Machine Learning using Descriptive Complexity and Propositional Solvers

Skip Jordan and Łukasz Kaiser

ERATO Group, JST & Hokkaido University
LIAFA, CNRS & Université Paris Diderot

Highlights ’13
Why Logic in Machine Learning?

What is machine learning?

Given \((x^{(i)}, y^{(i)})\)_{i=1...m} find \(h \in H\) such that \(h(x^{(i)}) \approx y^{(i)}\)
Why Logic in Machine Learning?

What is machine learning?

Given \((x^{(i)}, y^{(i)})_{i=1 \ldots m}\) find \(h \in H\) such that \(h(x^{(i)}) \approx y^{(i)}\)

Issues

• Is \(H\) abstract (e.g. PTIME) or concrete (e.g. \(\{\theta \cdot \text{input} \mid \theta\}\))?
• Can we get efficient algorithms and theoretical guarantees?
Why Logic in Machine Learning?

What is machine learning?

Given \((x^{(i)}, y^{(i)})_{i=1}^{m}\) find \(h \in H\) such that \(h(x^{(i)}) \approx y^{(i)}\)

Issues

• Is \(H\) abstract (e.g. PTIME) or concrete (e.g. \(\{\theta \cdot \text{input} \mid \theta\}\))?  
• Can we get efficient algorithms and theoretical guarantees?

How can we use logic?

• Idea: \(H\) are parametrized formulas
• Descriptive complexity gives theoretical guarantees
• Propositional solvers used for efficient learning
Learning Board Game Rules
Representing Board Games

\[ \exists x_1 \ldots x_5 \left( \bigwedge_{1 \leq i \leq 5} G(x_i) \wedge \left( \bigwedge_{1 \leq i \leq 5} R(x_i, x_{i+1}) \vee \bigwedge_{1 \leq i \leq 5} C(x_i, x_{i+1}) \right) \vee \bigwedge_{1 \leq i \leq 5} \exists y(R(x_i, y) \wedge C(y, x_{i+1})) \vee \bigwedge_{1 \leq i \leq 5} \exists y(R(x_i, y) \wedge C(x_{i+1}, y)) \right) \]
Learning Winning Conditions

Positive Example $\mathcal{A}$

Find minimal $\varphi$ such that $\mathcal{A} \models \varphi$, $\mathcal{B} \models \neg \varphi$
Learning Winning Conditions

Positive Example $\mathcal{A}$

Negative Example $\mathcal{B}$

Find minimal $\varphi$ such that $\mathcal{A} \models \varphi$, $\mathcal{B} \models \neg \varphi$

Which logic and minimality?

- Full FO, minimal quantifier rank: PSPACE-complete (Pezzoli ’98)
- $\text{FO}^k + \text{C}$, minimal quantifier rank: PTIME (Grohe ’99)
- $k = 16$ and $\log(n)$ quantifiers suffice for … (Pikhurko, Verbitsky ’10)
Learning Winning Conditions

Positive Example $\mathcal{A}$

Negative Example $\mathcal{B}$

Find minimal $\varphi$ such that $\mathcal{A} \models \varphi$, $\mathcal{B} \models \neg \varphi$

Which logic and minimality?

- Full FO, minimal quantifier rank: PSPACE-complete (Pezzoli '98)
- FO$^k$+$C$, minimal quantifier rank: PTIME (Grohe '99)
- $k = 16$ and $\log(n)$ quantifiers suffice for ... (Pikhurko, Verbitsky '10)

Extensions: TC$^m$ and guarded formulas, greedy shortening, ...

computed formula: $\exists x (W(x) \land \forall y \neg C(x, y))$
Learning Winning Conditions

Positive Example $\mathcal{A}$

Negative Example $\mathcal{B}$

Find minimal $\varphi$ such that $\mathcal{A} \models \varphi$, $\mathcal{B} \models \neg \varphi$

Which logic and minimality?

- Full FO, minimal quantifier rank: PSPACE-complete (Pezzoli ’98)
- FO$^k$+C, minimal quantifier rank: PTIME (Grohe ’99)
- $k = 16$ and $\log(n)$ quantifiers suffice for … (Pikhurko, Verbitsky ’10)

Extensions: TC$^m$ and guarded formulas, greedy shortening, …

computed formula: $\exists x \ (W(x) \land \forall y \neg C(x, y))$

Learning game rules from videos (K., AAAI-12)

http://toss.sf.net/learn.html
Learning Reductions
Formula Outlines

Conjunction outline (conjunction with Boolean guards on all atoms)

\[ X_1E(x_1, x_1) \land X_2E(x_1, x_2) \land X_3E(x_2, x_1) \land X_4E(x_2, x_2) \land X_5\neg E(x_1, x_1) \land X_6\neg E(x_1, x_2) \land X_7\neg E(x_2, x_1) \land X_8\neg E(x_2, x_2) \]
Formula Outlines

Conjunction outline (conjunction with Boolean guards on all atoms)

\[ X_1 E(x_1, x_1) \land X_2 E(x_1, x_2) \land X_3 E(x_2, x_1) \land X_4 E(x_2, x_2) \land X_5 \neg E(x_1, x_1) \land X_6 \neg E(x_1, x_2) \land X_7 \neg E(x_2, x_1) \land X_8 \neg E(x_2, x_2) \]

l-DNF outline (all quantifier-free formulas)

\[ C_1 \lor C_2 \lor \cdots \lor C_l \]
Formula Outlines

Conjunction outline (conjunction with Boolean guards on all atoms)

\[ X_1 E(x_1, x_1) \land X_2 E(x_1, x_2) \land X_3 E(x_2, x_1) \land X_4 E(x_2, x_2) \land \\
X_5 \neg E(x_1, x_1) \land X_6 \neg E(x_1, x_2) \land X_7 \neg E(x_2, x_1) \land X_8 \neg E(x_2, x_2) \]

\(/-\)DNF outline (all quantifier-free formulas)

\[ C_1 \lor C_2 \lor \cdots \lor C_l \]

Extensions

• *k*-Variable \( \exists /-\)DNF outline

\[ \exists x_1 \ldots x_k (C_1 \lor C_2 \lor \cdots \lor C_l) \]
Formula Outlines

Conjunction outline (conjunction with Boolean guards on all atoms)

\[ X_1 E(x_1, x_1) \land X_2 E(x_1, x_2) \land X_3 E(x_2, x_1) \land X_4 E(x_2, x_2) \land X_5 \neg E(x_1, x_1) \land X_6 \neg E(x_1, x_2) \land X_7 \neg E(x_2, x_1) \land X_8 \neg E(x_2, x_2) \]

/-DNF outline (all quantifier-free formulas)

\[ C_1 \lor C_2 \lor \cdots \lor C_l \]

Extensions

- **k-Variable ∃ /-DNF outline**

  \[ \exists x_1 \ldots x_k \ (C_1 \lor C_2 \lor \cdots \lor C_l) \]

- **m-Predicate k-Variable ∃ /-DNF outline**

  \[ P_i(x) = \exists x_1 \ldots x_k \ (C_1 \lor C_2 \lor \cdots \lor C_l), \quad i = 1 \ldots m \]
Formula Outlines

Conjunction outline (conjunction with Boolean guards on all atoms)

\[ X_1 E(x_1, x_1) \land X_2 E(x_1, x_2) \land X_3 E(x_2, x_1) \land X_4 E(x_2, x_2) \land X_5 \neg E(x_1, x_1) \land X_6 \neg E(x_1, x_2) \land X_7 \neg E(x_2, x_1) \land X_8 \neg E(x_2, x_2) \]

/DNF outline (all quantifier-free formulas)

\[ C_1 \lor C_2 \lor \cdots \lor C_l \]

Extensions

• \( k \)-Variable \( \exists / \)-DNF outline

\[ \exists x_1 \ldots x_k (C_1 \lor C_2 \lor \cdots \lor C_l) \]

• \( m \)-Predicate \( k \)-Variable \( \exists / \)-DNF outline

\[ P_i(x) = \exists x_1 \ldots x_k (C_1 \lor C_2 \lor \cdots \lor C_l), \quad i = 1 \ldots m \]

• \( n \)-Layer \( m \)-Predicate \( k \)-Variable \( \exists / \)-DNF outline

\[ P^1_i(x) = \ldots (\text{atoms}); \quad P^2_i(x) = \ldots (P^1s); \quad \ldots \quad P^n_i(x) = \ldots (P^{n-1}s) \]
Automatic Reduction Finding

Representing reductions by $k$-dimensional quantifier-free queries

\[ (k = 2, \varphi_U = T, \psi_E(x_1, x_2, y_1, y_2) = E(x_1, y_1) \land (x_2 = y_2 \lor y_2 = s)) \]

\[ s, \circ \sim \circ, s \]
Automatic Reduction Finding

Representing reductions by $k$-dimensional quantifier-free queries

\( (k = 2, \varphi_U = T, \psi_E(x_1, x_2, y_1, y_2) = E(x_1, y_1) \land (x_2 = y_2 \lor y_2 = s)) \)

Finding reductions by CEGAR and SAT-solvers

- Find a $l$-DNF reduction $\theta_i$ good on counter-examples $\mathcal{E}_0, \ldots, \mathcal{E}_i$
- Find a counter-example $\mathcal{E}_{i+1}$ to $\theta_i$, iterate

(Jordan, K., SAT ’13 improving on Crouch, Immerman, Moss ’10)
Automatic Reduction Finding

Representing reductions by \(k\)-dimensional quantifier-free queries
\((k = 2, \varphi_U = \top, \psi_E(x_1, x_2, y_1, y_2) = E(x_1, y_1) \land (x_2 = y_2 \lor y_2 = s))\)

Finding reductions by CEGAR and SAT-solvers
- Find a \(l\)-DNF reduction \(\theta_i\) good on counter-examples \(\mathcal{E}_0, \ldots, \mathcal{E}_i\)
- Find a counter-example \(\mathcal{E}_{i+1}\) to \(\theta_i\), iterate

(Jordan, K., SAT ’13 improving on Crouch, Immerman, Moss ’10)

Easy example: s-t reachability to strongly connected (both NL-complete)

\[
\text{Reach} = \left[ \text{tc}_{x,y} E(x, y) \right](.s, .t) \quad \text{SC} := \forall x, y (\text{tc}_{x,y} E(x, y))
\]
Automatic Reduction Finding

Representing reductions by $k$-dimensional quantifier-free queries

$$ (k = 2, \ \varphi_U = \top, \ \psi_E(x_1, x_2, y_1, y_2) = E(x_1, y_1) \land (x_2 = y_2 \lor y_2 = s)) $$

Finding reductions by CEGAR and SAT-solvers

- Find a $l$-DNF reduction $\theta_i$ good on counter-examples $\mathcal{E}_0, \ldots, \mathcal{E}_i$
- Find a counter-example $\mathcal{E}_{i+1}$ to $\theta_i$, iterate

(Jordan, K., SAT ’13 improving on Crouch, Immerman, Moss ’10)

Easy example: s-t reachability to strongly connected (both NL-complete)

$$ \text{Reach} = \left[ \text{tc}_{x,y} \ E(x, y) \right](s, t) \quad \text{SC} := \forall x, y(\text{tc}_{x,y} \ E(x, y)) $$

http://toss.sf.net/reduct.html
Automatic Reduction Finding

Representing reductions by $k$-dimensional quantifier-free queries

\[ (k = 2, \ \varphi_U = \top, \ \psi_E(x_1, x_2, y_1, y_2) = E(x_1, y_1) \land (x_2 = y_2 \lor y_2 = s)) \]

Finding reductions by CEGAR and SAT-solvers

- Find a $I$-DNF reduction $\theta_i$ good on counter-examples $E_0, \ldots, E_i$
- Find a counter-example $E_{i+1}$ to $\theta_i$, iterate

(Jordan, K., SAT ’13 improving on Crouch, Immerman, Moss ’10)

Easy example: $s$-$t$ reachability to strongly connected (both NL-complete)

\[
\text{Reach} = \left[ t_{c_x,y} \ E(x, y) \right] (.s, .t) \quad \text{SC} := \forall x, y (t_{c_x,y} \ E(x, y))
\]

\[ (k = 1, \ \varphi_U = \top, \ \psi_E = x_1 = s \lor x_2 = t \lor E(x_2, x_1)) \]
Automatic Reduction Finding

Representing reductions by $k$-dimensional quantifier-free queries

\[(k = 2, \varphi_U = \top, \psi_E(x_1, x_2, y_1, y_2) = E(x_1, y_1) \land (x_2 = y_2 \lor y_2 = s))\]

Finding reductions by CEGAR and SAT-solvers

- Find a $l$-DNF reduction $\theta_i$ good on counter-examples $\mathcal{E}_0, \ldots, \mathcal{E}_i$
- Find a counter-example $\mathcal{E}_{i+1}$ to $\theta_i$, iterate

(Jordan, K., SAT ’13 improving on Crouch, Immerman, Moss ’10)

Easy example: s-t reachability to strongly connected (both NL-complete)

\[\text{Reach} = [\text{tc}_{x,y} E(x, y)](.s,.t) \quad \text{SC} := \forall x, y (\text{tc}_{x,y} E(x, y))\]

\[(k = 1, \varphi_U = \top, \psi_E = x_1 = s \lor x_2 = t \lor E(x_2, x_1))\]

Other applications: game rule learning, program synthesis, …
Looking Forward
Machine Learning Motivation

It’s hard to prove anything about deep learning systems
Y. LeCun, COLT ’13
What are Convolutional Networks?

**Neuron**: \( \sigma(\text{weighted sum}) \)

**Network with shared weights**

Example (LeNet-five.fitted, zero.fitted, nine.fitted) MNIST error rate

(credit: EBLearn (eblearn.cs.nyu.edu))
What are Convolutional Networks?

**Neuron:** $\sigma(\text{weighted sum})$

**Network** with shared weights

Example (LeNet-5, 0.95% MNIST error rate)

(credit: EBLearn (eblearn.cs.nyu.edu))
Threshold Convolutional Formulas

Convolution

\[ P^1(x) = \exists y, z, v. (R(x, y) \land C(x, z) \land R(z, v)) \varphi \]

with \( \varphi = w_1 \chi[B(x)] + \cdots + w_4 \chi[B(v)] \geq t \)

Subsampling (max-pooling)

\[ P^2(x) = \exists y, z, v. (R_2(x, y) \land C_2(x, z) \land \ldots) \psi \]

where \( C_2(x, y) = \exists z(C(x, z) \land C(z, y)) \)

and \( \psi = w_1 \chi[P^1_1(x)] + \cdots + w_4 \chi[P^1_m(v)] \geq t' \)
Threshold Convolutional Formulas

Convolution

\[ P^1(x) = \exists y, z, v. (R(x, y) \land C(x, z) \land R(z, v)) \varphi \]

with \( \varphi = w_1 \chi[B(x)] + \cdots + w_4 \chi[B(v)] \geq t \)

Subsampling (max-pooling)

\[ P^2(x) = \exists y, z, v. (R_2(x, y) \land C_2(x, z) \land \cdots) \psi \]

where \( C_2(x, y) = \exists z (C(x, z) \land C(z, y)) \)

and \( \psi = w_1 \chi[P^1_1(x)] + \cdots + w_4 \chi[P^1_m(v)] \geq t' \)

= n-Layer \( m \)-Predicate \((k \times k)\)-Variable \( \exists \) guarded threshold outline

\[ P^1_i(x) = \ldots (atoms); \ P^2_i(x) = \ldots (P^1_i s); \ldots \ P^n_i(x) = \ldots (P^{n-1}_i s) \]
Threshold Convolutional Formulas

**Convolution**

\[ P_1^1(x) = \exists y, z, v. (R(x, y) \land C(x, z) \land R(z, v)) \varphi \]

with \( \varphi = w_1 \chi[B(x)] + \cdots + w_4 \chi[B(v)] \geq t \)

**Subsampling (max-pooling)**

\[ P_2^2(x) = \exists y, z, v. (R_2(x, y) \land C_2(x, z) \land \ldots) \psi \]

where \( C_2(x, y) = \exists z(C(x, z) \land C(z, y)) \)

and \( \psi = w_1 \chi[P_1^1(x)] + \cdots + w_4 \chi[P_m^1(v)] \geq t' \)

\[ = n\text{-Layer } m\text{-Predicate } (k \times k)\text{-Variable } \exists \text{ guarded threshold outline} \]

\[ P_1^1(x) = \ldots (\text{atoms}); \ P_2^2(x) = \ldots (P_1^1s); \ldots P_i^n(x) = \ldots (P_i^{n-1}s) \]

**Inspiring goal:** uniform learning platform and theory
Threshold Convolutional Formulas

Convolution

\[ P^1(x) = \exists y, z, v. (R(x, y) \land C(x, z) \land R(z, v)) \varphi \]

with \( \varphi = w_1 \chi[B(x)] + \cdots + w_4 \chi[B(v)] \geq t \)

Subsampling (max-pooling)

\[ P^2(x) = \exists y, z, v. (R_2(x, y) \land C_2(x, z) \land \cdots) \psi \]

where \( C_2(x, y) = \exists z (C(x, z) \land C(z, y)) \)

and \( \psi = w_1 \chi[P^1_1(x)] + \cdots + w_4 \chi[P^1_m(v)] \geq t' \)

= n-Layer m-Predicate \((k \times k)\)-Variable \( \exists \) guarded threshold outline

\[ P^1_i(x) = \ldots \text{(atoms)}; \; P^2_i(x) = \ldots (P^1_s); \; \ldots \; P^n_i(x) = \ldots (P^{n-1}_s) \]

Inspiring goal: uniform learning platform and theory