Alternative Specifications in Machine Learning

Percy Liang

Jacob Steinhardt, Fereshte Khani

FLI Workshop

January 5, 2016
Foundations of machine learning

Train $p_0(x)$
Foundations of machine learning

Train $p_0(x)$  
Test $p_1(x)$

Classic statistical learning theory:

training distribution $= \text{test distribution}$
Foundations of machine learning

Train $p_0(x)$

Test $p_1(x)$

Classic statistical learning theory:

training distribution $=$ test distribution

Relaxation: domain adaptation, mild adversaries

training distribution $\approx$ test distribution
Foundations of machine learning

Train $p_0(x)$  
Test $p_1(x)$

Classic statistical learning theory:

training distribution = test distribution

Relaxation: domain adaptation, mild adversaries

training distribution $\approx$ test distribution

Issue:

doesn't address large changes (disasters, adversaries)
Changes and changes
Changes and changes

Long-term risks of AI: unknown unknowns
What’s the right specification?

**Specification: standard machine learning**

**Input:** training data

**Output:** model that does obtains low *expected* test error

Is *expected* test error enough?
What’s the right specification?

**Specification:** standard machine learning

**Input:** training data  
**Output:** model that does obtains low *expected* test error

Is *expected* test error enough?

**Scenario:**

- Err on 1% on instances
- Agents maximize, adversaries minimize, could drive us there!
Specification: selective prediction

Input: training data
Output: model that outputs correct answer or "don’t know"

Previous work: Chow (1970); Tortorella (2000); El-Yaniv & Wiener (2010); Balsubramani (2016)
Unanimous prediction

Assumption: exists mapping with zero error
Unanimous prediction

\[
S = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 \\
\end{bmatrix}
\]

\[
M = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
T = \begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
\end{bmatrix}
\]
Unanimous prediction

Models consistent with training data:

\[ C = \{ M \geq 0 : SM = T \} \]
Unanimous prediction

Models consistent with training data:

\[ \mathcal{C} = \{ M \geq 0 : SM = T \} \]

Challenge:

Checking all consistent \( M \in \mathcal{C} \) is slow...
Fast two point scheme

- Choose $M_1, M_2 \in \mathcal{C}$ randomly enough
- Return "don’t know" iff $M_1$ and $M_2$ disagree

\[ z \leq 0 \]
\[ -z \leq 0 \]
\[ -x \leq 0 \]
\[ -y \leq 0 \]
\[ x + y \leq 6 \]
Experimental results

- GeoQuery semantic parsing dataset (800 train, 280 test)

What is the population of Texas?
New specification 2/2

[NIPS 2016]

**Specification: unsupervised risk estimation**

**Input:** unlabeled examples and model
**Output:** estimate of labeled accuracy

**Previous work:** Donmez et al. (2010); Dawid/Skene (1979); Zhang et al. (2014); Jaffe et al. (2015); Balasubramanian et al. (2011)
Is this possible?

Compute $\mathbb{E}[\text{loss}(x, y; \theta)]$
Assumptions

Conditional independence:

\[ y \quad x_1 \quad x_2 \quad x_3 \]
Assumptions

Conditional independence:

\[ y \quad x_1 \quad x_2 \quad x_3 \]

Loss function decomposes:

\[ A(x; \theta) - f_1(x_1, y; \theta) - f_2(x_2, y; \theta) - f_3(x_3, y; \theta) \]
Assumptions

Conditional independence:

\[ A(x; \theta) - f_1(x_1, y; \theta) - f_2(x_2, y; \theta) - f_3(x_3, y; \theta) \]

only conditional independence structure
Intuition

Three views agree → (probably) low error
Intuition

Three views disagree $\rightarrow$ high error
Tensor factorization

\( y \) \( x_1 \) \( x_2 \) \( x_3 \)

\((k\) labels, views \(v = 1, 2, 3\))

\[
\begin{bmatrix}
  f_v(x, 1) \\
  \ldots \\
  f_v(x, k)
\end{bmatrix}
\]

[Anandkumar et al., 2013]
Tensor factorization

\[ y \]

\[ x_1 \quad x_2 \quad x_3 \]

\((k\) labels, views \(v = 1, 2, 3)\)

\[ M_v = \begin{bmatrix}
\mathbb{E}[f_v(x, 1) \mid y = 1] & \cdots & \mathbb{E}[f_v(x, 1) \mid y = k] \\
\cdots & \cdots & \cdots \\
\mathbb{E}[f_v(x, k) \mid y = 1] & \cdots & \mathbb{E}[f_v(x, k) \mid y = k]
\end{bmatrix} \]

[Anandkumar et al., 2013]
Tensor factorization

- Observe $\mathbb{E}[f_1(x, a)f_2(x, b)]$

(k labels, views $\nu = 1, 2, 3$)

[Anandkumar et al., 2013]
Tensor factorization

\[ y \]

\( x_1 \quad x_2 \quad x_3 \)

(k labels, views \( v = 1, 2, 3 \))

- Observe \( \mathbb{E}[f_1(x, a)f_2(x, b)f_3(x, c)] \)

[Anandkumar et al., 2013]
Tensor factorization

- Observe $\mathbb{E}[f_1(x, a) f_2(x, b) f_3(x, c)]$
- Perform tensor factorization to obtain
  \[ M_{vba} = \mathbb{E}[f_v(x, b) \mid y = a] \]
Tensor factorization

\[ M_{vba} = \mathbb{E}[f_v(x, b) | y = a] \]

- Observe \( \mathbb{E}[f_1(x, a)f_2(x, b)f_3(x, c)] \)
- Perform tensor factorization to obtain
- Use to compute risk (up to label permutation)

\[ \mathbb{E}[A(x; \theta) - f_1(x_1, y; \theta) - f_2(x_2, y; \theta) - f_3(x_3, y; \theta)] \]
Results
Discussion

- Maximize expected accuracy ⇒ selective prediction, unsupervised risk estimation

- Key question: Can we weaken the assumptions?
Code and data

worksheets.codalab.org

Collaborators

Fereshte Khani
Jacob Steinhardt

Thank you!