universität innsbruck

Machine Learning for Theorem Proving

Lecture 2 (VU)

Cezary Kaliszyk

Overview

Last Lecture

- Theorem Proving Overview
- Proof Assistants
- Comparison with other tools

Today

- Mini-rehearsal on machine learning
- Machine Learning problems in proving
- Lemma selection
- Statistical methods

Summary of Last Lecture

- Proofs in mathematics and computer are getting more complex
  - some of them are beyond humans to verify
  - rare in mathematics, but more common for large software
- This increases the attractivity of proof assistants. Definition:
  - a computer program to assist a mathematician
    - (how: keep track of theories, definitions, assumptions, check individual steps, provide decision procedures)
  - in the production of a proof
  - that is mechanically checked
    - (means: completely verifiable in a formal logical system)
- Human work: translate human proof accurately and fill in gaps
  - Automation/ATPs help!
- Can machine learning help in this process?

Homework

Find Common Formats for Theorem Proving Problems

- What are their specifics?

Homework

Find Common Formats for Theorem Proving Problems

- What are their specifics?

Specify one LICS exercise in TPTP

- Can an automated prover solve it?
- Compare the number of useful/unused inferences/proof size

Homework

Find Common Formats for Theorem Proving Problems

- What are their specifics?

Specify one LICS exercise in TPTP

- Can an automated prover solve it?
- Compare the number of useful/unused inferences/proof size

Specify it in an interactive theorem prover

- Can you perform any natural-deduction like inference step?

Recently we had a fast progress in machine learning

What is Machine Learning?

Recently we had a fast progress in machine learning

What is Machine Learning?

- Tuning a large number of parameters

Algorithms that improve their performance based on data

Recently we had a fast progress in machine learning

What is Machine Learning?

- Tuning a large number of parameters

Algorithms that improve their performance based on data

- Face detection
- Recommender systems
- Speech recognition
- Stock prediction
- Spam detection
- Molecule modeling
- Automated translation
- LLMs ...

Exercise

Given some values of a function, find the function

Exercise

Given some values of a function, find the function

Given some colored points, how to classify remaining points on the plane?

Exercise

Given some values of a function, find the function

Given some colored points, how to classify remaining points on the plane?

`https://scikit-learn.org/0.15/_images/plot_classifier_comparison_0011.png`

Exercise

Given some values of a function, find the function

Given some colored points, how to classify remaining points on the plane?

`https://scikit-learn.org/0.15/_images/plot_classifier_comparison_0011.png`

ML Tasks

- Classification
- Regression
- Clustering
- Density Estimation
- Dimensionality Reduction

Tasks related to proofs and reasoning

Tasks involving logical inference

- Natural language question answering *[Sukhbaatar+2015]*
- Knowledge base completion *[Socher+2013]*
- Automated translation *[Wu+2016]*

Games

AlphaGo (Zero) problems similar to proving *[Silver+2016]*

- Node evaluation
- Policy decisions

Problems

Main machine learning problems

- Regression: estimate a parameter (1 function)
- Classification: estimate a class (point in plane, digit)
- Clustering

Correspond to

- Supervised learning
- Unsupervised learning
- ...

AI theorem proving techniques (1/3)

High-level AI guidance

- Problems where machine learning helps predict and suggest actions that a human selects
- First such problem is **premise selection**: given a statement that we are trying to prove and a very large library of facts, find the parts of the knowledge that are most useful. For example if we have a fact that looks like topology, a human can find books on topology in a library. A machine learning algorithm can process thousands of books and facts in them - can it select the most useful ones for our goal?
- Second such problem is **tactic selection**: again given a conjecture we have a number of known proof techniques. Predicting that a particular statement should be proved by induction, or for example by integration by parts is a task that a machine learning system could be trained on.
- For both these tasks we need a sufficiently good **characterization** of the goals, proof states, lemmas etc. As such it is necessary to find suitable features and learning techniques. And gather sufficient learning **data**.
- We often have 5–10 seconds to perform the actual prediction,
with more the human will likely do better.

AI theorem proving techniques (2/3)

Mid-level AI guidance

- Problems where the learning can select a strategy for an automated prover, for a tactic, choose a heuristic
- Selecting lemmas that can be reused, proposing intermediate lemmas
- Suggesting new conjectures
- We often have 0.1–3 seconds to perform the actual prediction, with more time running a tool for longer will likely do better human will likely do better.

AI theorem proving techniques (3/3)

Low-level AI guidance

- AI can also be used to directly guide the actions of an automated theorem prover
- This means selecting (almost) every inference step by previous knowledge
- Depending on a calculus this may mean hundreds to tens of thousands of inferences per second
- In some ATPs this may be the selection of a clause to expand, in some ATPs this may be the selection of substitutions to apply, in some cases a branch etc.
- In order to make very fast decisions it is necessary to have very good proof-state characterizations and fast relevance: typically minimal statistical methods
- Most recently also decision tree -based predictions, despite slowing the prover down 2-5 times can allow having overall higher performance

All Problems for Machine Learning

- Is my conjecture true?

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: “Is $a^n + b^n = c^n$ true?” is a tough one...

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: “Is $a^n + b^n = c^n$ true?” is a tough one...
- Is a statement is useful?

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: “Is $a^n + b^n = c^n$ true?” is a tough one…
- Is a statement is useful?
  - Optionally: For a conjecture

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: “Is $a^n + b^n = c^n$ true?” is a tough one...
- Is a statement is useful?
  - Optionally: For a conjecture
- What are the dependencies of statement? (premise selection)

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: “Is $a^n + b^n = c^n$ true?” is a tough one...
- Is a statement is useful?
  - Optionally: For a conjecture
- What are the dependencies of statement? (premise selection)
- Should a theorem be named? How?

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: "Is $a^n + b^n = c^n$ true?" is a tough one...
- Is a statement is useful?
  - Optionally: For a conjecture
- What are the dependencies of statement? (premise selection)
- Should a theorem be named? How?
- What should the next proof step be?
  - Tactic? Instantiation?

All Problems for Machine Learning

- Is my conjecture true?
  - Be careful: “Is $a^n + b^n = c^n$ true?” is a tough one...
- Is a statement is useful?
  - Optionally: For a conjecture
- What are the dependencies of statement? (premise selection)
- Should a theorem be named? How?
- What should the next proof step be?
  - Tactic? Instantiation?
- What new problem is likely to be true?
  - Intermediate statement for a conjecture

Premise selection

Intuition

Given:

- set of theorems T (together with proofs)
- conjecture c

Find: minimal subset of T that can be used to prove c

More formally

$$\arg\min_{t \subseteq T} \{ |t| \mid t \vdash c \}$$

(or $\emptyset$ if not provable)

Note: implicit assumption on a proving system. ATP in practice.

In machine learning terminology

Multi-label classification

Input: set of samples $\mathbb{S}$, where samples are triples $s, F(s), L(s)$

- s is the sample ID
- $F(s)$ is the set of features of s
- $L(s)$ is the set of labels of s

Output: function f : features $\rightarrow$ labels

Predicts n labels (sorted by relevance) for set of features

Sample features

Sample add_comm ($a + b = b + a$) characterized by:

- F(add_comm) = {“+”, “=”, “num”}
- L(add_comm) = {num_induct, add_0, add_suc, add_def}

Not exactly the usual machine learning problem

Labels correspond to premises and samples to theorems

- Very often same, a theorem is usually proved so it is both something that can be predicted and is a label of a training example. If a is proved using b and b is proved using c should we predict c for a as well? Should we predict a for a itself?

Not exactly the usual machine learning problem

Labels correspond to premises and samples to theorems

- Very often same, a theorem is usually proved so it is both something that can be predicted and is a label of a training example. If a is proved using b and b is proved using c should we predict c for a as well? Should we predict a for a itself?

Similar theorems are likely to be useful in the proof

- Also likely to have similar premises

Not exactly the usual machine learning problem

Labels correspond to premises and samples to theorems

- Very often same, a theorem is usually proved so it is both something that can be predicted and is a label of a training example. If *a* is proved using *b* and *b* is proved using *c* should we predict *c* for *a* as well? Should we predict *a* for *a* itself?

Similar theorems are likely to be useful in the proof

- Also likely to have similar premises

Theorems sharing logical features are similar

- Theorems sharing rare features are very similar

Not exactly the usual machine learning problem

Labels correspond to premises and samples to theorems

- Very often same, a theorem is usually proved so it is both something that can be predicted and is a label of a training example. If *a* is proved using *b* and *b* is proved using *c* should we predict *c* for *a* as well? Should we predict *a* for *a* itself?

Similar theorems are likely to be useful in the proof

- Also likely to have similar premises

Theorems sharing logical features are similar

- Theorems sharing rare features are very similar

Temporal order

- Recently considered theorems and premises are important (also in evaluation)

Not exactly for the usual machine learning tools

Needs efficient learning and prediction

- Frequent major data updates: Whenever a user changes some definition or statement does it mean we have to relearn everything?
- Automation cannot wait more than 10 seconds
- Often less, the user / reasoning can do more in that time...

Multi-label classifier output

- Often asked for thousands of most relevant lemmas
- Most tools do 3–10

Easy to get many interesting features

- Complicated feature relations
- Both linguistic techniques and feature space reduction: PCA / LSA / ...?

Additional Literature (not required)

The paper describes a number of learning problems on a particular dataset.

Cezary Kaliszyk, François Chollet, and Christian Szegedy.
Holstep: A machine learning dataset for higher-order logic theorem proving.
In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Summary

This Lecture

- Machine Learning problems
- Lemma selection

Next

- k-nearest neighbours and naive Bayes classifiers
- other statistical methods
- kernels
- decision trees
- random forests

Work Here / Homework

Find a large interactive proof

- Present an outline
- Try to characterize the main statement and its main premises
- Apply the ML problems to this informally
  (e.g. where to search main premises, main proof techniques, ...)

Work Here / Homework

Find a large interactive proof

- Present an outline
- Try to characterize the main statement and its main premises
- Apply the ML problems to this informally
  (e.g. where to search main premises, main proof techniques, ...)

Find a large automated proof

- Discuss main technique
- Which ML problems are applicable to shorten / speed up the proof?