## Making Woody Parallel

### Applying Parallel Computing to decision trees

Hugh McGrade wbr412@alumni.ku.dk Mads Obitsø scr411@alumni.ku.dk Titus Robroek robroek@di.ku.dk

Department of Computer Science DIKU

November 6, 2018



## Introduction

#### Making Woody Parallel

### Introduction

### Working with Woody

Decision tree evaluation Extracting from Woody

Interoperation of Woody ar Futhark

Tree and test data encodir for Futhark

#### Writing Futhark

Basic Futhark implementation

implementation

Eiltoring Eutha

Transalus Deservation

Treesolve Matri:

#### Experimen

Evanrimontal Catu

Results

-----

0---!--

■ What is Woody?

■ Decision trees and Machine Learning



## Working with Woody Decision tree evaluation

Making Woody Parallel

Introductio

Working with Woody

Decision tree evaluation

Extracting from Woody

Futhark

Tree and test data encoding for Futhark

#### Writing Futhar

Basic Futhark implementation Layered Futhark

Filtering Fuths

implementation

Treesolve Matri:

#### Experimen

Experimental Se

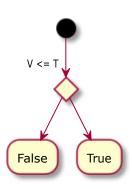
Evaluation

Future Resea

Conclusion

A decision tree consists of nodes. Non-leaf nodes consist of a conditional check, where:

- V: Feature value checked for this node
- T: Threshold value of the node that is checked against





## Working with Woody Extracting from Woody

#### Making Woody Parallel

#### Introduction

Working with Woody

Decision tree evaluation

Extracting from Woody

Interoperation of Wo

Tree and test data encodi

for Futhark

#### Writing Futhar

Basic Futhark implementation

implementatio

Filtering Futha

Treesolver Precomp

Treesolve Matri

#### Experimen

Experimental Set

Results

\_ - - -

- Python library run against a dataset
- Split data into training and test set
- 3 Fit on training set to create a forest of trees
- Run predictions with test set on this forest



# Working with Woody Interoperation of Woody and Futhark

#### Making Woody Parallel

#### Introduction

#### Working with Woody

Decision tree evaluation

Extracting from Woody
Interoperation of Woody and

Futhark

for Futhark

#### Writing Futhark

Basic Futhark implementation Lavered Futhark

implementation

Filtering Futhar

Treesolver Precompt

Treesolve Matri

#### Experimer

Evanrimontal Cat

Results

-----

Conclusion





### Run as a library or run stand-alone?



# Working with Woody Tree and test data encoding for Futhark

### Making Woody Parallel

#### Introduction

Working with Woody

Decision tree evaluati Extracting from Wood

Interoperation

Tree and test data encoding

#### Mission Futbook

Basic Futhark implementation

Layered Futhark

implementation

implementation
Transplant Processory

Treesolve Matri:

#### Experimer

Result

Evaluation

Conclusion

In order to pass the tree and test data from Woody to Futhark, we encoded each as a series of flat arrays.

- treeLeftid
- treeRightid
- treeFeature
- treeThres\_or\_leaf
- Xtest
- nXtest
- dXt.est.



# Writing Futhark Basic Futhark implementation

#### Making Woody Parallel

#### Introduction

#### Working with Woody

Decision tree evaluation

Extracting from Woody

Futhark

Tree and test data encoding for Futbark

#### Writing Futhark

#### Basic Futhark implementation

Layered Futha

implementation

Filtering Fuths

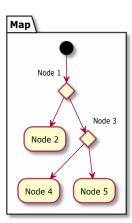
Transactions December

Treesolve Matri

#### Experimen

Experimental Setur

Results





# Writing Futhark Basic Futhark implementation

#### Making Woody Parallel

#### Introduction

#### Working with Woody

Decision tree evaluation

Interoperation of W

Tree and test data encodi

#### Writing Futhark

#### Basic Futhark implementation

Layered Futha

Eiltoring Euth

Transal as Dansan

Treesolve Matrix

#### Experimen

Experimental Set

Result

Futuro Rospor

```
unsafe map (\ i →
let idx = if dindices > 0 then indices[i]
   else i
let row start = idx * dXtest
in loop node_id = TREE_ROOT_ID
      while treeLeftid[node_id] !=
         TREE CHILD ID NOT SET do
            if Xtest[row_start +
               treeFeature[node_id]] <=
               treeThres_or_leaf[node_id]
              then treeLeftid[node_id]
              else treeRightid[node_id]
           ) (iota n_preds)
```



# Writing Futhark Layered Futhark implementation

#### Making Woody Parallel

#### Introduction

#### Working with Woody

Decision tree evaluation

Extracting from Woody

Futhark

Tree and test data encoding

#### Writing Futhark

Basic Futhark implementation

#### Layered Futhark implementation

implementation

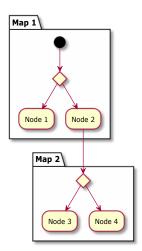
Filtering Fuths

Treesolver Precomp

#### Treesolve Matri

Results

Evaluation





# Writing Futhark Layered Futhark implementation

#### Making Woody Parallel

Introduction

Working with Woody

Decision tree evaluat

Interoperation of Woody a

Tree and test data end for Futbark

#### Vriting Futharl

Basic Futhark implementation

Layered Futhark implementation

Filteriae Futbo

Treesolver Precompu

Experiment

Experimental Se

Evaluation

Conclusion

```
loop node_array for row in iota(depth) do
    unsafe map (\ (node_id, data_row_start)
                  if (treeLeftid[node_id] !=
                     0)
                     then (if Xtest[
                         data row start +
                         treeFeature[node_id]]
                        <= treeThres_or_leaf[</pre>
                         node_id1
                         then treeLeftid[
                             node_id1
                         else treeRightid[
```

node\_id])

Thi abon age



# Writing Futhark Filtering Futhark implementation

### Making Woody Parallel

#### Introduction

#### Working with Woody

Decision tree evaluation

Interoperation of Woody a

Tree and test data enci

#### Writing Futhark

Basic Futhark implementation Layered Futhark

#### Filtering Futhark implementation

Treesolver Precompu

#### Evporimor

Results

⊏valuation

0---!--

 Layered provides a structured way of iterating over layers

- Many passes end early
- These are still considered by the maps



### Writing Futhark Treesolver Precompute

#### Making Woody Parallel

#### Working with Woody

Extracting from Woody

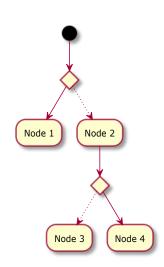
Tree and test data encoding

#### Writing Futhark

Basic Futhark implementation

#### Treesolver Precompute

Results





## Writing Futhark **Treesolver Precompute**

#### Making Woody Parallel

### Working with Woody

Basic Futhark implementation

### Treesolver Precompute

Results

```
let next node
  (row: []f64)
  ((left, right, feature, thres): (i32, i32,
      i32, f64)) : i32 =
    row[feature] <= thres then left else
     right
```

let make next tree (tree: [](i32, i32, i32, f64))

(row : []f64) : []i32 =

map (next\_node row) tree

let traverse

 $(next\_nodes: []i32): i32 =$ 

let (last current) - (0 next hodes[0])



# Writing Futhark Treesolve Matrix

#### Making Woody Parallel

```
Introduction
```

```
Working with Woody
```

Extracting from Woody

Futhark

#### Writing Futher

Basic Futhark implementation Lavered Futhark

Eiltoring Eutho

implementation

### Treesolve Matrix

### Experiment

Experimental Set Results

Future Research

Conclusion

```
let repeated_criteria = flatten (replicate
    nXtest treeFeature)
let repeated offsets = flatten (man () i
```

let repeated\_offsets = flatten (map (\ i ->
 replicate treelength i) (steps 0 nXtest
 dXtest))

let flcr = map2 (+) repeated\_offsets
 repeated\_criteria

let scattered\_features = unsafe map (\ i ->
 Xtest[i]) flcr

let threshold\_result = map2 (<=)
 scattered\_features (flatten (replicate
 nXtest treeThres\_or\_leaf))</pre>

let  $left_or_right = (\ b \ l \ r \rightarrow if \ b \ then \ l$  else r)

let repeated left = flatten (replicate nXtest)



# Experimental Setup

#### Making Woody Parallel

Introduction

Working with Woody

Decision tree evaluation

Extracting from Wood

Futhark

Tree and test data encoding

for Futhark

Writing Futhar

Basic Futhark implementation

implementatio

Filtering Futhark

Treesolver Precompu

Experimen

Experimental Setup

Results

Evaluation

Conclusio

To evaluate our Futhark implementation, we performed various comparisons with the woody implementation.

—Perhaps add an extra slide with an overview of the framework as an image? Itemize the tests—



## Experiment Results: Varying train data size

### Making Woody Parallel

#### Introduction

#### Working with Woody

Decision tree evaluatio

Interoperation of Woody Futhark

Tree and test data encoding

#### Writing Futhark

Basic Futhark implementation

implementation

Filtering Futharl implementation

Treesolver Precompu

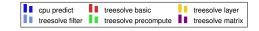
#### Evnorimor

#### Results

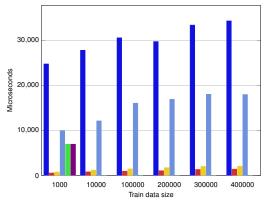
\_ . . . .

\_ . . .

Conclusion



#### Microseconds used predicting a single tree as average of 10 runs







## Experiment Results: Varying number of predictions

#### Making Woody Parallel

Introduction

#### Working with Woody

Docision trop auglication

Extracting from Woody
Interoperation of Woody a

Tree and test data enco

#### Writing Futhark

Basic Futhark implementation Lavered Futhark

implementation

Filtering Futhark

Transalus Matr

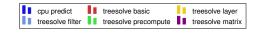
#### Experimen

Experimental Sets

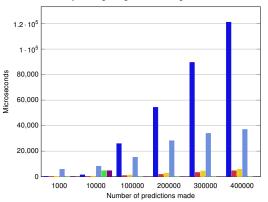
#### Results

Eutura Passar

Conclusion



#### Microseconds used predicting a single tree as average of 10 runs







## Experiment Evaluation

#### Making Woody Parallel

#### Working with Woody

#### Writing Futhark

Basic Futhark implementation

Results

Evaluation

- Basic and layer are fast.
- GPU code scales better with bigger tests



## Experiment Future Research

#### Making Woody Parallel

#### Working with Woody

### Writing Futhark

Basic Futhark implementation

Results

Future Research

- Matrix and Precompute might be promising
- GPU code scales better with bigger tests



## Conclusion

### Making Woody Parallel

Working with Woody

Basic Futhark implementation

Conclusion

We have proposed a number of approaches to parallelising the evaluation of decision trees using Futhark. Our findings show that as a whole this parallelisation is promising for the performance of decision tree evaluation on large datasets.



#### Making Woody Parallel

#### Introduction

#### Working with Woody

Extracting from Woody Interoperation of Woody and

Tree and test data encodi for Futhark

#### Writing Futhark

Basic Futhark implementation Layered Futhark

implementation

Treesolver Precomout

E....

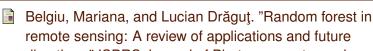
Experiment

= ...

Evaluation

Future Research

#### Conclusion



remote sensing: A review of applications and future directions." ISPRS Journal of Photogrammetry and Remote Sensing 114 (2016): 24-31.

Gieseke, Fabian, and Christian Igel. "Training Big Random Forests with Little Resources." arXiv preprint arXiv:1802.06394 (2018).