Weak Supervision Overview

Chris Ré
Stanford University
Various techniques for limited labeled data

- **Active learning**: Select points to label more intelligently
- **Semi-supervised learning**: Use unlabeled data as well
- **Transfer learning**: Transfer from one training dataset to a new task
- **Weak supervision**: Label data in cheaper, higher-level ways

This lecture.

https://www.snorkel.org/blog/weak-supervision
Biased by active on-going work.
ML Application =

Model + Data + Hardware

State-of-the-art models and hardware are available.
Training data is not
But supervision comes from god herself....
... but training data usually comes from a dirty, messy process.

Can we provide mathematical and systems structure for this messy process?
Supervision is where the action is...

Model differences overrated, and supervision differences underrated.
Automated Chest X-ray Triage

Optimizing Workflows with Automated Prioritization, Radiology 19

What’s the Problem?

Too many of these!

Radiologist shortage leaves patient care at risk, warns royal college

*Bmj* 2017; 359 doi: https://doi.org/10.1136/bmj.j4683 (Published 11 October 2017)
Cite this as: *Bmj* 2017;359:j4683

Improving Patient Safety: Avoiding Unread Imaging Exams in the National VA Enterprise Electronic Health Record.

Bastawrous S1,2, Carney B3.
Is Deep Learning the Answer?

This is not an easy question...
- No benchmark dataset
- Effects of data quality are unclear
- No assessment of existing algorithms
- No feedback from clinical community

...so we spent a year trying to answer it!
- Created large dataset of clinical labels
- Evaluated effect of label quality
- Work published in a clinical journal

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOVW + KSVM</td>
<td>0.88</td>
</tr>
<tr>
<td>AlexNet</td>
<td>0.87</td>
</tr>
<tr>
<td>ResNet-18</td>
<td>0.89</td>
</tr>
<tr>
<td>DenseNet-121</td>
<td>0.91</td>
</tr>
</tbody>
</table>

Often: Differences in models ~ 2-3 points.

Later: Label quality & quantity > model choice.
Even in Benchmarks: Data Augmentation is Critical

Ex: 13.4 pt. avg. accuracy gain from data augmentation across top ten CIFAR-100 models—difference in top-10 models is less!
Training Signal is key to pushing SotA

New methods for gathering signal leading the state of the art

Google AI: AutoAugment: Using learned data augmentation policies (RandAugment)
  • Augmentation Policies first in Ratner et al. NIPS ’17

Facebook Hash tag weakly supervised pre-training
  • Pre-train using a massive dataset with weak training signal
Training data: the new bottleneck

Slow, expensive, and static
Manual Labels

- Slow
- Expensive
- Static

- Labels
- Time

- $10 - $100/hr

Programmatic Labels

- Fast
- Cheap
- Dynamic

- Labels
- Time

- write programs
- run programs

- aws
- $0.10/hr

Trade-off: programmatic labels are noisy...
Key Idea: Model Training Creation Process

This talk:

1. An interface for generating training data via weak supervision
2. An approach to learn quality and correlations of sources
3. Training an end model in various domains
Snorkel: Formalizing Programmatic Labeling

Observation: Weak supervision applied in ad hoc and isolated ways.
Snorkel: Formalizing Programmatic Labeling

Goal: Replace *ad hoc* weak supervision with a formal, unified, theoretically grounded approach for programmatic labeling
Running Example: NER

Dr. Bob Jones is a specialist in cardiomyopathy treatment, leading the cardiology division at Saint Francis.

Goal: Label training data using weak supervision strategies for these three tasks
Weak Supervision as Labeling Functions

Dr. Bob Jones is a specialist in cardiomyopathy treatment, leading the cardiology division at Saint Francis.

Problem: These noisy sources conflict and are correlated.
The Snorkel Pipeline

1. Users write labeling functions to generate noisy labels

2. Snorkel models and combines the noisy labels into probabilities

3. The resulting probabilistic labels train a model

KEY IDEA: Probabilistic training point carries accuracy. No hand labeled data needed.
Reason #1: Improved Generalization

Empirically, the end model boosts recall by 43% on average!
Reason #1: Improved Generalization

Task: identify disease-causing chemicals

Phrases mentioned in LFs:

“treats”, “causes”, “induces”, “prevents”, …

Phrases given large weights by end model:

“could produce a”, “support diagnosis of”, …

The end model learned to take advantage of features that were helpful for prediction, but never explicitly mentioned in the LFs
Reason #2: Scaling with Unlabeled Data

Add more unlabeled data—without changing the LFs—and performance improves!
Reason #3: Cross-Model Supervision

**Available at test time**

This is servable!

**Not available at test time**

Not servable

Use training data as a medium for knowledge transfer

---

Report 47:

**Indication:** Chest pain.
**Findings:** Pneumothorax.
**Operation recommended.**

---

**Hours of weak supervision**

matches manual labels collected over **person years**!
Manual Labels

Slow

Expensive

Manual Labels

Static

{Positive, Negative}

$10 - $100/hr

Programmatic Labels

Fast

Cheap

Programmatic Labels

Dynamic

{Positive, Neutral, Negative}

write programs

run programs

Fast

Cheap

Easy

Home

write programs

run programs

$0.10/hr
Snorkel: In use at the world’s largest companies

Snorkel DryBell” collaboration with Google Ads. Bach et al. SIGMOD19.

Used in production in many industries, startups, and other tech companies!
Collaboration Highlight: Google + Snorkel

• **Snorkel DryBell** is a production version of Snorkel focused on:
  • Using *organizational knowledge resources* to train ML models
  • Handling *web-scale* data
  • Non-servable to servable feature transfer.

[Bach et. al., SIGMOD 2019]
Let’s look under the hood and take a peak at some math
The Snorkel Pipeline

1. Users write labeling functions for multiple related tasks

2. We model the labeling functions' behavior to de-noise them

3. We use the probabilistic labels to train a multi-task model

No hand-labeled training data!


Users write labeling functions for multiple related tasks.

We model the labeling functions' behavior to de-noise them.

We use the probabilistic labels to train a multi-task model.

How to represent diverse sources of weak supervision?
Users write labeling functions for multiple related tasks. We model the labeling functions' behavior to de-noise them. We use the probabilistic labels to train a multi-task model.

How can we do anything without the ground truth labels?
def existing_classifier(x):
    return off_shelf_classifier(x)

def upper_case_existing_classifier(x):
    if all(map(is_upper, x.split())) and \    off_shelf_classifier(x) == 'PERSON':
        return 'PERSON'

def is_in_hospital_name_DB(x):
    if x in HOSPITAL_NAMES_DB:
        return 'HOSPITAL'

Model as Generative Process

How to learn the parameters of this model (accuracies & correlations) without $Y$?
Intuition: Learn from the Overlaps

Sources.

- `def existing_classifier(x):`
  - `return off_shelf_classifier(x)`

- `def upper_case_existing_classifier(x):`
  - `if all(map(is_upper, x.split())) and \`
    - `off_shelf_classifier(x) == ‘PERSON’:`
    - `return PERSON`

- `def is_in_hospital_name_DB(x):`
  - `if x in HOSPITAL_NAMES_DB:
    - `return HOSPITAL`

Key idea: We can observe overlapping judgements on many points!