Data engineering - II

Creating training datasets

Data: full of potential for biases!

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

Stratification

- If we are comparing the death rates of smokers vs. non-smokers
- If we divide our sample population into just smokers/non-smokers and looked at death rates, we may have problems such as:
  - What if we had more men on the smoking side?
  - What if smokers also are more older and over-weight than non-smokers?

Stratification

- Stratification: Dividing samples into sub-groups or subunits prior to sampling.
- Need to do stratification for train-test split

Stratification and Imbalance

- However, stratification would not fix any imbalance problems in a dataset, rather it would keep it as it is!
- For example, in a recent work of ours, we has breathing sounds of 20 people and we would like to classify these people using these sounds.
- We have differing number o samples for each person: “The dataset contains a minimum of 89, a maximum of 710 and an average of 254 intraspeech breath instances per participant.”
- Now if we split the data as 80% training 20% test and when we measure accuracy, this score would be dominated by persons having higher number of samples.

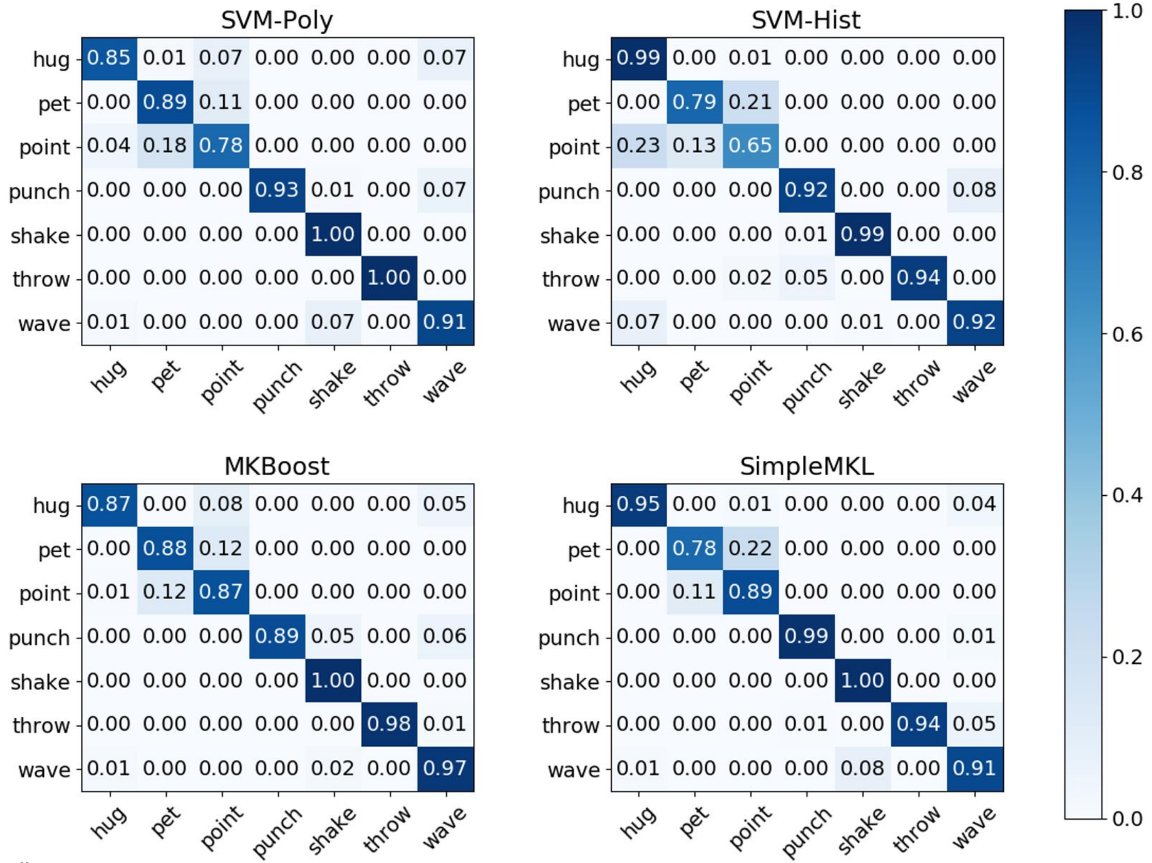
- Accuracy is not a very good metric when you have an imbalanced dataset

Stratification and Imbalance

- Solution 1: Create a balanced test set by taking the same number of samples from different classes

- Solution 2: Report the results per class (This would be preferable when there is a limited number of classes, i.e. persons in this case).
    - You can use confusion matrix to visualize.

Stratification and Imbalance



SVM-Poly

	hug	pet	point	punch	shake	throw	wave
hug	0.85	0.01	0.07	0.00	0.00	0.00	0.07
pet	0.00	0.89	0.11	0.00	0.00	0.00	0.00
point	0.04	0.18	0.78	0.00	0.00	0.00	0.00
punch	0.00	0.00	0.00	0.93	0.01	0.00	0.07
shake	0.00	0.00	0.00	0.00	1.00	0.00	0.00
throw	0.00	0.00	0.00	0.00	0.00	1.00	0.00
wave	0.01	0.00	0.00	0.00	0.07	0.00	0.91

SVM-Hist

	hug	pet	point	punch	shake	throw	wave
hug	0.99	0.00	0.01	0.00	0.00	0.00	0.00
pet	0.00	0.79	0.21	0.00	0.00	0.00	0.00
point	0.23	0.13	0.65	0.00	0.00	0.00	0.00
punch	0.00	0.00	0.00	0.92	0.00	0.00	0.08
shake	0.00	0.00	0.00	0.01	0.99	0.00	0.00
throw	0.00	0.00	0.02	0.05	0.00	0.94	0.00
wave	0.07	0.00	0.00	0.00	0.01	0.00	0.92

MKBoost

	hug	pet	point	punch	shake	throw	wave
hug	0.87	0.00	0.08	0.00	0.00	0.00	0.05
pet	0.00	0.88	0.12	0.00	0.00	0.00	0.00
point	0.01	0.12	0.87	0.00	0.00	0.00	0.00
punch	0.00	0.00	0.00	0.89	0.05	0.00	0.06
shake	0.00	0.00	0.00	0.00	1.00	0.00	0.00
throw	0.00	0.00	0.00	0.00	0.00	0.98	0.01
wave	0.01	0.00	0.00	0.00	0.02	0.00	0.97

SimpleMKL

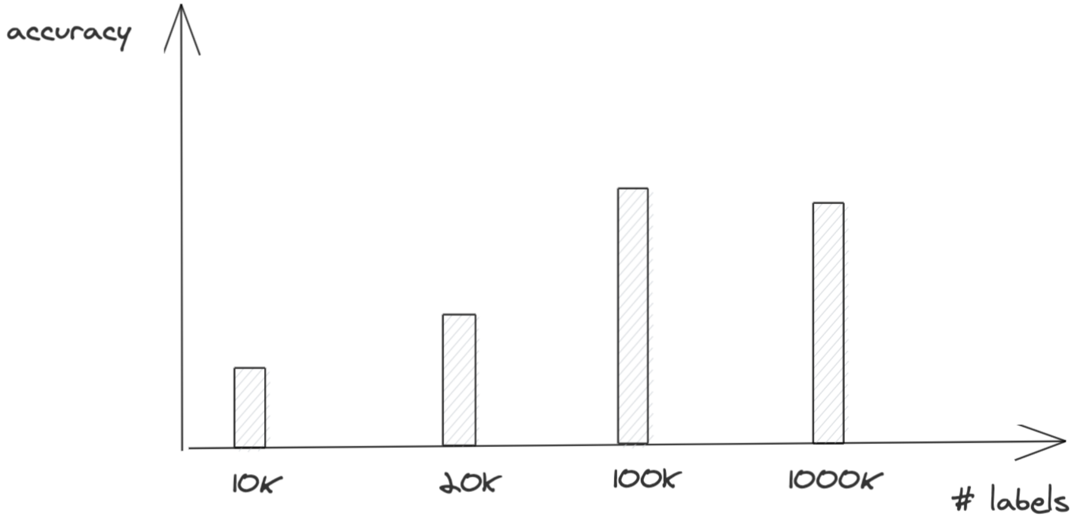
	hug	pet	point	punch	shake	throw	wave
hug	0.95	0.00	0.01	0.00	0.00	0.00	0.04
pet	0.00	0.78	0.22	0.00	0.00	0.00	0.00
point	0.00	0.11	0.89	0.00	0.00	0.00	0.00
punch	0.00	0.00	0.00	0.99	0.00	0.00	0.01
shake	0.00	0.00	0.00	0.00	1.00	0.00	0.00
throw	0.00	0.00	0.00	0.01	0.00	0.94	0.05
wave	0.01	0.00	0.00	0.00	0.08	0.00	0.91

M.A. Arabacı, F. Özkan, E. Surer, Peter Jancovic, A. Temizel, “Multi-modal Egocentric Activity Recognition using Multi-Kernel Learning”, Multimedia Tools and Applications, April 2021.

K-Fold Validation

- K-folds should be designed N different times with randomization and the folds need to be stored
- To compare performances of different methods, stored folds should be used to evaluate the methods on the same grounds
- It would be good to report standard deviation of the scores in addition to mean value.

More data is not always better



Why is the model getting worse?

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.

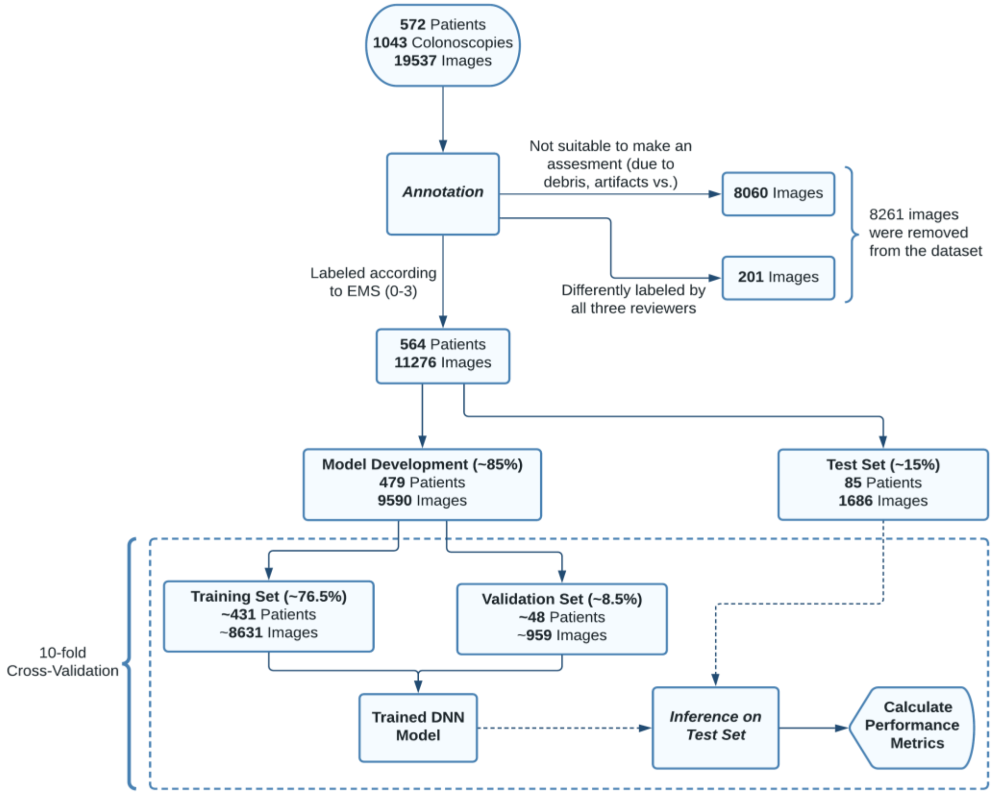
Annotator	# entities	Annotation
1	3	[**Darth Sidious**], known simply as the Emperor, was a [**Dark Lord of the Sith**] who reigned over the galaxy as [**Galactic Emperor of the First Galactic Empire**]
2	6	[**Darth Sidious**], known simply as the [**Emperor]**, was a **[Dark Lord**] of the [**Sith**]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



Slide credit: Görkem Polat

Label multiplicity: some recommendations

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

For example, clarify with them to pick the entity that comprises the longest substring:
Galactic Emperor of the First Galactic Empire instead of **Galactic Emperor** and **First Galactic Empire**.

Label multiplicity: solution

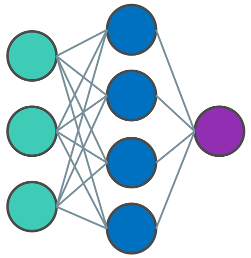
- Clear problem definition
- Annotation training
- Data lineage: track where data/labels come from
- Learning methods with noisy labels
  - Learning with Noisy Labels (Natarajan et al., 2013)
  - Loss factorization, weakly supervised learning and label noise robustness (Patrini et al., 2016)
  - Cost-Sensitive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

Training data is the bottleneck

Data

+

Algorithms



=

ML Model

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

- 1-2 days
- <1 pt. differences

How to get training data in days?

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

Hand labeling data is ...

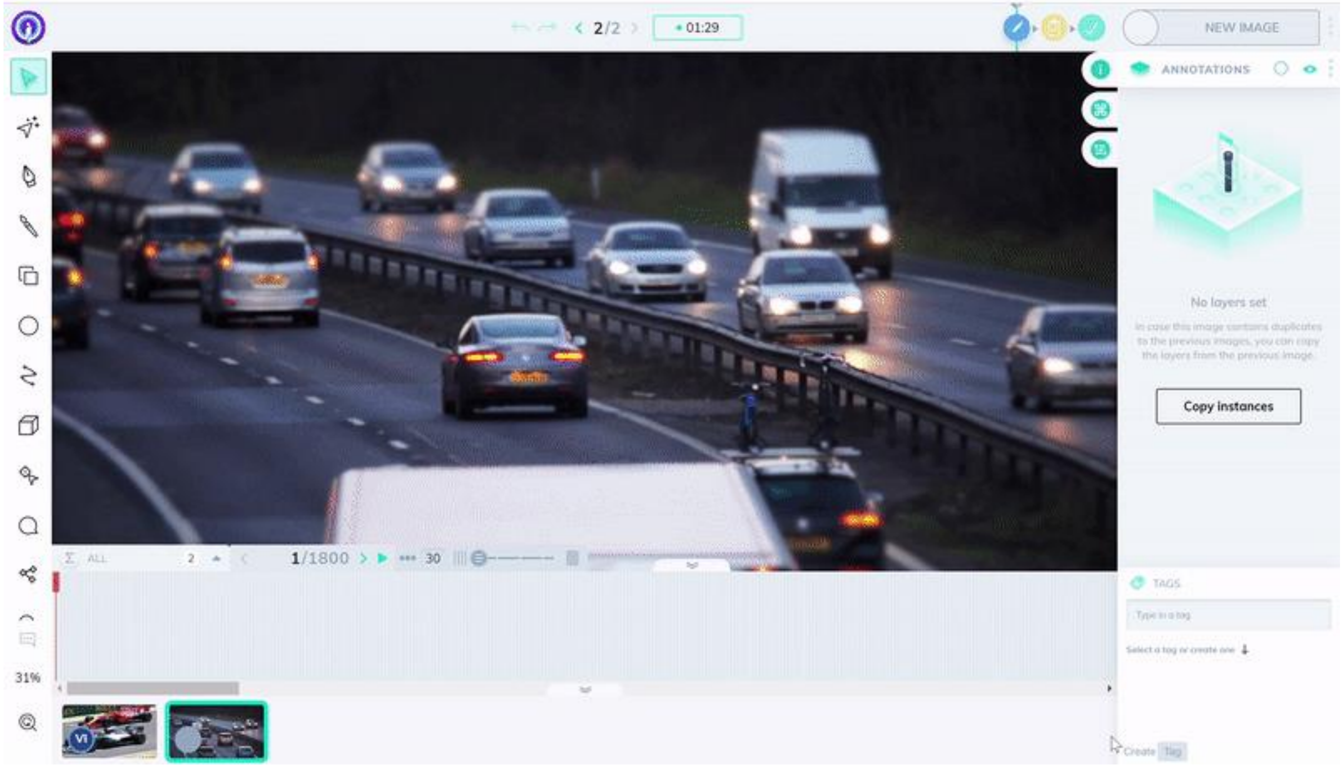
- **Expensive:** Esp. when subject matter expertise required
- **Non-private:** Need to ship data to human annotators
- **Slow:** Time required scales linearly with # labels needed
- **Non-adaptive:** Every change requires re-labeling the dataset

Programmatic labeling

Programmatic data labeling is **the process of using scripts to automatically label data**.

This process can automate tasks including image and text annotation, which eliminates the need for large numbers of human labelers.

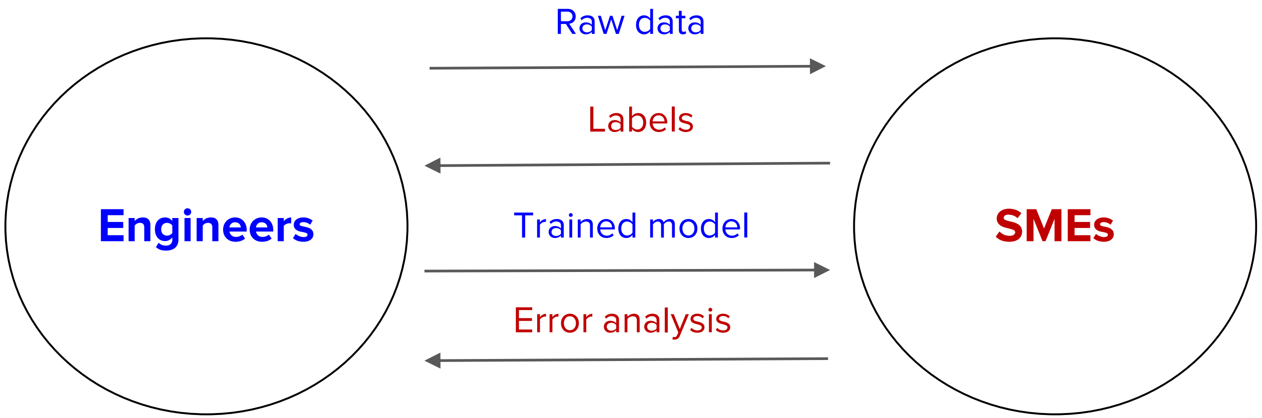
Annotation Tools



Darwin V7

https://humansintheloop.org/best-annotation-tools-for-computer-vision-of-2021/

Cross-functional communication



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

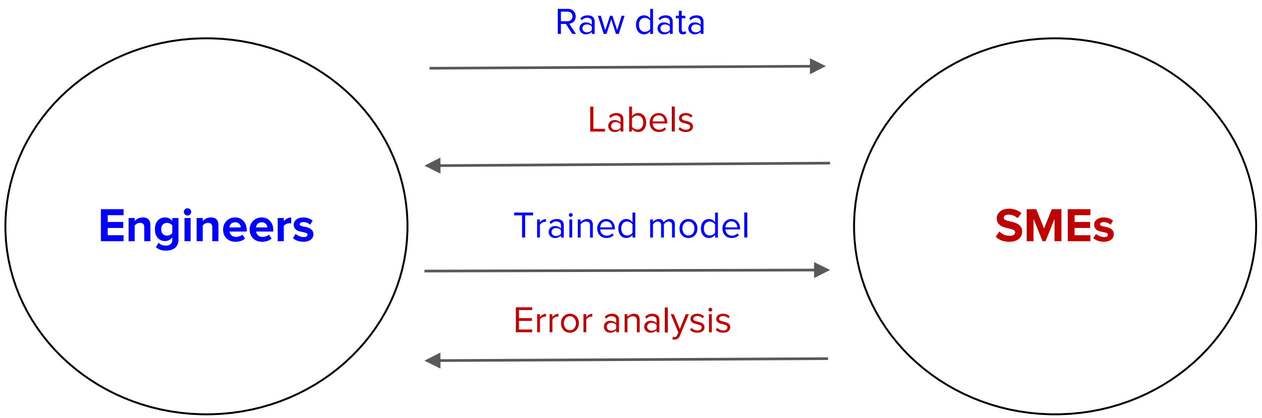
```
If the nurse's note mentions
serious conditions like pneumonia,
the patient's case should be given
priority consideration.
```

Code: version control, reuse, share

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

Snorkel

SME as labeling functions



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

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

Labeling functions (LFs): Encode SME heuristics as functions 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: 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** …"

??

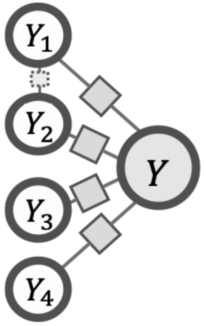
- **Noisy**: Unknown, inaccurate
- **Overlapping**: LFs may be correlated
- **Conflicting**: different LFs give different labels
- **Narrow**: Don't generalize well

LF labels are combined to generate ground truths

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

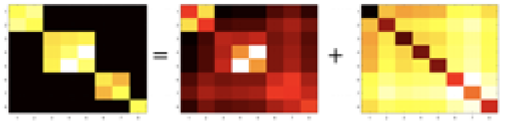
```
def LF_from_grandma(x):
    if x.sender.name is "Grandma":
        return "HAM"
```

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



[Intuition]

Look at agreements & disagreements



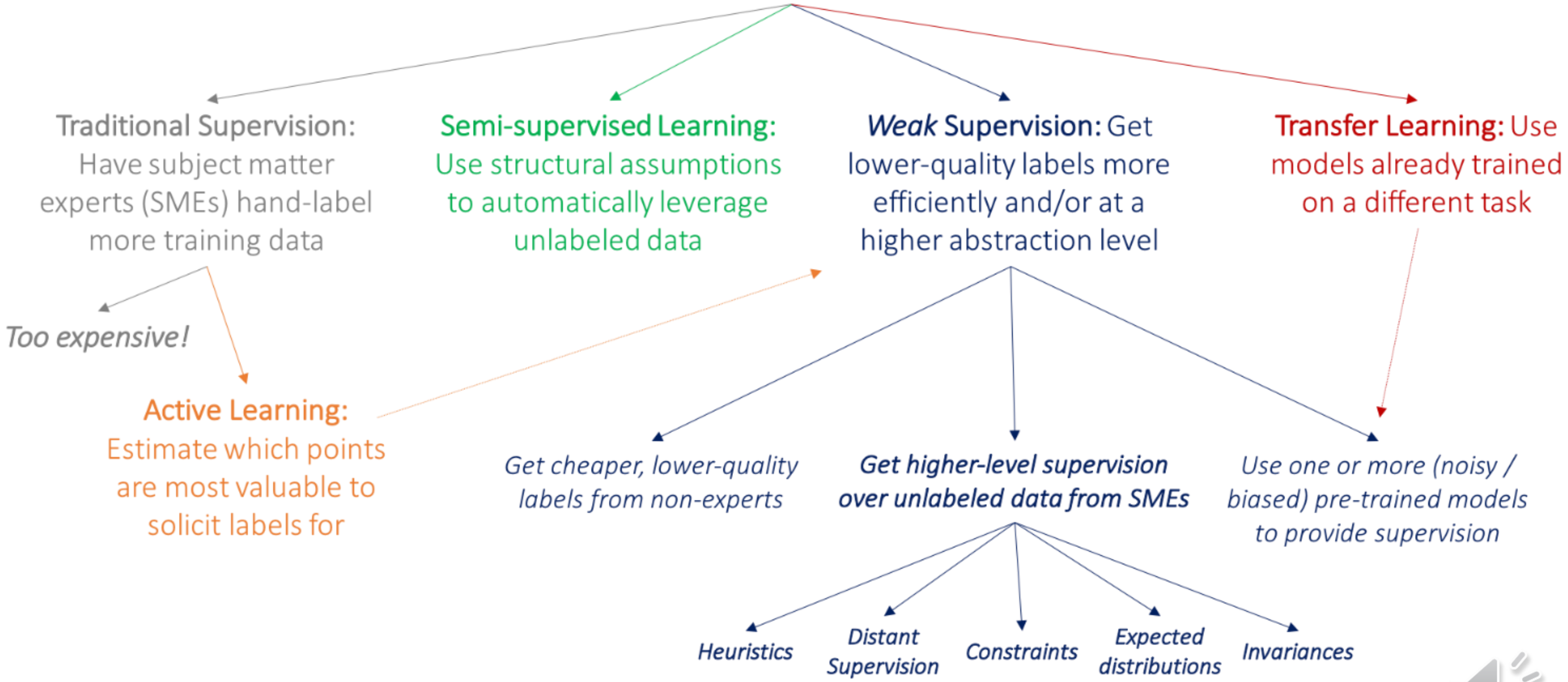
$$(\Sigma^{-1})_O = \Sigma_O^{-1} + zz^T$$

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!

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)

Active learning

- Label only samples that are estimated to be most valuable to the model
    - e.g. label only CT scans close to the decision boundary

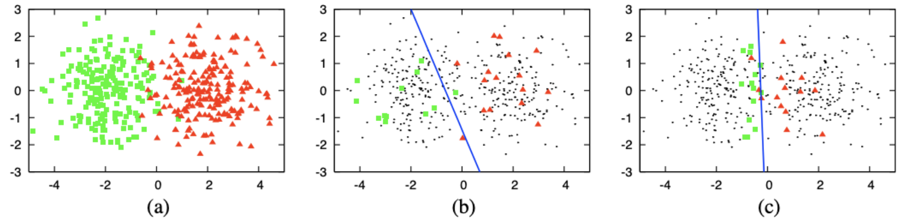


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem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

Self Supervised Learning (SSL)

Self-supervised learning (or self-supervision) is a relatively recent learning technique where the training data is autonomously labelled.

It is still supervised learning, but the datasets do not need to be manually labelled by a human,

They can e.g. be labelled by finding and exploiting the relations (or correlations) between different input signals (that is, input coming from different sensor modalities).

https://ai.stackexchange.com/questions/10623/what-is-self-supervised-learning-in-machine-learning

Self Supervised Learning (SSL)

For example, consider a robot which is equipped with a proximity sensor (which is a short-range sensor capable of detecting objects in front of the robot at short distances) and a camera (which is long-range sensor, but which does not provide a direct way of detecting objects).

Consider now the task of detecting objects in front of the robot at longer ranges than the range the proximity sensor allows. In general, we could train a CNN to achieve that.

However, to train such CNN, in supervised learning, we would first need a labelled dataset, which contains labelled images (or videos), where the labels could e.g. be "object in the image" or "no object in the image".

In supervised learning, this dataset would need to be manually labelled by a human, which clearly would require a lot of work.

https://ai.stackexchange.com/questions/10623/what-is-self-supervised-learning-in-machine-learning

Self Supervised Learning (SSL)

To overcome this issue, we can use a self-supervised learning approach. In this example, the basic idea is to associate the output of the proximity sensors at a time step t′>t with the output of the camera at time step t (a smaller time step than t′).

At time *t′*, the robot is at position (*x′*,*y′*). At time step *t′*, the output of the proximity sensor will e.g. be "object in front of the robot" or "no object in front of the robot".

Without loss of generality, suppose that the output of the proximity sensor at *t′*>*t* is "no object in front of the robot", then the label associated with the output of the camera (an image frame) at time *t* will be "no object in front of the robot".

For more details: Learning to Perceive Long-Range Obstacles Using Self-Supervision from Short-Range Sensors

M Nava, J Guzzi, RO Chavez-Garcia, LM Gambardella, A Giusti, IEEE Robotics and Automation Letters 4 (2), 1279-1286, 2019

https://ai.stackexchange.com/questions/10623/what-is-self-supervised-learning-in-machine-learning

Self-supervised vs. supervised learning

Self-supervised Learning is supervised learning because its goal is to learn a function from pairs of inputs and labeled outputs.

Explicit use of labeled input-outputs pairs in self-supervised learning is not needed. Instead, correlations, embedded metadata, or domain knowledge available within the input is implicitly and autonomously extracted from the data and used as supervisory signals.

Like supervised learning, self-supervised learning has use cases in regression and classification.

https://hackernoon.com/self-supervised-learning-gets-us-closer-to-autonomous-learning-be77e6c86b5a

Self-supervised vs. unsupervised learning

Self-supervised learning is like unsupervised Learning because the system learns without using explicitly-provided labels.

It is different from unsupervised learning because we are not learning the inherent structure of data.

Self-supervised learning, unlike unsupervised learning, is not centered around clustering and grouping, dimensionality reduction, recommendation engines, density estimation, or anomaly detection.

https://hackernoon.com/self-supervised-learning-gets-us-closer-to-autonomous-learning-be77e6c86b5a

Semi-supervised learning

Semi-supervised learning is a class of machine learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data.

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy.

https://www.wikiwand.com/en/Semi-supervised_learning

Self-supervised vs. semi-supervised learning

Combination of labeled and unlabeled data is used to train semi-supervised learning algorithms, where smaller amounts of labeled data in conjunction with large amounts of unlabeled data can speed up learning tasks.

Self-supervised learning is different as systems learn entirely without using explicitly-provided labels.

https://hackernoon.com/self-supervised-learning-gets-us-closer-to-autonomous-learning-be77e6c86b5a

Transfer Learning with CNNs

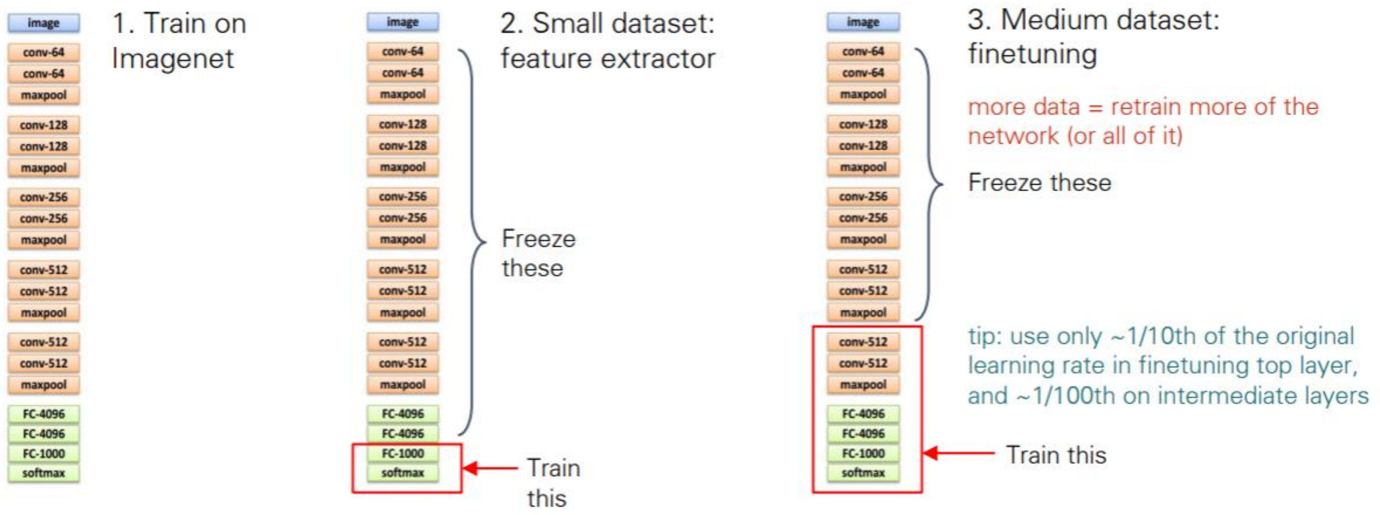
- A CNN trained on a (large enough) dataset generalizes to other visual tasks

A. Joulin, L.J.P. van der Maaten, A. Jabri, and N. Vasilache **Learning visual features from Large Weakly supervised Data.** ECCV 2016

Slide credit: Joan Bruna

Transfer Learning with CNNs

- Keep layers 1-7 of our ImageNet-trained model fixed
- Train a new softmax classifier on top using the training images of the new dataset.



more data = retrain more of the network (or all of it)

tip: use only ~1/10th of the original learning rate in finetuning top layer, and ~1/100th on intermediate layers