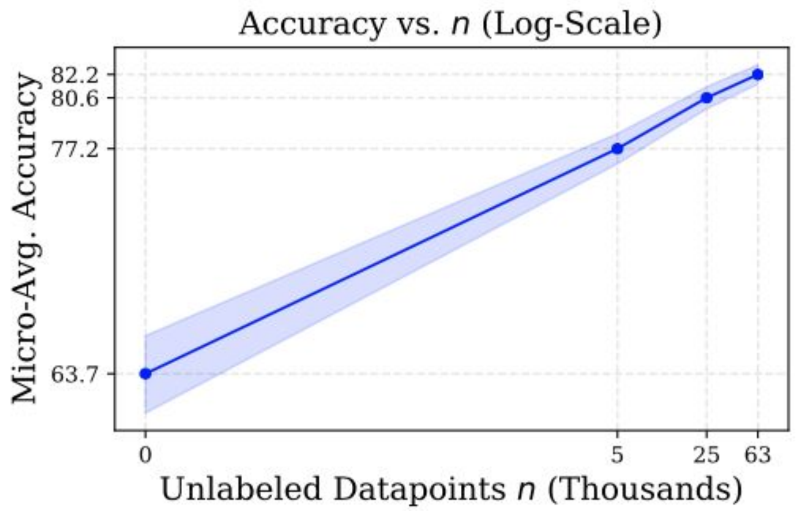
$$(\Sigma^{-1})_O = \Sigma_O^{-1} + zz^T$$

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]

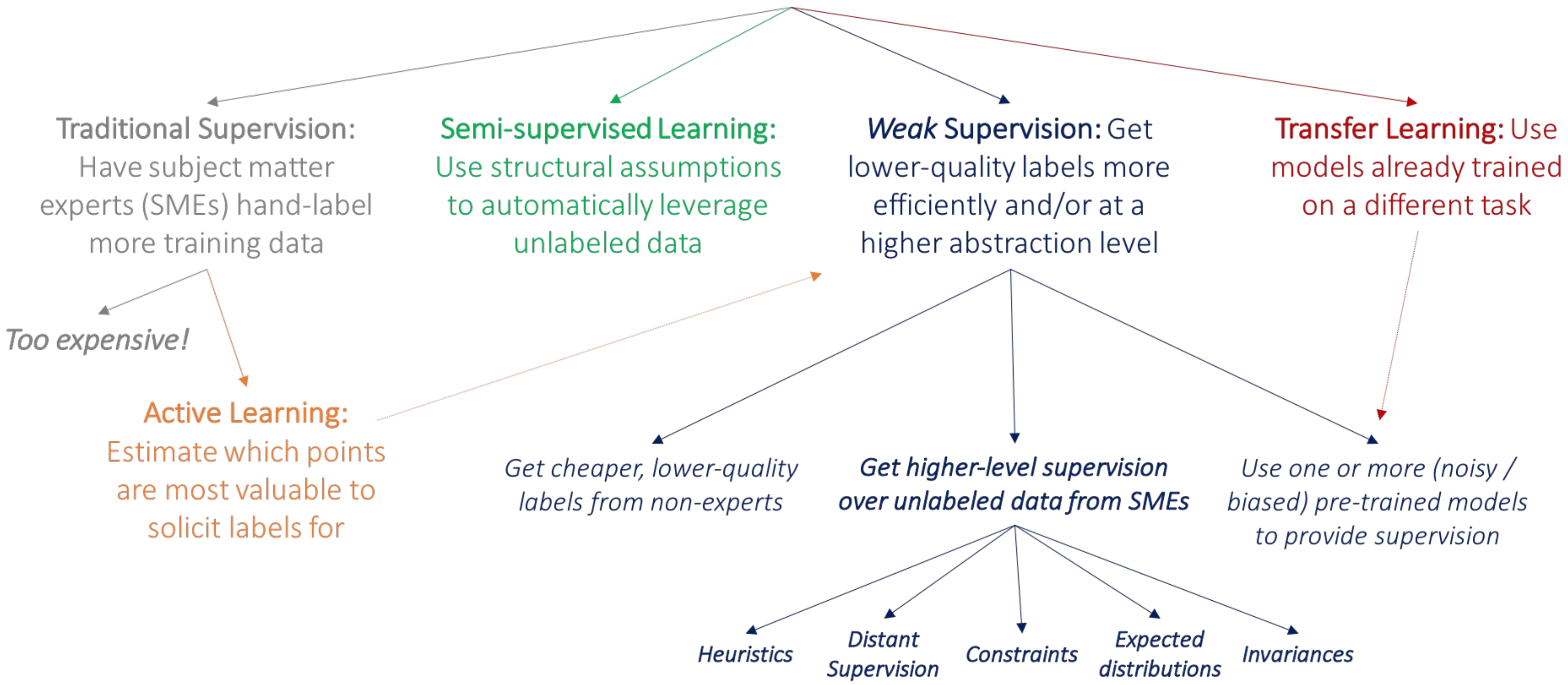
Hand labeling	Programmatic labeling
Expensive: esp. when subject matter expertise required	**Cost saving**: Expertise can be versioned, shared, reused across organization
Non-private: Need to ship data to human annotators	**Privacy**: Create LFs using a cleared data subsample then apply LFs to other 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



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

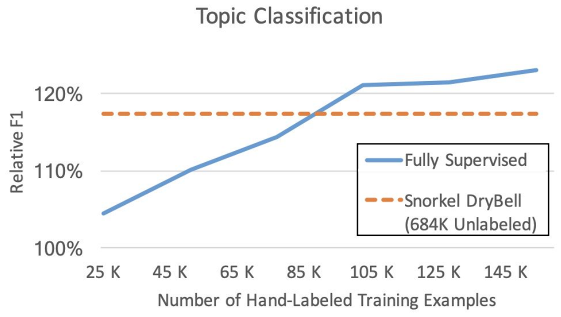
How to get more labeled training data?

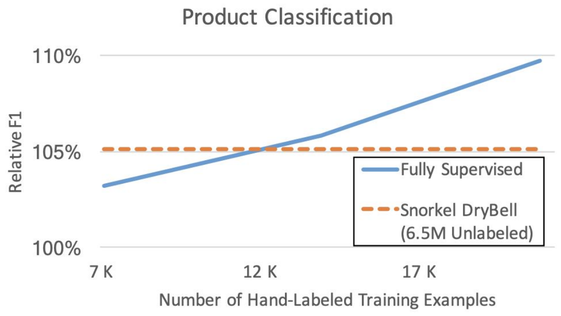

Traditional Supervision: Have subject matter experts (SMEs) hand-label more training data
Semi-supervised Learning: Use structural assumptions to automatically leverage unlabeled data
Weak Supervision: Get lower-quality labels more efficiently and/or at a higher abstraction level
Transfer Learning: Use models already trained on a different task
Too expensive!
Active Learning: Estimate which points are most valuable to solicit labels for
Get cheaper, lower-quality labels from non-experts
Get higher-level supervision over unlabeled data from SMEs
Use one or more (noisy / biased) pre-trained models to provide supervision
Heuristics
Distant Supervision
Constraints
Expected distributions
Invariances

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

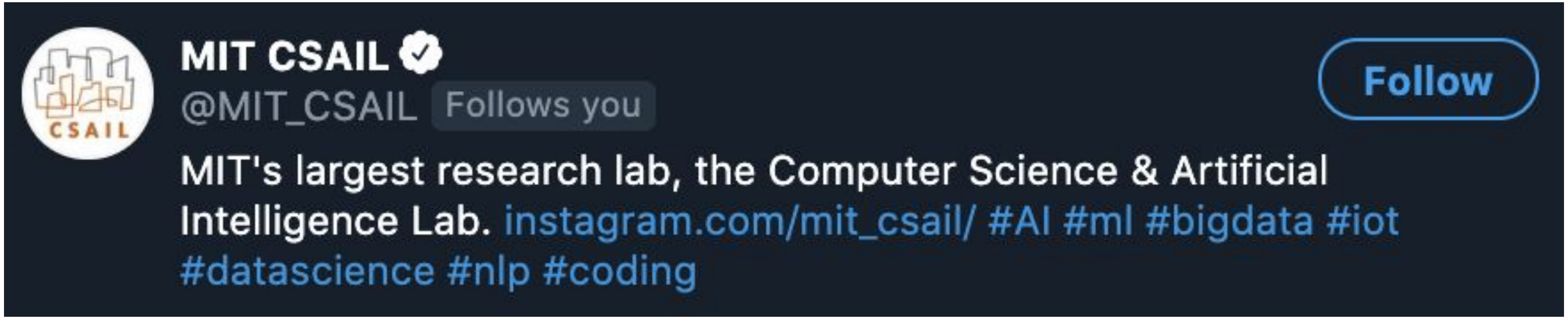




Harnessing Organizational Knowledge for Machine Learning (Ratner et al., Google AI Blog 2019)

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



MIT CSAIL
@MIT_CSAIL
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

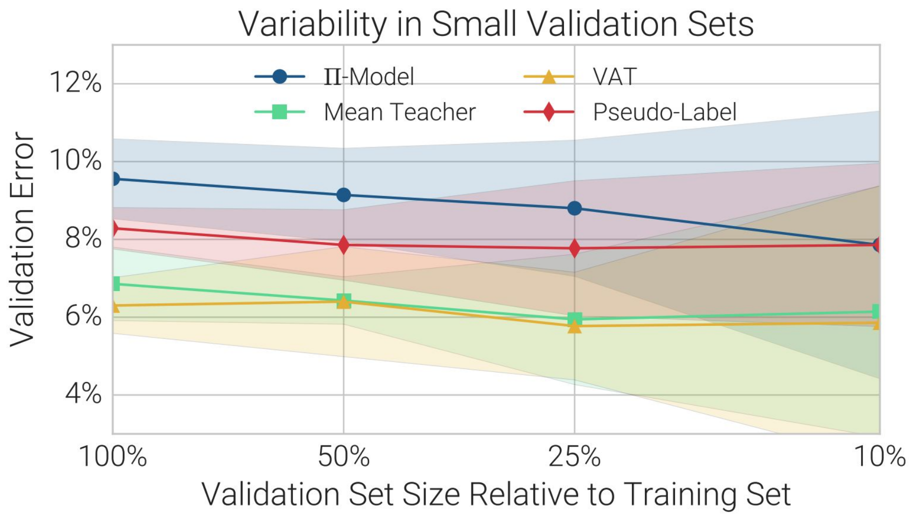
Assumption: small perturbation wouldn’t change a sample’s label

- 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



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

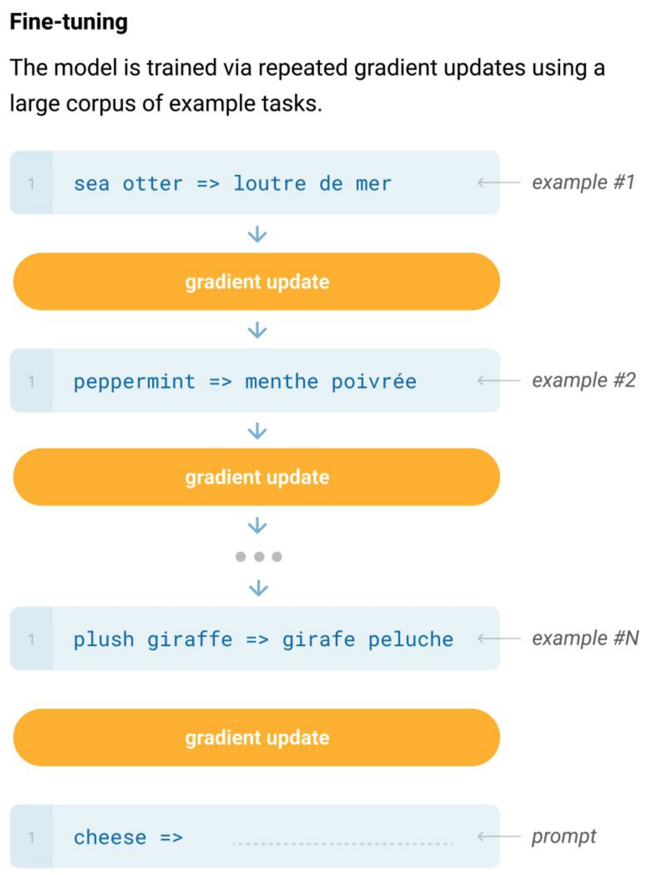
Transfer learning

Apply model trained for one task to another task

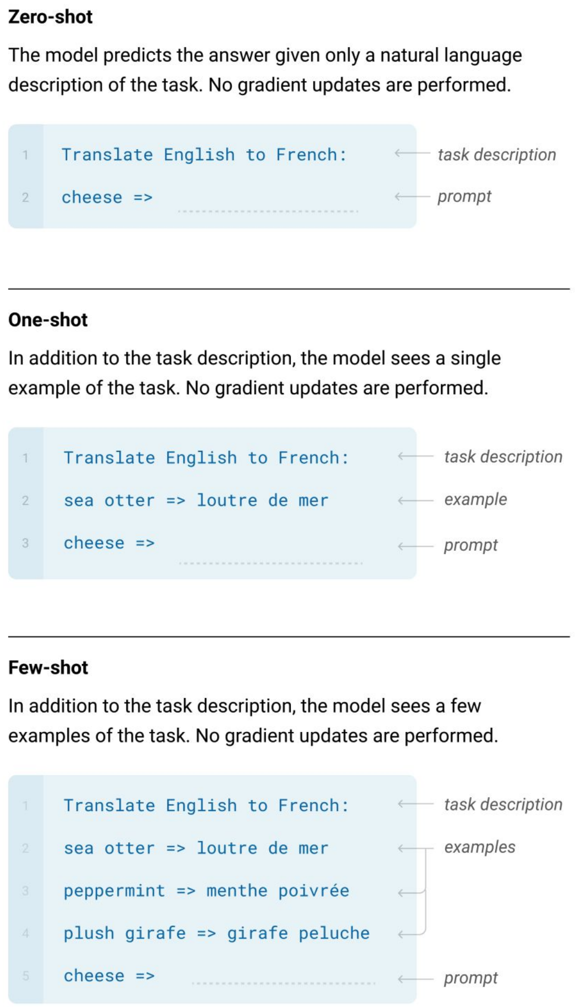
1. Fine-tuning
2. Prompt-based

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.
sea otter => loutre de mer
example #1
gradient update
peppermint => menthe poivrée
example #2
gradient update
plush giraffe => girafe peluche
example #N
gradient update
cheese =>
prompt

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.

```
Translate English to French:
cheese => 
```

task description
prompt

One-shot

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

```
Translate English to French:
sea otter => loutre de mer
cheese => 
```

task description
example
prompt

Few-shot

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

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese => 
```

Language Models are Few-Shot Learners (OpenAI 2020)

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 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

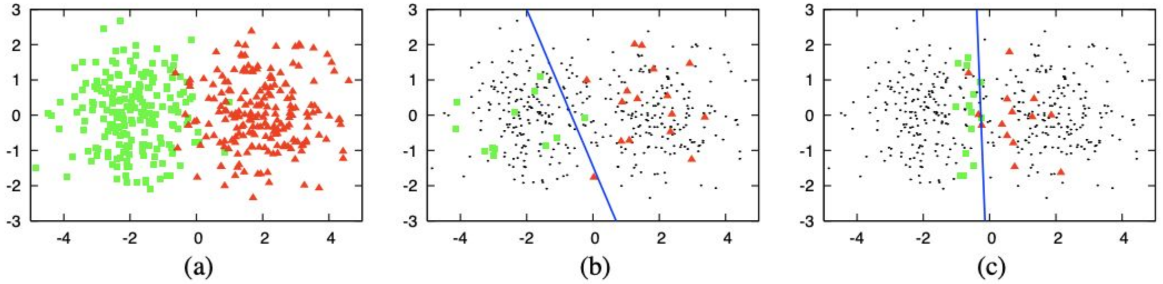


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem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

Machine Learning Systems Design

Data Lifecycle

Next Lecture: Data Preparation

CE 40959 Spring 2023

Ali Zarezade

SharifMLSD.github.io

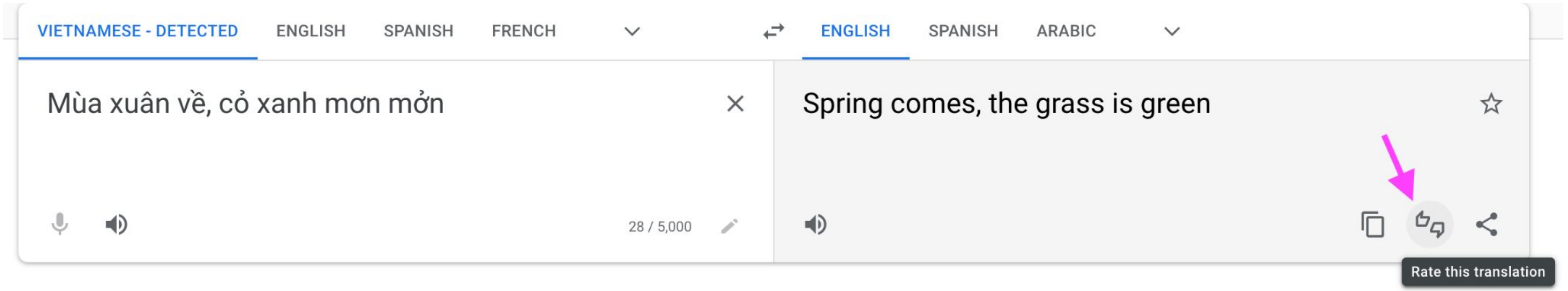
Natural labels & feedback loops

Natural labels

- The model’s predictions can be automatically evaluated or partially evaluated by the system.
- Examples:
    - ETA
    - Ride demand prediction
    - Stock price prediction
    - Ads CTR
    - Recommender system

Natural labels

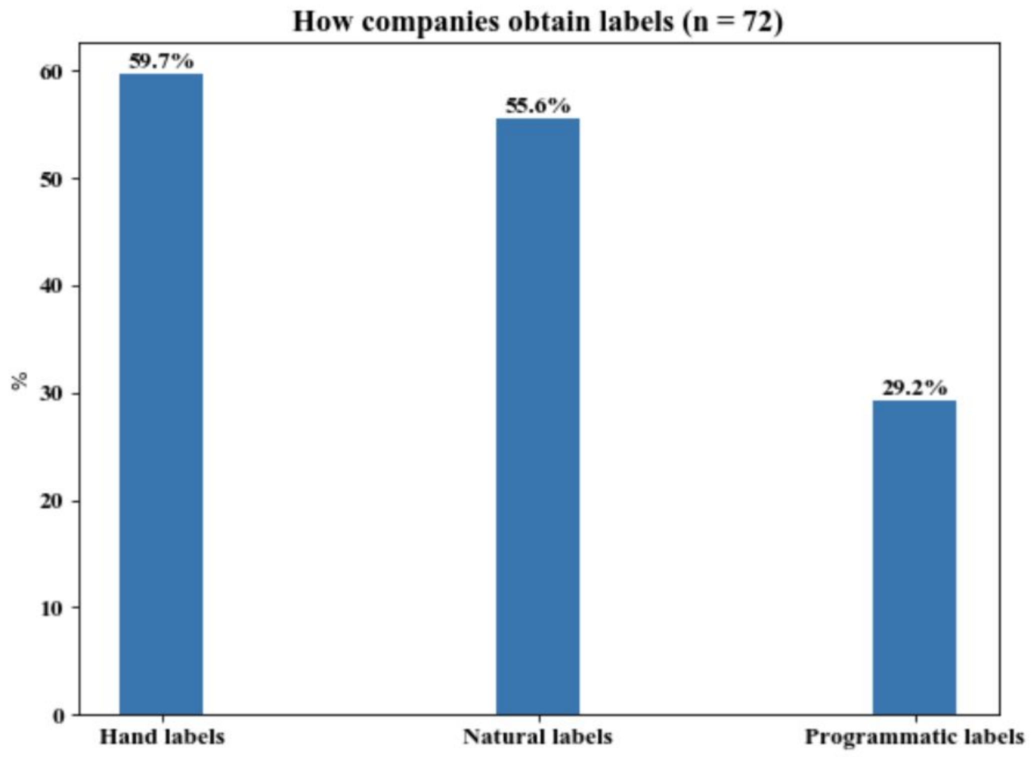
- You can engineer a task to have natural labels



Mùa xuân về, cỏ xanh mơn mởn

Spring comes, the grass is green

Natural labels: surprisingly common

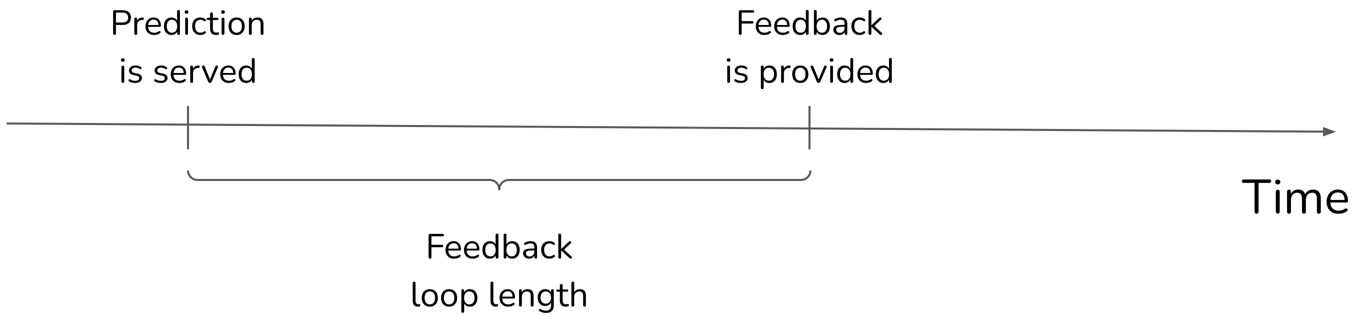


⚠️ Biases ⚠️

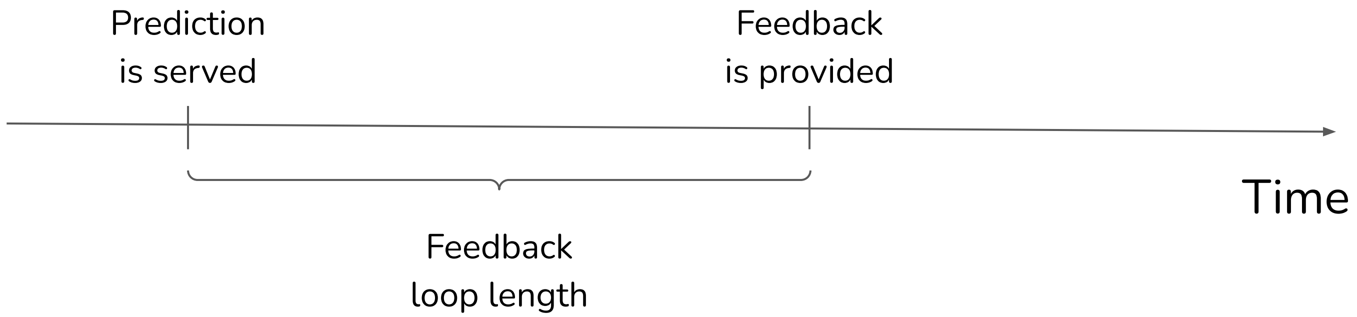
- Small sample size
- Companies might only use ML for tasks with natural labels

Claypot AI's real-time ML survey (2022)

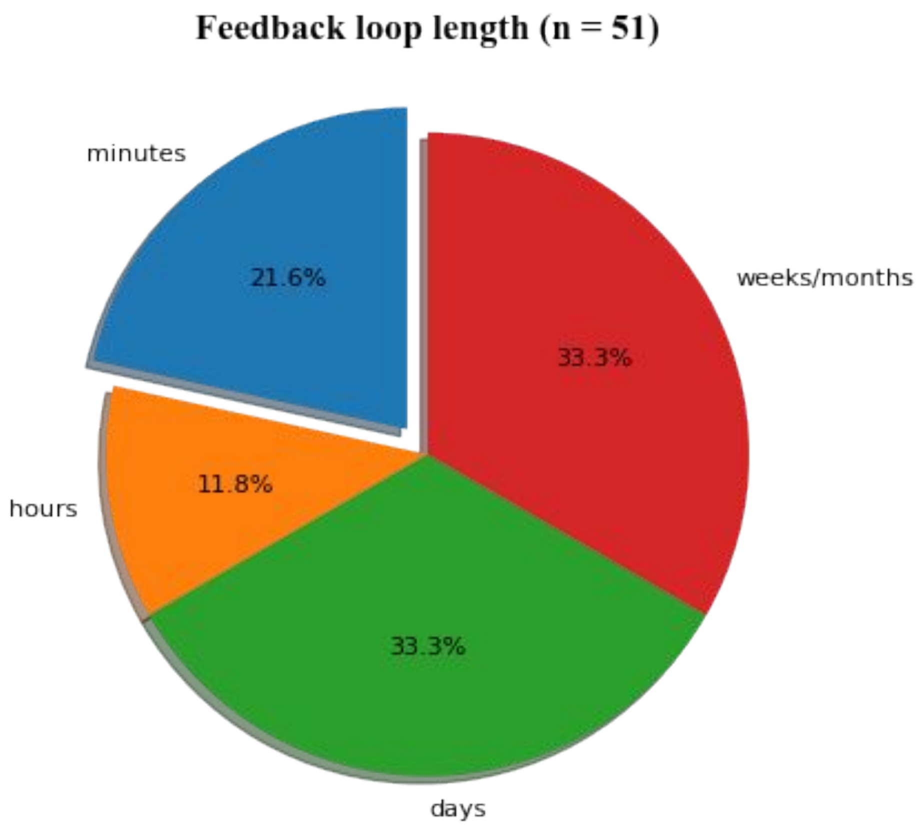
Delayed labels



Delayed labels



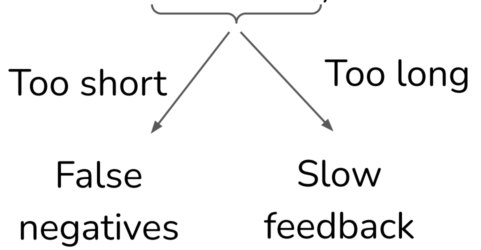
- Short feedback loop: minutes -> hours
  - Reddit / Twitter / TikTok's recommender systems
- Long feedback loop: weeks -> months
  - Stitch Fix's recommender systems
  - Fraud detection


Feedback loop length (n = 51)
minutes
21.6%
weeks/months
33.3%
hours
11.8%
33.3%
days

Perfectly balanced...
...As all things should be

⚠️ Labels are often assumed ⚠️

- Recommendation:
  - Click -> good rec
  - After X minutes, no click -> bad rec



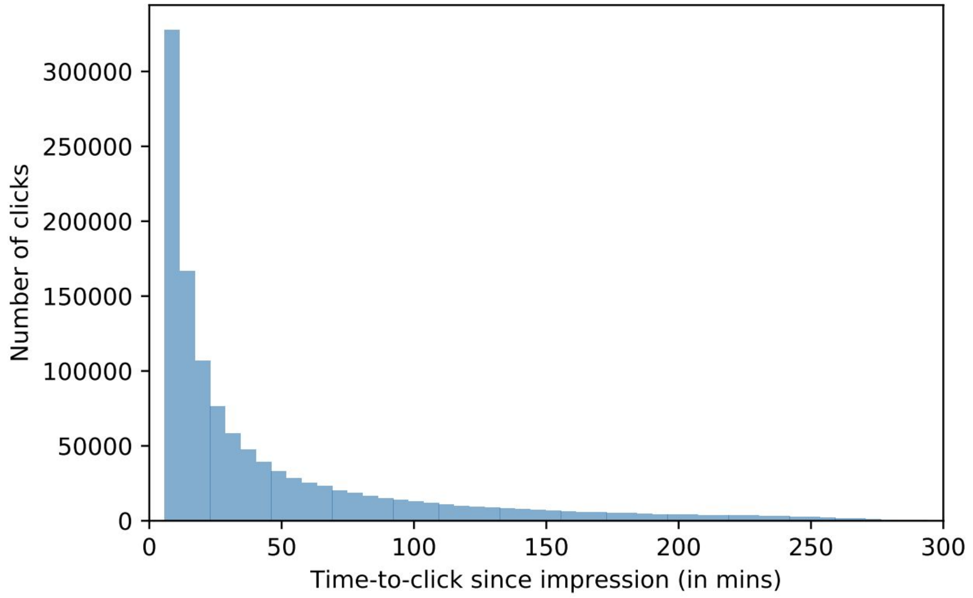
Speed vs. accuracy tradeoff

⚠️ Labels are often assumed ⚠️

- Recommendation:
  - Click -> good rec
  - After X minutes, no click -> bad rec

Too short → False negatives

Too long → Slow feedback



Addressing Delayed Feedback for Continuous Training with Neural Networks in CTR prediction (Ktena et al., 2019)