SERGEI OBIEDKOV

LEARNING HORN FORMULAS WITH QUERIES

SUPERVISED LEARNING

- **Input:** a training set divided into (for example) two classes w.r.t. a certain target property.
  - positive examples
  - negative examples
- **Build** a classifier that determines whether a previously unseen object has the target property.

SUPERVISED LEARNING

- **Input:** a training set divided into (for example) two classes w.r.t. a certain target property.
  - positive examples
  - negative examples
- **Build** a classifier that determines whether a previously unseen object has the target property.

an image of a pipe

not an image of a pipe

LEARNING WITH QUERIES (ANGLUIN 1988)

- **Input:** an oracle capable of answering queries of certain predefined types concerning a target property.
- **Build** a classifier that determines whether a previously unseen object has the target property.

TYPES OF QUERIES

- **Membership query:** Does the object have the target property?

TYPES OF QUERIES

- **Membership query:** Does the object have the target property?

Is this a dog?

Yes!

No!

TYPES OF QUERIES

- **Membership query:** Does the object have the target property?
- **Equivalence query:** Does the hypothesis *H* accurately describe the set of objects with the target property?

TYPES OF QUERIES

- **Membership query:** Does the object have the target property?
- **Equivalence query:** Does the hypothesis *H* accurately describe the set of objects with the target property? If not, the oracle must provide
  - a positive counterexample that has the target property, but is not covered by the hypothesis

TYPES OF QUERIES

- **Membership query:** Does the object have the target property?
- **Equivalence query:** Does the hypothesis *H* accurately describe the set of objects with the target property? If not, the oracle must provide
  - a positive counterexample that has the target property, but is not covered by the hypothesis

or

  - a negative counterexample that doesn't have the property, but satisfies the hypothesis.

TYPES OF QUERIES

- **Equivalence query:** Does the hypothesis H accurately describe the set of objects with the target property?

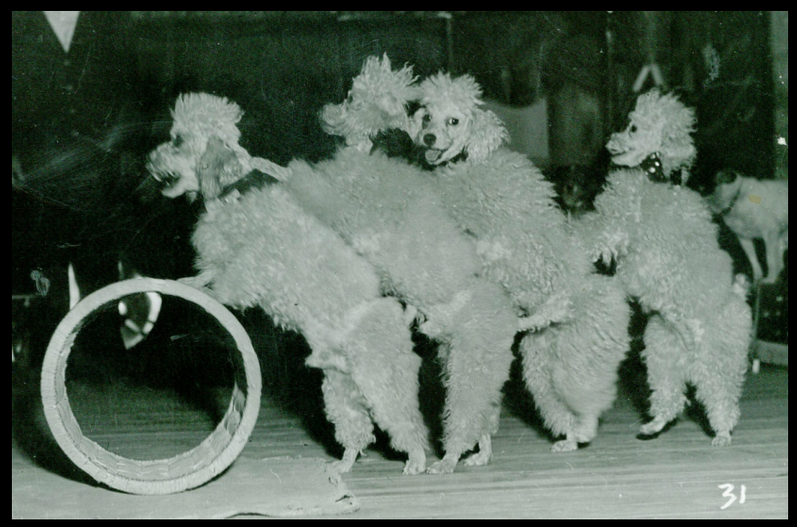
Are dogs animals with four legs and curly hair?

TYPES OF QUERIES

- **Equivalence query:** Does the hypothesis H accurately describe the set of objects with the target property?

Are dogs animals with four legs and curly hair?

Well, some of the are…



TYPES OF QUERIES

- **Equivalence query:** Does the hypothesis H accurately describe the set of objects with the target property?

Are dogs animals with four legs and curly hair?

…but no!

positive counterexample

TYPES OF QUERIES

- **Equivalence query:** Does the hypothesis H accurately describe the set of objects with the target property?

Are dogs animals with four legs and curly hair?

…but no!

positive counterexample

negative counterexample

TYPES OF QUERIES

TYPES OF QUERIES

- **Subset query:** Does the hypothesis *H* describe only objects with the target property?
  - If not, provide a negative counterexample.

TYPES OF QUERIES

- **Subset query:** Does the hypothesis *H* describe only objects with the target property?
  - If not, provide a negative counterexample.
- **Superset query:** Does the hypothesis *H* describe all the objects with the target property?
  - If not, provide a positive counterexample.

WHAT CAN BE LEARNT WITH QUERIES?

- Deterministic finite automata
- Description logic theories
- Restricted first-order Horn theories
- Preferences
- The structure of social networks

LOOPLESS DIRECTED GRAPHS

VARIABLES

Strongly Connected		Rooted
Weakly Connected	Tournament	Acyclic
Disconnected		Transitive

We want to build the implication theory of loopless directed graphs w.r.t. these seven variables.
It will include theorems such as

ACYCLIC TOURNAMENTS ARE TRANSITIVE.

STRONGLY CONNECTED GRAPHS ARE ROOTED.

HORN FORMULAS

- Horn clause is a disjunction of literals with at most one unnegated variable.
- Horn formula is a conjunction of Horn clauses.
- Example: $(\neg c \vee \neg d \vee a) \quad \wedge \quad (\neg a \vee c) \quad \wedge \quad (\neg a \vee d) \quad \wedge \quad (\neg a \vee \neg b \vee \neg c)$

HORN FORMULAS

- Horn clause is a disjunction of literals with at most one unnegated variable.
- Horn formula is a conjunction of Horn clauses.

- Example: $(\neg c \vee \neg d \vee a) \quad \wedge \quad (\neg a \vee c) \quad \wedge \quad (\neg a \vee d) \quad \wedge \quad (\neg a \vee \neg b \vee \neg c)$
- As a conjunction of implications:

$$(c \wedge d \rightarrow a) \quad \wedge \quad (a \rightarrow c) \quad \wedge \quad (a \rightarrow d) \quad \wedge \quad (a \wedge b \wedge c \rightarrow \bot)$$

HORN FORMULAS

- Horn clause is a disjunction of literals with at most one unnegated variable.
- Horn formula is a conjunction of Horn clauses.

- Example: $(\neg c \vee \neg d \vee a) \quad \wedge \quad (\neg a \vee c) \quad \wedge \quad (\neg a \vee d) \quad \wedge \quad (\neg a \vee \neg b \vee \neg c)$
- As a conjunction of implications:

$$(c \wedge d \to a) \quad \wedge \quad (a \to c) \quad \wedge \quad (a \to d) \quad \wedge \quad (a \wedge b \wedge c \to \bot)$$

- or:

$$\{c, d\} \to \{a\} \qquad \{a\} \to \{c, d\} \qquad \{a, b, c\} \to \bot$$

LEARNING IMPLICATIONS WITH QUERIES

- A set of implications over Φ describes a set of models—assignments that satisfy these implications.

- We identify assignments with the sets of variables they set to 1.

LEARNING IMPLICATIONS WITH QUERIES

- A set of implications over Φ describes a set of models—assignments that satisfy these implications.

- We identify assignments with the sets of variables they set to 1.

Membership queries: Is $A \subseteq M$ a model of H^*?

LEARNING IMPLICATIONS WITH QUERIES

- A set of implications over Φ describes a set of models—assignments that satisfy these implications.

- We identify assignments with the sets of variables they set to 1.

Membership queries: Is $A \subseteq M$ a model of H^*?

Equivalence queries: Is an implication set H equivalent to H^*?

- If not, the oracle must provide a counterexample: a model of H^*, but not of H (positive counterexample), or vice versa (negative counterexample).

POLYNOMIAL-TIME ALGORITHM

(Angluin et al. 1992)

- Computes the representation of the target Horn formula with the smallest number of clauses.
- Makes $O(m^2n)$ membership and $O(mn)$ equivalence queries
  - m is the size of the number of clauses in the computed formula
  - n is the number of variables

WHEN DO WE NEED QUERIES?

- Little data, but we can talk to domain experts.
- Too much data, but we can query it efficiently.
- An abstract domain, but we have automatic procedures for theorem proving and counterexample generation.

EQUIVALENCE QUERIES FOR LEARNING IMPLICATIONS

- Positive counterexamples are (relatively) easy.
- Negative counterexamples are hard.

MEMBERSHIP QUERIES FOR LEARNING IMPLICATIONS

Is C a model of H^*?

- Horn domain (closed under intersection):
  - Check if C is a model.
- Non-Horn domains:
  - Check if C is an intersection of some models.

Angluin's algorithm asks queries about the set of all models of the implications of the domain, not about the domain itself.

HORN ENVELOPE

Horn envelope of formula ϕ is a set of implications that

- includes only
- and from which logically follow all

implications that logically follow from ϕ.

H is a Horn envelope of ϕ if and only if its models are the intersections of the models of ϕ.

We want to be able to generate Horn envelopes of arbitrary formulas using “realistic” queries.

ATTRIBUTE EXPLORATION

- Use queries of the form "Does *A* imply *B*?" (with positive counterexamples).
- Can take exponentially long.

CLOSE ENOUGH IS GOOD ENOUGH

Let H^* be a Horn envelope of ϕ. Then H is an ε-*Horn approximation* of ϕ if

$$\frac{|\operatorname{Mod} H \triangle \operatorname{Mod} H^*|}{2^{|\Phi|}} \leq \varepsilon$$

CLOSE ENOUGH IS GOOD ENOUGH

Let H^* be a Horn envelope of ϕ. Then H is an ε-*Horn approximation* of ϕ if

$$\frac{|\operatorname{Mod} H \triangle \operatorname{Mod} H^*|}{2^{|\Phi|}} \leq \varepsilon$$

Too easy to achieve if

$$|\operatorname{Mod} H^*| \ll 2^{|\Phi|}$$

CLOSE ENOUGH IS GOOD ENOUGH

Let H^* be a Horn envelope of ϕ. Then H is an *ε-Horn approximation* of ϕ if

$$\frac{|\operatorname{Mod} H \triangle \operatorname{Mod} H^*|}{2^{|\Phi|}} \leq \varepsilon$$

Too easy to achieve if

$$|\operatorname{Mod} H^*| \ll 2^{|\Phi|}$$

Let H^* be a Horn envelope of ϕ. Then H is an *ε-strong Horn approximation* of ϕ if

$$\frac{|\{V \subseteq \Phi \mid H(V) \neq H^*(V)\}|}{2^{|\Phi|}} \leq \varepsilon$$

OFTEN ENOUGH CLOSE ENOUGH IS GOOD ENOUGH (PAC LEARNING)

Goal

For ε and δ, use the implication oracle for ϕ to compute H such that

$$\Pr(H \text{ is an } \varepsilon\text{-(strong) Horn approximation of } \phi) \geq 1 - \delta.$$

OFTEN ENOUGH CLOSE ENOUGH IS GOOD ENOUGH

Goal

For ε and δ, use the implication oracle for ϕ to compute H such that

$$\Pr(H \text{ is an } \varepsilon\text{-(strong) Horn approximation of } \phi) \geq 1 - \delta.$$

Solution

In Angluin's algorithm:

- simulate membership queries w.r.t. the Horn envelope of ϕ by implication queries w.r.t. ϕ;
- replace equivalence queries by sampling.

SIMULATING QUERIES

Membership queries

- Use the implication oracle to implement the membership oracle:

$$A \subseteq \Phi \text{ is a model of } H^* \iff A \to \{a\} \text{ for no } a \in \Phi \setminus A$$

- Use counterexamples provided by the oracle as additional positive counterexamples.

Equivalence queries

Replace the exact equivalence oracle by the *sampling oracle*:

- Consider randomly chosen sets $X \subseteq \Phi$
- If $X \in \operatorname{Mod} \mathcal{H} \mathbin{\triangle} \operatorname{Mod} H^*$, return X.
- If after sufficiently many iterations no counterexample has been found, declare H and H^* "`equivalent`".

HOW MANY IS SUFFICIENTLY MANY?

If in the ith call to the sampling equivalence oracle at least

$$\left\lceil \frac{1}{\varepsilon} \cdot \left(i + \log_2 \left(\frac{1}{\delta} \right) \right) \right\rceil$$

randomly chosen sets X are considered, then, with probability at least $1 - \delta$, the result will be an ε-Horn approximation.

MANY POSSIBLE EXTENSIONS

- Background knowledge
- Exceptions
- Symmetries
- Incompletely specified examples
- First-order rule exploration
- Exploration for description logics
- Learning from imperfect oracles
- Learning association rules
- Collaborative exploration

FINALLY, LOOPLESS DIRECTED GRAPHS

- Rooted graphs are weakly connected.
- Tournaments are rooted.
- Transitive graphs are acyclic.
- Disconnected graphs are transitive.
- And so are acyclic tournaments.
- No graph is both weakly connected and disconnected.
- Strongly connected graphs are rooted.
- Strongly connected acyclic graphs are tournaments.

FINALLY, LOOPLESS DIRECTED GRAPHS

- Rooted graphs are weakly connected.
- Tournaments are rooted.
- Transitive graphs are acyclic.
- Disconnected graphs are transitive.
- And so are acyclic tournaments.
- No graph is both weakly connected and disconnected.
- Strongly connected graphs are rooted.
- Strongly connected acyclic graphs are tournaments.