universität
innsbruck

Machine Learning for Theorem Proving

Lecture 5 (VU)

Cezary Kaliszyk

Overview

Last Lecture

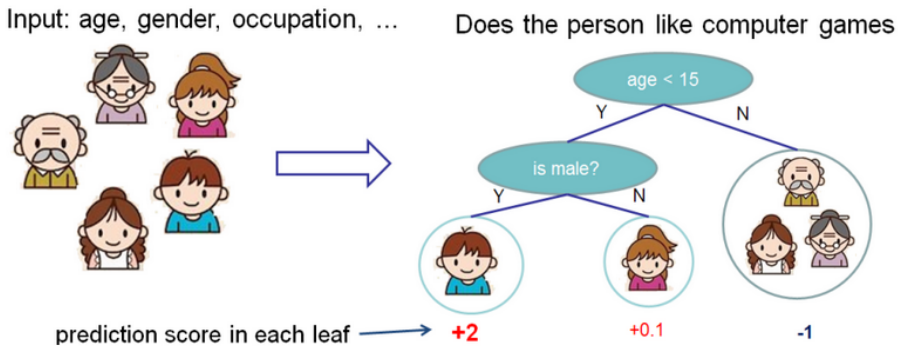
- Syntactic methods for premise selection
- Random forests

Today

- short reminder on neural networks
- deep learning for premise selection

Decision Trees (1/2)

Decision trees explained (reminded) by example. Given a set of samples characterized by features, we build a tree that in a leaf stores the average of the set of samples with these features.

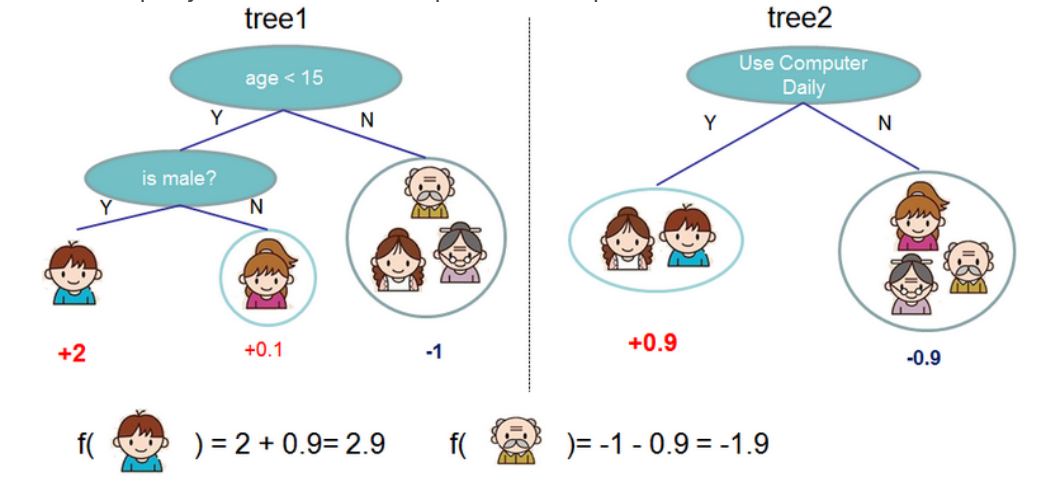


We build a number of such trees with different pre-selected features as the decision nodes.

[*Chen,Guestrin*]

Decision Trees (2/2)

In order to query the forest, we now pass our sample to all the trees and sum the results.



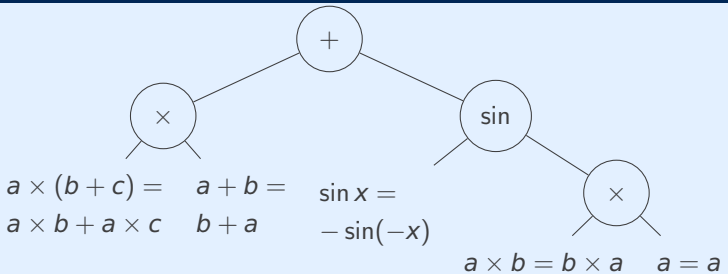
[*Chen,Guestrin*]

Decision Trees for premise selection

Definition

- each leaf stores a set of samples
- each branch stores a feature f and two subtrees, where:
  - the left subtree contains only samples having f
  - the right subtree contains only samples not having f

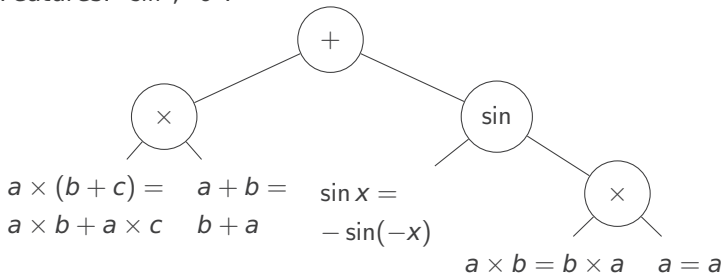
Example



Single-path query

Note that we only get predictions that exactly match the features. But what if a single features is missing or is too much? Then single path-query does not work too well. Two examples:

Query tree for conjecture “$\sin(0) = 0$”.
Features: "sin", "0".

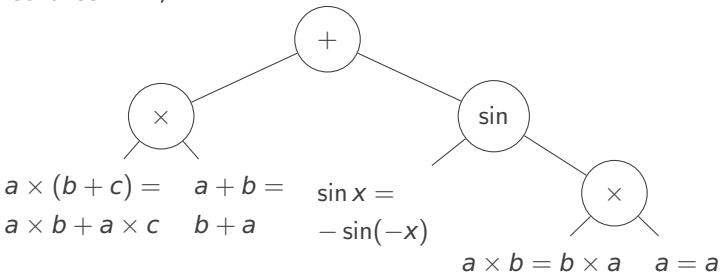


The overall result will be the premises of $\sin x = -\sin(-x)$.

Single-path query (2)

Example 2

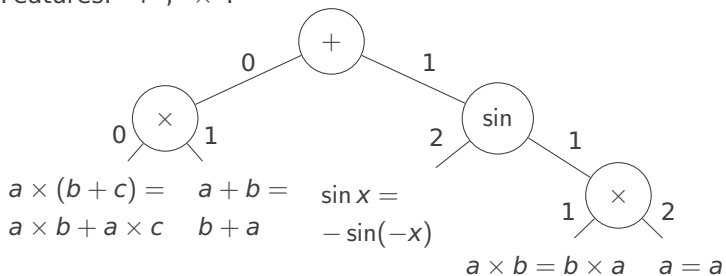
Query tree for conjecture “$(a + b) \times c = a \times c + b \times c$”.
Features: "+", "×".



$a \times b = b \times a$ is not considered!

Multi-path query

Weight samples by the number of errors on each path.
Features: “+”, “×”.



By allowing for one or two “errors”, we also get an improved ranking of further premises.

Splitting feature

An important choice in building the random forest is the selection of features that will be selected in the next tree. Different approaches are common:

Agrawal et al.

- Take n random features from samples and choose feature with lowest Gini impurity (probability of mis-labeling)
- Problem: Gini impurity calculation slow
- Choose feature that divides samples most evenly ($|S_f| \approx |S_{\neg f}|$)

Online / Offline forests

tree is updated or completely rebuilt [*Agraval, Saffari*]

Approach for premise selection

- when a branch learns new samples, check whether the branch feature is still an optimal splitting feature wrt. the new data
- if yes, update subtrees with new data
- if no, rebuild tree

Homework

Decision Tree

- Build a premise selection decision tree for the Mizar dataset
  `http://cl-informatik.uibk.ac.at/teaching/ws23/mltp/mltp.tgz`
  This means: choose a splitting feature up to a certain depth.
- Predict the useful theorems for new features using the tree
- No need to include any optimizations / multiple trees
  (but this gives you bonus points if you missed any homeworks)