Machine Learning is Changing Software

Chris Ré

Stanford University

http://hazyresearch.stanford.edu/
Software 2.0 is eating Software 1.0

1000x Productivity: Google shrinks language translation code from 500k LoC to 500 lines of dataflow.

Classical problems ML 1

- ETL & Cleaning (Inductiv.ai)
- DB Tuning OtterTune (CMU)
- Networks Pensieve (MIT)
- NeuroCore (Stanford)


“Software 2.0”, Andrej Karpathy, https://medium.com/@karpathy/software-2-0-a64152b37c35
Easier to build, deploy, and maintain

**Build** products faster. Speed is amazing.

**Deploy** is critical: NNs “new JVM”
- Dataflow has regular run-times.
- Qualification easier means “ship faster.”
- See Kunle’s ISCA/NeurIPS keynote for more info.

**Maintain**: ”retrain”—no “ninja” dependence.

**SW2.0 View**: eng. changes are **significant**.

Kunle Olukotun
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

We spent a year on this challenge
• 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.

Label quality & quantity > model choice.
Simple 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**
  - **Augmentation Policies** first in Ratner et al. NIPS ’17

- **Facebook Hash tag weakly supervised pre-training**
  - Pre-train using a massive dataset with **hashtags**

[Images of Henry Ehrenberg and Alex Ratner (to: Washington)
Sharon Y. Li (to: Wisconsin)]
Automating the Art of Data Augmentation

Part I Overview

Our approach: Uncertainty-based sampling [ICML20]

- **Key idea:** Instead of randomly sampling, reduce the frequencies of transformations that the neural net has learned!
- **Empirical result:** 84.54% on CIFAR-100 using Wide-ResNet-28-10, improves RandAugment (Cubuk et al.’19) by 1.24%.
- **Theory:** Analyze the effect of different transformations in a high-dimensional setting, including revealing the regularization effect of a curious mixup augmentation!

**Blog post:** hazyresearch.stanford.edu/data-aug-part-3, **Code:** https://bit.ly/32E2V7n
Training data: the new bottleneck

Slow, expensive, and static
Manual Labels

- Slow
- Expensive
- Static

- Manual Labels
- $10 - $100/hr
- {Positive, Negative}

Programmatic Labels

- Fast
- Cheap
- Dynamic

- Programmatic Labels
- $0.10/hr
- {Positive, Neutral, Negative}

Trade-off: programmatic labels are noisy…
Snorkel: Formalizing Programmatic Labeling

**Pattern Matching**

\[ \text{regex.match(} \]
\[ r"\{A\} \text{ is caused by } \{B\}" \]
\[ ) \]

[e.g. Hearst 1992, Snow 2004]

**Distant Supervision**

\[ \text{Subset A} \]
\[ \text{Subset B} \]
\[ \text{Subset C} \]

[e.g. Mintz 2009]

**Augmentation**

“Change abbreviate names, and replace…”

**Topic Models**

[e.g. Hingmire 2014]

**Third-Party Models**

[e.g. Schapire 1998]

**Crowdsourcing**

[e.g. Dalvi 2013]

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.
The Real Work

Stephen Bach  Braden Hancock  Henry Ehrenberg  Alex Ratner  Paroma Varma

Snorkel.org
Running Example: NER

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

Let’s look at labeling “Person” versus “Hospital”

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

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

```
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
```

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

Users write labeling functions to generate noisy labels

Snorkel models and combines the noisy labels into probabilities

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

Use programmatic labeling for knowledge transfer

Report 47:

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

Not available at test time

*Not servable*

Available at test time

*This is servable!*

**ABNORMAL**

Hours of weak supervision matches manual labels collected over person years!
Snorkel: In use at the world’s largest companies

Http://snorkel.org

“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.

Thank you, Google!
(More soon)
You probably have used it...

Overton: A Data System for Monitoring and Improving Machine-Learned Products

Christopher Ré
Apple

Feng Niu
Apple

Pallavi Gudipati
Apple

Charles Srisuwananukorn
Apple

Migrating a Privacy-Safe Information Extraction System to a Software 2.0 Design

Ying Sheng
Google
Mountain View, CA, USA
yingsheng@google.com

Nguyen Vo
Google
Mountain View, CA, USA
nguyenvo@google.com

James B. Wendt
Google
Mountain View, CA, USA
jwendt@google.com

Sandeep Tata
Google
Mountain View, CA, USA
tata@google.com

Marc Najork
Google
Mountain View, CA, USA
najork@google.com
It has changed use real systems...

<table>
<thead>
<tr>
<th>Resourcing</th>
<th>Error Reduction</th>
<th>Amount of Weak Supervision</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>65% (2.9×)</td>
<td>80%</td>
</tr>
<tr>
<td>Medium</td>
<td>82% (5.6×)</td>
<td>96%</td>
</tr>
<tr>
<td>Medium</td>
<td>72% (3.6×)</td>
<td>98%</td>
</tr>
<tr>
<td>Low</td>
<td>40% (1.7×)</td>
<td>99%</td>
</tr>
</tbody>
</table>

A couple of highlights

- Used by multiple teams with good error reduction over production.
- Take away: many systems are almost entirely weak supervision based.
Weak Supervision in Science & Medicine

Cross-Modal Weak Supervision

"Indication: Chest pain. Findings: No focal consolidation or pneumothorax."

Auxiliary modality \( x^A \)

LABELING FUNCTIONS (Uf)

GENERATIVE MODEL

LSTM TEXT MODEL

TARGET MODALITY END MODEL

TARGET MODALITY

Probabilistic training label

\( y \)

\( \text{OPTIMIZER} \)


Blog: http://hazyresearch.stanford.edu/ws4science

A. Callahan et al., NPJ Dig Med, 2020

V. Kuleshov et al., Nat Comms, 2019

Imaging & Diagnostics

J. Fries et al., Nat Comms, 2019

J. Dunnmon et al., Radiology, 2019

K. Saab et al., NPJ Dig Med, 2020
High-Level Related Work

Software 2.0

Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale
Let’s look under the hood and take a peak at some math
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?
Model as Generative Process

```
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
```

How to learn the parameters of this model (accuracies & correlations) without $Y$?
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

Intuition: Learn from the Overlaps

Key idea: We observe agreements (+1) and disagreements (-1) on many points!
Just knowing the lineage is powerful!

- Wrote several high-precision, low-coverage distant supervision rules... 10k points @ 90%
- ... and one low-precision, high-coverage rule ... 1M points @ 60%

Without modeling lineage, isomorphic to 60.3% accuracy!

Sources correlated and noisy—want to estimate without labeled data
Just knowing the lineage is powerful!

- Wrote several high-precision, low-coverage distant supervision rules... 10k points @ 90%
- ... and one low-precision, high-coverage rule ... 1M points @ 60%

Without modeling lineage, isomorphic to 60.3% accuracy!

Sources correlated and noisy—want to estimate without labeled data
Solution Sketch: Using the covariance

Measure expected agreement rates of labeling functions.

Can only observe *part* of the covariance... if we observe rest, we'd be done! ($E[y \lambda] \sim \text{accuracy}$)
Idea: Use graph-sparsity of the inverse

\[ \Sigma_0 \]

\[
(\Sigma^{-1})_0 = \Sigma_0 + \mathbf{z} \mathbf{z}^T \\
\text{Rank-1 params to solve for (\sim function of accuracies)}
\]

Observed overlaps

Roughly, zero where corresponding pair of variables has no edge [Loh & Wainwright 2013, Ratner et al. 2019]

For now, assume we know the graph (dependency structure)...

matrix inversion lemma

\[
\lambda_1 \lambda_2 \lambda_3 \lambda_1 \lambda_2 \lambda_3 \lambda_1 \lambda_2 \lambda_3 \lambda_1 \lambda_2 \lambda_3 \lambda_1 \lambda_2 \lambda_3 \lambda_1 \lambda_2 \lambda_3
\]
Result: A matrix completion problem?

For any pair \( i \neq j \) with no edge in graph—the lhs is 0

\[
0 = \left( \Sigma^{-1}_O \right)_{i,j} + Z_i Z_j
\]

Observed overlaps

Low-rank parameters to solve for

Panic: \( \Sigma \) is full rank, so can’t use matrix completion…

Key: \( \Sigma = I + uu^T \) for some \( u \) so intuitively close…
Couple of Technical Comments

\[ 0 = \left( \Sigma_0^{-1} \right)_{i,j} + Z_i Z_j \]

- Symmetry: \( z \) and \(-z\) are solutions? What does this mean?
- \( z_i = 0 \) when accuracy 0.5, i.e., total noise! (more samples)
- Effective rank \( \text{er}(\Sigma) = \text{tr}(\Sigma)/|\Sigma|_2 \) (effectively, use this!)
  - small when single large: \(|z|_2 \) is large.
- If each source bounded away from pure noise, all good.
Users write labeling functions for multiple related tasks.

We model the labeling functions' behavior to denoise them.

We use the probabilistic labels to train a multi-task model.
Recovery Results (Informal)

Result:

• Given $n$ unlabeled data points—that overlap.
• And a sufficiently independent set of LFs (for recovery)
• The end model test set error should decrease as $n^{-1/2}$

\[
E[\| l_{\hat{w}} - l_{w^*} \|] = 0 \left( \frac{1}{\sqrt{n}} \right)
\]

Same asymptotic rate as with labeled data!

NB: Generalization straightforward—if you assume coverage.
## Empirical Results: NLP Experiments

<table>
<thead>
<tr>
<th></th>
<th>Ontonotes (Fine-grained NER)</th>
<th>TACRED (Relation Extraction)</th>
<th>OpenI (Document Classification)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gold Labels (n=300)</strong></td>
<td>63.7 ± 2.1</td>
<td>28.4 ± 2.3</td>
<td>62.7 ± 4.5</td>
<td>51.6</td>
</tr>
<tr>
<td><strong>Majority Vote</strong></td>
<td>76.9 ± 2.6</td>
<td>43.9 ± 2.6</td>
<td>74.2 ± 1.2</td>
<td>65.0</td>
</tr>
<tr>
<td><strong>Pipelined Snorkel</strong></td>
<td>78.4 ± 1.2</td>
<td>49.0 ± 2.7</td>
<td>75.8 ± 0.9</td>
<td>67.7</td>
</tr>
<tr>
<td><strong>Snorkel MeTaL</strong></td>
<td>82.2 ± 0.8</td>
<td>56.7 ± 2.1</td>
<td>76.6 ± 0.4</td>
<td>71.8</td>
</tr>
</tbody>
</table>

- Avg. over Traditionally Supervised: +20 points
- Avg. over Majority Vote: +7 points
- Avg. over Single Task Modeling: +4 points
Changes how you iterate...
Automated Chest X-ray Triage

Optimizing Workflows with Automated Prioritization, Radiology 19

Cross-Modal Chest X-ray Classification

![ROC curve for Cross-Modal Chest X-ray Classification](image)

- True Positive Rate
- False Positive Rate
- Months

- Line: FS, 5K (AUC=0.88)
Cross-Modal Chest X-ray Classification

Days

Years

Months

- FS, 50K (AUC=0.95)
- FS, 5K (AUC=0.88)
Applying Weak Supervision Across Modalities

We can leverage data programming across modalities to make weak supervision of complex tasks easier!
Indication: Chest pain. Findings:
Mediastinal contours are within normal limits. Heart size is within normal limits. No focal consolidation, pneumothorax or pleural effusion. Impression: No acute cardiopulmonary abnormality.

20 Labeling Functions
Indication: Chest pain. Findings: Mediastinal contours are within normal limits. Heart size is within normal limits. No focal consolidation, pneumothorax or pleural effusion. Impression: No acute cardiopulmonary abnormality.
Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging

New Abstractions, New Problems

These eyes haunt me…

Any model may pick out **unintended signal**. Deep models may pick out **more** unintended signal.

**Upshot:**
Picked up on **mascara**

Kuehlkamp et al. *Gender-from-Iris or Gender from-Mascara*

Do we know how well these models are really performing?
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

Are we sure those differences are causal? Anticausal?
- 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><strong>0.91</strong></td>
</tr>
</tbody>
</table>

**Often:** Differences in models ~ 2-3 points.

Later: Label quality & quantity > model choice.
It’s not just those eyes...

Melanoma Recognition (Surgical Marks)

Pneumonia Detection

No Drain

With Drain

Pneumothorax detection 0.87 AUC, which is superhuman

... with chest drains—**Chest drain means already treated!** Down to 0.77 when removed...

Image Credits

One issue: Hidden Stratification.

- Classical: Never write features that say
  - If drain then pneumonia
  - If purple dot then cancer
  - But new SW abstraction, new bugs

- Accidental—not adversarial—attacks. A subset of a class (stratum) performs worse.
  - E.g., Abnormal consists of many unlabeled subclasses or strata.

Develop a theory & techniques to handle hidden stratification?

http://hazyresearch.stanford.edu/hidden-stratification
Related Work
Related Work in Weak Supervision

- **Distant Supervision:** Mintz et. al. 2009, Alfonesca et. al. 2012, Takamatsu et. al. 2012, Roth & Klakow 2013, Augenstein et. al. 2015, etc.
- **Crowdsourcing:** Dawid & Skene 1979, Karger et. al. 2011, Dalvi et. al. 2013, Ruvolo et. al. 2013, Zhang et. al. 2014, Berend & Kontorovich 2014, etc.
- **Co-Training:** Blum & Mitchell 1998
- **Noisy Learning:** Bootkrajang et. al. 2012, Mnih & Hinton 2012, Xiao et. al. 2015, etc.
- **Indirect Supervision:** Clarke et. al. 2010, Guu et. al. et. al. 2017, etc.
- **Feature and Class-distribution Supervision:** Zaidan & Eisner 2008, Druck et. al. 2009, Liang et. al. 2009, Mann & McCallum 2010, etc.
- **Boosting & Ensembling:** Schapire & Freund, Platanios et. al. 2016, etc.
- **Constraint-Based Supervision:** Bilenko et. al. 2004, Koestinger et. al. 2012, Stewart & Ermon 2017, etc.
- **Propensity SVMs:** Joachims 17
More Related work

• So much more! *Work was inspired by classics and new Cotraining, GANs, capsule networks, semi-supervised learning, crowd-sourcing and so much more!*

• Please see blog for summary. [https://www.snorkel.org/blog/weak-supervision](https://www.snorkel.org/blog/weak-supervision)
Future Work
Programming Stack

- Application Interfaces
- Declarative Language
- High-Level Language
- Assembly Language
- Machine Language

Supervision Stack

- LFs Auto-Generated from User Behavior
- LFs Compiled from Natural Language
- LFs Built on Advanced Primitives
- LFs Coded Directly
- Individual Labels

Levels:
- High-level
- Low-level

Publications:
- NeurIPS16, VLDB18
- NeurIPS17&18, VLDB19
- ACL 18
- ICLR 19.
Software 2.0 and Data Programming: Lessons Learned, and What’s Next

Dan Fu, Laurel Orr, and students of HazyResearch

Posted on February 28, 2020

The only view that matters: student & postdoc view

http://Hazyresearch.Stanford.edu
Conclusion

• **Snorkel**: A 1st of its kind software 2.0 system

• **Nugget**: estimate quality without using labeled data

• The change to programming by supervision changes what systems you build and how you build them.

Snorkel.org