Learning Dependency Structures for Weak Supervision Models

Fred Sala, Paroma Varma, Ann He, Alex Ratner, Chris Ré
Snorkel and Weak Supervision

A system for rapidly creating training sets with weak supervision


Frequent use in industry!

Bach et al., “Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale”, SIGMOD (Industrial) 2019.
The Snorkel/Weak Supervision Pipeline

Users write labeling functions to noisily label data.

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

We use the probabilistic labels to train an arbitrary end model.

Requires Dependency Structure!

Takeaway: No hand-labeled training data needed!
Model as Generative Process

Problem: learn the parameters of this model (accuracies & correlations) without $Y$?
Solution Sketch: Using the covariance

\[ \Sigma = \Sigma_0 \]

Can only observe part of the covariance…
Idea: Use graph-sparsity of the inverse

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

Observed overlaps
Low-rank parameters to solve for
Is zero where corresponding pair of variables has no edge

[Loh & Wainwright 2013]

Key: we must know the dependency structure
Idea: Use graph-sparsity of the inverse

\[
(S^{-1})_o = \Sigma_o^{-1} + ZZ^T
\]

Example: 8 LFs
1 triangle, 2 pairs, 1 singleton
Inverse Encodes The Structure…

\[
(\Sigma^{-1})_o = \Sigma^{-1}_o + ZZ^T
\]
But Observed Matrix Doesn’t

$\left( \Sigma^{-1} \right)_o = \Sigma_o^{-1} + ZZ^T$
Need the Sparse Component...

Can we extract the sparse part?

\[
\Sigma^{-1}_O = (\Sigma^{-1})_O - ZZ^T
\]

Observed \quad Sparse \quad Low-Rank
... & Robust PCA Recovers It!

Need to decompose:

\[
\Sigma^{-1}_O = (\Sigma^{-1})_O - ZZ^T
\]

**Observed**  
**Sparse**  
**Low-Rank**

**Robust PCA**: Decompose a matrix into *sparse* and *low-rank* components; *sparse* part contains graph structure

**Convex optimization**:  
\[
\text{argmin}_{S,L} \| \Sigma^{-1}_O - (S + L) \|_2 + \lambda \| S \|_1 + \mu \| L \|_*
\]

Candes et al., “Robust Principal Components Analysis?”, Chandrasekaran et al., “Rank-Sparsity Incoherence for Matrix Decomposition”
Theory Results: Sample Complexity

\( m \) is # of LFs, \( d \) is largest degree for a dependency

- **Prior work**: samples to recover WS dependency structure w. h. p.
  

  \[ \Omega(m \log m) \]

  Doesn’t exploit \( d \): sparsity of the graph structure

- Recent application of RPCA for general latent-variable structure learning


  \[ \Omega(d^2 m) \]

  Linear in \( m \).
Theory Results: Sample Complexity

\( m \) is \# of LFs, \( d \) is largest degree for a dependency

**Ours:** for \( \tau < 1 \), an eigenvalue decay factor in blocks of LFs

\[ \Omega(d^2m^{\tau}) \]

**Ours:** When there is a dominant block of correlated LFs

\[ \Omega(d^2\log m) \]

Idea: exploit sharp concentration inequalities on sample covariance matrix \( \Sigma_0 \) via the *effective rank* [Vershynin ’12]
We pick up all the edges—

+4.64 F1 points, over indep., + 4.13 over Bach et al.
More Resources

• **Blog Post:** Intro to weak supervision
  [https://dawn.cs.stanford.edu/2017/12/01/snorkel-programming/](https://dawn.cs.stanford.edu/2017/12/01/snorkel-programming/)

• **Blog Post:** Gentle Introduction to Structure Learning

• **Software:** [https://github.com/HazyResearch/metal](https://github.com/HazyResearch/metal)

Fred Sala: [https://stanford.edu/~fredsala](https://stanford.edu/~fredsala)