Scalability of Predictive Modeling Algorithms

Master's thesis presentation

Ing. Tomáš Frýda

Supervisor: Ing. Pavel Kordík, Ph.D.

Outline

FAKE GAME

Base Models

Ensembles

Evolution

Outline

Outline

Outline

FAKE GAME
- Base Models
- Ensembles
- Evolution

H2O
- Overview
- Architecture
- Usage
- Related Software

Hyper-parameter Optimization
- SMBO

Implementation

Outline

Outline

Outline

Motivation

- Model selection usually does not depend solely on predictive performance
- I took in to account time, which gives me two basic use cases:
  - Good enough model trained using limited computational resources
  - Highly accurate model trained using as much computational resources as needed
- Make FAKE GAME usable on big data

FAKE GAME

- Originally created for small data
- Base models
  - Decision tree, KNN, etc
  - Regression models
- Ensembles
  - Bagging
  - Boosting
  - Stacking, Cascade Correlation, ...
- Genetic programming-based ensemble creation

Base Models

- regression models
  - linear
  - polynomial
  - sigmoid
  - sine, . . .
- regression models are used as discriminant functions for classification
- decision trees
- k-NN
- . . .

Ensembles

- Arbitrating[1]
- Bagging
- Boosting
- Cascade Generalization
- Cascading[1]
- Delegating[1]
- Stacking

[1] used only for classification

Evolution

Genetic programming used for evolving templates that can be expanded to hierarchical ensembles

H2O

- framework for distributed machine learning based on MapReduce
- support for preprocessing and data manipulation
- RESTful API used by various language bindings (R, Python, . . .)

Overview

Machine Learning algorithms included in H2O:

- Deep Learning
- Distributed Random Forest
- Gradient Boosting Machines
- Generalized Linear Model
- Naïve Bayes
- K-Means
- PCA
- GLRM
- ...

Architecture

- in-memory MapReduce
- uses distributed key-value storage
- tries to keep related data in the same or nearby node in order to minimize network usage
- columns are compressed and lazily decompressed just in time of usage in CPU registers
- parallel data load

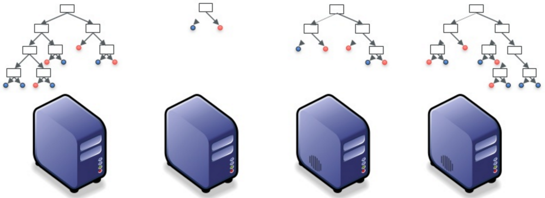
See more at
http://blog.h2o.ai/2014/03/h2o-architecture/

How are decision trees built in H2O?

Implementation #1

Build independent trees per machine local data

- RVotes approach
- Each node builds a subset of forest



0xdata

Chawla, N., & Hall, L. (2004). *Learning ensembles from bites: A scalable and accurate approach.* The Journal of Machine Learning Research, 5, p421–451.

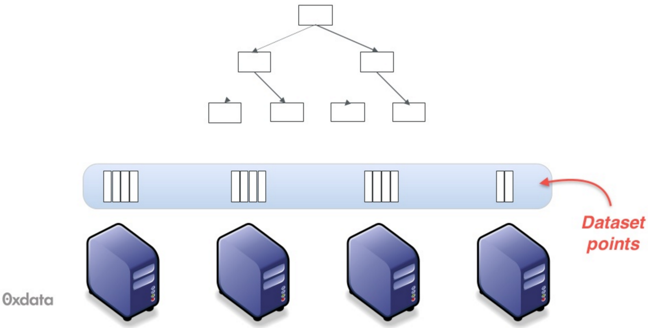
Implementation #1

- ✔ Fast - trees are independent and can be built in parallel
- ⊘ Data have to fit into memory
- ⊘ Possible accuracy decrease if each node can see only subset of data

0xdata

Implementation #2

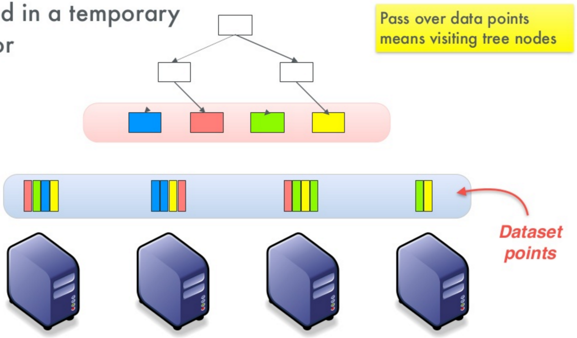
Build a distributed tree over all data



Implementation #2

Each data point has assigned a tree node

- stored in a temporary vector

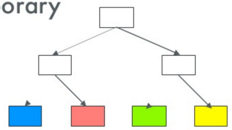


0xdata

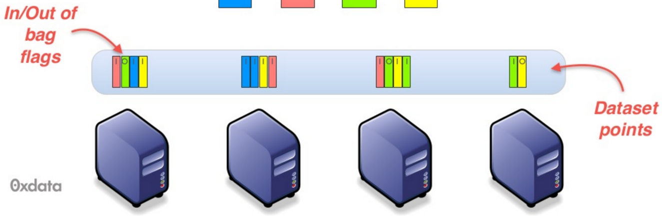
Implementation #2

Each data point has in/out of bag flag

- stored in a temporary vector



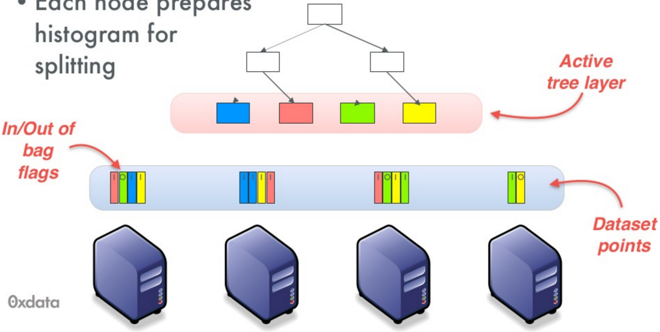
Trick for on-the-fly scoring: position out-of-bag rows inside a tree is tracked as well



Implementation #2

Tree is built per layer

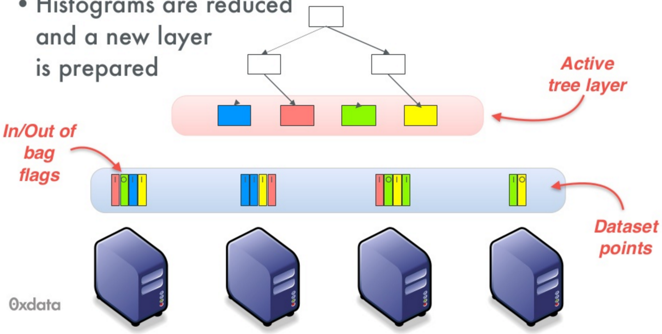
- Each node prepares histogram for splitting



Implementation #2

Tree is built per layer

- Histograms are reduced and a new layer is prepared



Implementation #2

- ✔ Exact solution - no decrease of accuracy
- ✔ Elegant solution merging tree building and OOB scoring
- ⊘ More data transfers to exchange histograms
- ⊘ Can produce huge trees (since tree size depends on data)

0xdata

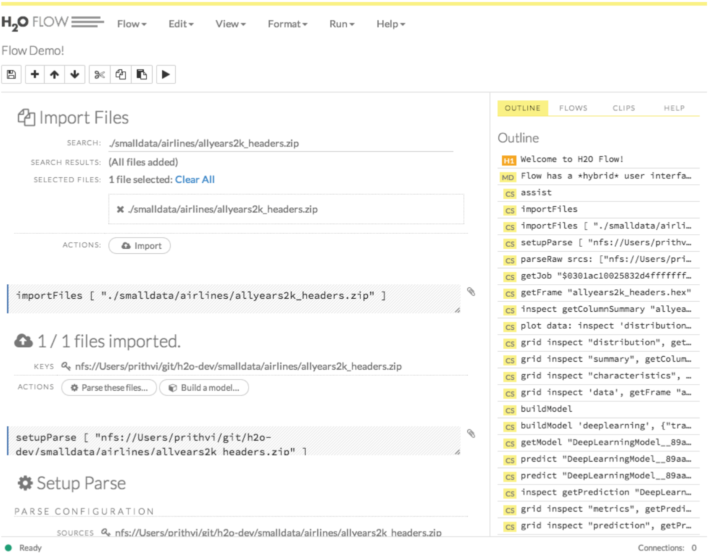
Python

```python
import h2o

h2o.init()
data = h2o.import_file(
        path="data.csv")

# Create test/train split
s = data["Year"].runif()
train = data[s <= 0.75]
test  = data[s > 0.75]


# Create an estimator
dl=H2ODeepLearningEstimator(
     hidden=[10,10],
     epochs=5,
     balance_classes=True)

# Train an estimator
dl.train(
     x=myX,
     y=myY,
     training_frame=train,
     validation_frame=test)
```

R

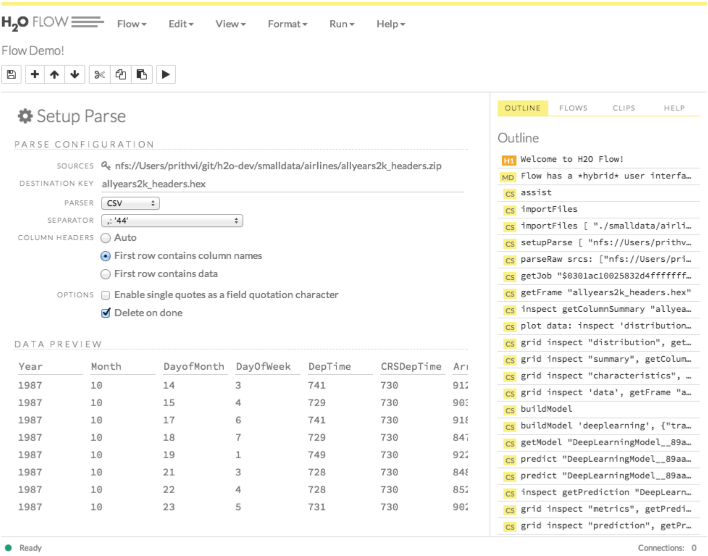
```r
library("h2o")

h2o.init()
dt <- h2o.importFile(
     path = "data.csv")


# Create test/train split
dt.split <- h2o.splitFrame(
     data = dt,
     ratios = 0.75)
train <- dt.split[[1]]
test <- dt.split[[2]]


# Create an estimator and
# train it
dl <- h2o.deeplearning(
  x = myX,
  y = myY,
  training_frame = train,
  validation_frame = test,
  hidden=c(10, 10))
```

H2O Flow

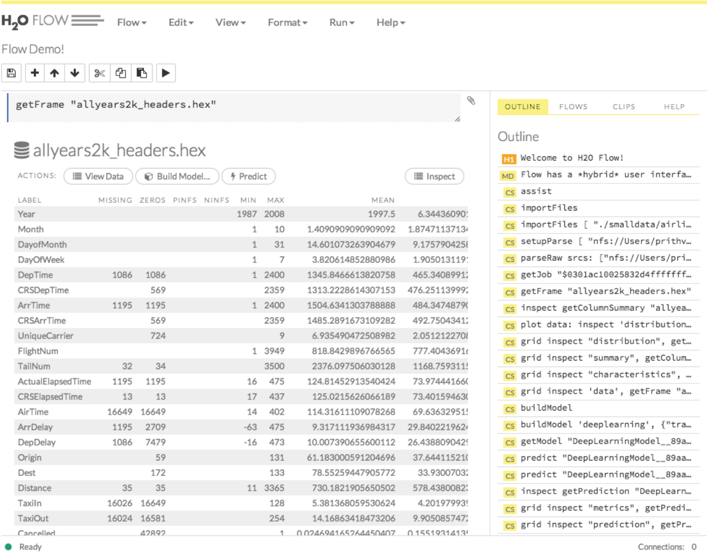


H2O FLOW
Flow Edit View Format Run Help
Flow Demo!
Import Files
SEARCH: ./smalldata/airlines/allyears2k_headers.zip
SEARCH RESULTS: (All files added)
SELECTED FILES: 1 file selected: Clear All
./smalldata/airlines/allyears2k_headers.zip
ACTIONS: Import
importFiles ["./smalldata/airlines/allyears2k_headers.zip"]
1 / 1 files imported.
KEYS nfs://Users/prithvi/git/h2o-dev/smalldata/airlines/allyears2k_headers.zip
ACTIONS Parse these files... Build a model...
setupParse ["nfs://Users/prithvi/git/h2o-dev/smalldata/airlines/allyears2k_headers.zip"]
Setup Parse
PARSE CONFIGURATION
SOURCES nfs://Users/prithvi/git/h2o-dev/smalldata/airlines/allyears2k_headers.zip
OUTLINE FLOWS CLIPS HELP
Outline
H1 Welcome to H2O Flow!
MD Flow has a *hybrid* user interfa…
CS assist
CS importFiles
CS importFiles ["./smalldata/airli…
CS setupParse ["nfs://Users/prithv…
CS parseRaw srcs: ["nfs://Users/pri…
CS getJob "$0301ac10025832d4fffffff…
CS getFrame "allyears2k_headers.hex"
CS inspect getColumnSummary "allyea…
CS plot data: inspect 'distribution…
CS grid inspect "distribution", get…
CS grid inspect "summary", getColum…
CS grid inspect "characteristics", …
CS grid inspect 'data', getFrame "a…
CS buildModel
CS buildModel 'deeplearning', {"tra…
CS getModel "DeepLearningModel__89a…
CS predict "DeepLearningModel__89aa…
CS predict "DeepLearningModel__89aa…
CS inspect getPrediction "DeepLearn…
CS grid inspect "metrics", getPredi…
CS grid inspect "prediction", getPr…
Ready
Connections: 0

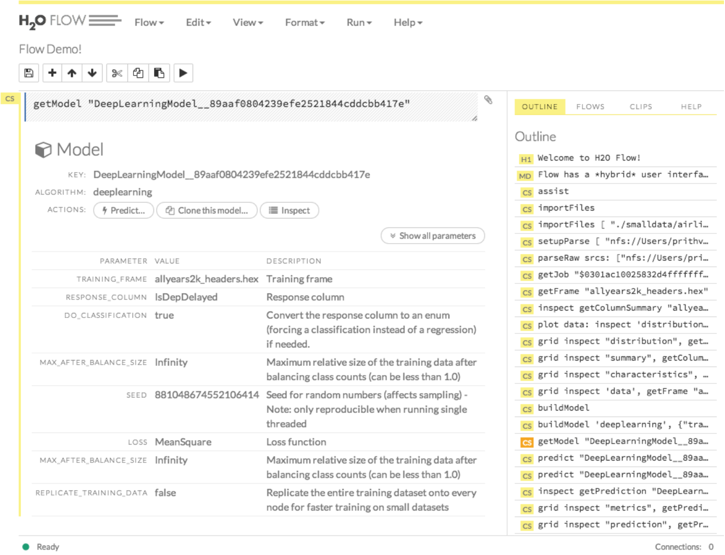
H2O Flow


H2O FLOW
Flow
Edit
View
Format
Run
Help
Flow Demo!
Setup Parse
PARSE CONFIGURATION
SOURCES nfs://Users/prithvi/git/h2o-dev/smalldata/airlines/allyears2k_headers.zip
DESTINATION KEY allyears2k_headers.hex
PARSER CSV
SEPARATOR ,: '44'
COLUMN HEADERS Auto
First row contains column names
First row contains data
OPTIONS Enable single quotes as a field quotation character
Delete on done
DATA PREVIEW
Year Month DayofMonth DayOfWeek DepTime CRSDepTime
1987 10 14 3 741 730
1987 10 15 4 729 730
1987 10 17 6 741 730
1987 10 18 7 729 730
1987 10 19 1 749 730
1987 10 21 3 728 730
1987 10 22 4 728 730
1987 10 23 5 731 730
OUTLINE FLOWS CLIPS HELP
Outline
H1 Welcome to H2O Flow!
MD Flow has a *hybrid* user interfa…
CS assist
CS importFiles
CS importFiles ["./smalldata/airli…
CS setupParse ["nfs://Users/prithv…
CS parseRaw srcs: ["nfs://Users/pri…
CS getJob "$0301ac10025832d4fffffff…
CS getFrame "allyears2k_headers.hex"
CS inspect getColumnSummary "allyea…
CS plot data: inspect 'distribution…
CS grid inspect "distribution", get…
CS grid inspect "summary", getColum…
CS grid inspect "characteristics", …
CS grid inspect 'data', getFrame "a…
CS buildModel
CS buildModel 'deeplearning', {"tra…
CS getModel "DeepLearningModel__89a…
CS predict "DeepLearningModel__89aa…
CS predict "DeepLearningModel__89aa…
CS inspect getPrediction "DeepLearn…
CS grid inspect "metrics", getPredi…
CS grid inspect "prediction", getPr…
Ready
Connections: 0

H2O Flow


H2O FLOW
Flow
Edit
View
Format
Run
Help
Flow Demo!
getFrame "allyears2k_headers.hex"
allyears2k_headers.hex
ACTIONS:
View Data
Build Model...
Predict
Inspect
LABEL MISSING ZEROS PINFS NINFS MIN MAX MEAN
Year 1987 2008 1997.5
Month 1 10 1.4090909090909092
DayofMonth 1 31 14.601073263904679
DayOfWeek 1 7 3.820614852880986
DepTime 1086 1086 1 2400 1345.8466613820758
CRSDepTime 569 2359 1313.2228614307153
ArrTime 1195 1195 1 2400 1504.6341303788888
CRSArrTime 569 2359 1485.2891673109282
UniqueCarrier 724 9 6.935490472508982
FlightNum 1 3949 818.8429896766565
TailNum 32 34 3500 2376.097506030128
ActualElapsedTime 1195 1195 16 475 124.81452913540424
CRSElapsedTime 13 13 17 437 125.0215626066189
AirTime 16649 16649 14 402 114.31611109078268
ArrDelay 1195 2709 -63 475 9.317111936984317
DepDelay 1086 7479 -16 473 10.007390655600112
Origin 59 131 61.183000591204696
Dest 172 133 78.55259447905772
Distance 35 35 11 3365 730.1821905650502
TaxiIn 16026 16649 128 5.381368059530624
TaxiOut 16024 16581 254 14.16863418473206
Cancelled 42892 1
OUTLINE
FLOWS
CLIPS
HELP
Outline
H1 Welcome to H2O Flow!
MD Flow has a *hybrid* user interfa…
CS assist
CS importFiles
CS importFiles ["./smalldata/airli…
CS setupParse ["nfs://Users/prithv…
CS parseRaw srcs: ["nfs://Users/pri…
CS getJob "$0301ac10025832d4fffffff…
CS getFrame "allyears2k_headers.hex"
CS inspect getColumnSummary "allyea…
CS plot data: inspect 'distribution…
CS grid inspect "distribution", get…
CS grid inspect "summary", getColum…
CS grid inspect "characteristics", …
CS grid inspect 'data', getFrame "a…
CS buildModel
CS buildModel 'deeplearning', {"tra…
CS getModel "DeepLearningModel__89a…
CS predict "DeepLearningModel__89aa…
CS predict "DeepLearningModel__89aa…
CS inspect getPrediction "DeepLearn…
CS grid inspect "metrics", getPredi…
CS grid inspect "prediction", getPr…
Ready
Connections: 0

H2O Flow


H2O FLOW
Flow
Edit
View
Format
Run
Help
Flow Demo!
CS
getModel "DeepLearningModel__89aaf0804239efe2521844cddcbb417e"
Model
KEY: DeepLearningModel__89aaf0804239efe2521844cddcbb417e
ALGORITHM: deeplearning
ACTIONS: Predict... Clone this model... Inspect
Show all parameters
PARAMETER VALUE DESCRIPTION
TRAINING_FRAME allyears2k_headers.hex Training frame
RESPONSE_COLUMN IsDepDelayed Response column
DO_CLASSIFICATION true Convert the response column to an enum (forcing a classification instead of a regression) if needed.
MAX_AFTER_BALANCE_SIZE Infinity Maximum relative size of the training data after balancing class counts (can be less than 1.0)
SEED 8810486745521064 14 Seed for random numbers (affects sampling) - Note: only reproducible when running single threaded
LOSS MeanSquare Loss function
MAX_AFTER_BALANCE_SIZE Infinity Maximum relative size of the training data after balancing class counts (can be less than 1.0)
REPLICATE_TRAINING_DATA false Replicate the entire training dataset onto every node for faster training on small datasets
OUTLINE FLOWS CLIPS HELP
Outline
H1 Welcome to H2O Flow!
MD Flow has a *hybrid* user interfa...
CS assist
CS importFiles
CS importFiles ["./smalldata/airli...
CS setupParse ["nfs://Users/prithv...
CS parseRaw srcs: ["nfs://Users/pri...
CS getJob "$0301ac10025832d4fffffff...
CS getFrame "allyears2k_headers.hex"
CS inspect getColumnSummary "allyea...
CS plot data: inspect 'distribution...
CS grid inspect "distribution", get...
CS grid inspect "summary", getColum...
CS grid inspect "characteristics", ...
CS grid inspect 'data', getFrame "a...
CS buildModel
CS buildModel 'deeplearning', {"tra...
CS getModel "DeepLearningModel__89a...
CS predict "DeepLearningModel__89aa...
CS predict "DeepLearningModel__89aa...
CS inspect getPrediction "DeepLearn...
CS grid inspect "metrics", getPredi...
CS grid inspect "prediction", getPr...
Ready
Connections: 0

H2O Flow

H2O FLOW Flow▾ Edit▾ View▾ Format▾ Run▾ Help▾

Flow Demo!

CS
```
grid inspect "metrics", getPredictions model:
"DeepLearningModel__89aaf0804239efe2521844cddcbb417e", frame:
"allyears2k_headers.hex"
```

metrics

Metrics for the selected predictions

CRITERIA	THRESHOLD	F1	F2	F0POINT5	ACCURACY	ERROR	PRECISION
maximum F1		0.68857443	0.8468044	0.5801671	0.525058	0.47494203	0.525058
maximum F2		0.68857443	0.8468044	0.5801671	0.525058	0.47494203	0.525058
maximum F0point5		0.68857443	0.8468044	0.5801671	0.525058	0.47494203	0.525058
maximum Accuracy		0.68857443	0.8468044	0.5801671	0.525058	0.47494203	0.525058
maximum Precision	0.0014023929	0.0063889488	0.004002683	0.015820755	0.47662467	0.52337533	1
maximum Recall		0.68857443	0.8468044	0.5801671	0.525058	0.47494203	0.525058
maximum Specificity	0.0014023929	0.0063889488	0.004002683	0.015820755	0.47662467	0.52337533	1
maximum absolute MCC	0.00034111855	0.37031785	0.3361132	0.4122729	0.43464914	0.56535083	0.44595584
minimizing max per class Error	0.00029882442	0.45200276	0.44506764	0.4591574	0.43910593	0.5608941	0.46405438

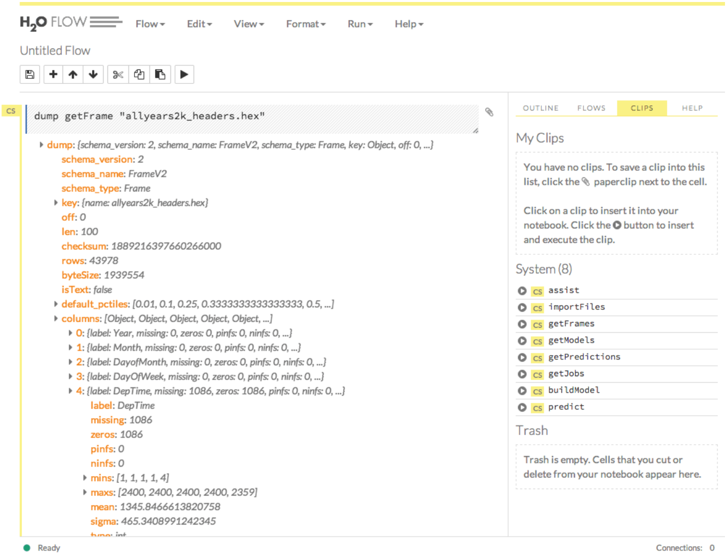
OUTLINE FLOWS CLIPS HELP

Outline

- H1 Welcome to H2O Flow!
- MD Flow has a *hybrid* user interfa…
- CS assist
- CS importFiles
- CS importFiles ["./smalldata/airli…
- CS setupParse ["nfs://Users/prithv…
- CS parseRaw srcs: ["nfs://Users/pri…
- CS getJob "$0301ac10025832d4fffffff…
- CS getFrame "allyears2k_headers.hex"
- CS inspect getColumnSummary "allyea…
- CS plot data: inspect 'distribution…
- CS grid inspect "distribution", get…
- CS grid inspect "summary", getColum…
- CS grid inspect "characteristics", …
- CS grid inspect 'data', getFrame "a…
- CS buildModel
- CS buildModel 'deeplearning', {"tra…
- CS getModel "DeepLearningModel__89a…
- CS predict "DeepLearningModel__89aa…
- CS predict "DeepLearningModel__89aa…
- CS inspect getPrediction "DeepLearn…
- CS grid inspect "metrics", getPredi…
- CS grid inspect "prediction", getPr…

Ready Connections: 0

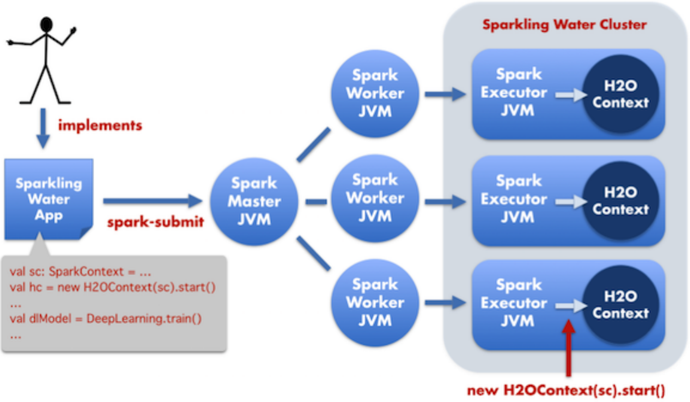
H2O Flow


H2O FLOW
Flow▾ Edit▾ View▾ Format▾ Run▾ Help▾
Untitled Flow
CS
dump getFrame "allyears2k_headers.hex"
dump: {schema_version: 2, schema_name: FrameV2, schema_type: Frame, key: Object, off: 0, ...}
schema_version: 2
schema_name: FrameV2
schema_type: Frame
key: {name: allyears2k_headers.hex}
off: 0
len: 100
checksum: 1889216397660266000
rows: 43978
byteSize: 1939554
isText: false
default_pctiles: [0.01, 0.1, 0.25, 0.3333333333333333, 0.5, ...]
columns: [Object, Object, Object, Object, Object, ...]
0: {label: Year, missing: 0, zeros: 0, pinfs: 0, ninfs: 0, ...}
1: {label: Month, missing: 0, zeros: 0, pinfs: 0, ninfs: 0, ...}
2: {label: DayofMonth, missing: 0, zeros: 0, pinfs: 0, ninfs: 0, ...}
3: {label: DayOfWeek, missing: 0, zeros: 0, pinfs: 0, ninfs: 0, ...}
4: {label: DepTime, missing: 1086, zeros: 1086, pinfs: 0, ninfs: 0, ...}
label: DepTime
missing: 1086
zeros: 1086
pinfs: 0
ninfs: 0
mins: [1, 1, 1, 1, 4]
maxs: [2400, 2400, 2400, 2400, 2359]
mean: 1345.8466613820758
sigma: 465.3408991242345
OUTLINE FLOWS CLIPS HELP
My Clips
You have no clips. To save a clip into this list, click the paperclip next to the cell.
Click on a clip to insert it into your notebook. Click the button to insert and execute the clip.
System (8)
assist
importFiles
getFrames
getModels
getPredictions
getJobs
buildModel
predict
Trash
Trash is empty. Cells that you cut or delete from your notebook appear here.
Ready
Connections: 0

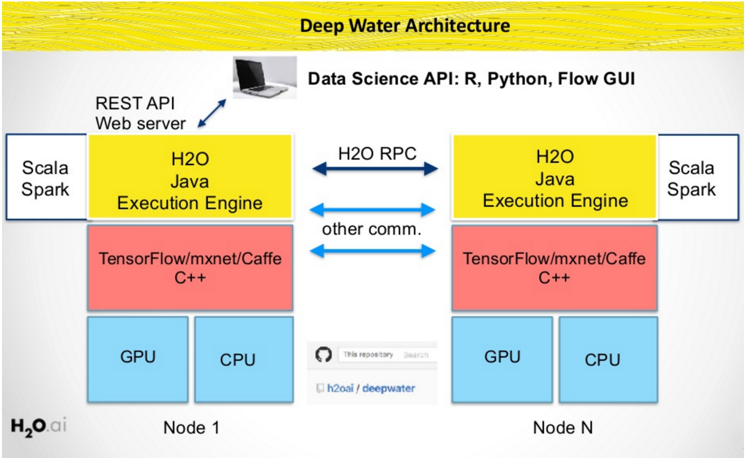
Related Software

- Sparkling Water
- Deep Water
- Steam — H2O deployment

Sparkling Water

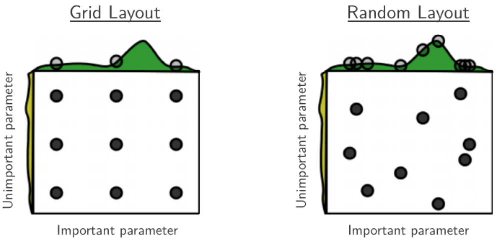


Deep Water


Deep Water Architecture
Data Science API: R, Python, Flow GUI
REST API
Web server
Scala
Spark
H2O
Java
Execution Engine
H2O RPC
other comm.
TensorFlow/mxnet/Caffe
C++
GPU
CPU
h2oai / deepwater
H2O
Java
Execution Engine
Scala
Spark
TensorFlow/mxnet/Caffe
C++
GPU
CPU
H2O.ai
Node 1
Node N

Hyper-Parameter Optimization

- Grid Search
- Random Search



- Bayesian optimization (SMAC)

Sequential Model-based Bayesian Optimization (SMBO)

1. evaluate random configuration and add it to the probabilistic model
2. select promising configuration based on probabilistic model using an acquisition function[2]
3. evaluate the configuration
4. add the new configuration to the probabilistic model
5. go to step 2

[2] usually Expected Improvement $\mathbf{E}\mathrm{I}(x) = \mathbf{E}(\max\{0, f_{t+1}(x) - f(x^+)\}|\mathrm{M}_t)$

Instances of SMBO

- Gaussian Process based SMBO
  - no obvious way how to deal with categorical parameters
- Tree-structured Parzen Estimator (TPE)
- Sequential Model-based Algorithm Configuration (SMAC)
- Hyperband

Tree-structured Parzen Estimator

$$p(x|y) = \begin{cases} l(x) & \textbf{if} \quad y < y^* \\ g(x) & \textbf{if} \quad y \geq y^* \end{cases}$$

- easy to sample space of promising values
- EI is proportional to $\left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1}$

y^* is chosen as some quantile (e.g., $p(y < y^*) = 0.15 = \gamma$)

Sequential Model-based Algorithm Configuration

- based upon ROAR — an racing optimization algorithm
- uses random forest as a probabilistic model
- usage of random forest makes it easy to use user-defined cost metric
- configuration to be evaluated is selected by following process
  1. take 10 best previously seen configurations
  2. initialize local search (using one-exchange neighbourhood for categorical, and four random neighbours for numerical variables)
  3. merge resulting 10 best configurations with 10 000 randomly sampled configurations
  4. sort by their EI
  5. interleave with uniformly sampled configurations

Hyperband

- SMBO with enhanced selection and evaluation phase
- uses information from training phase of a model that is being optimized and eventually stops it if it doesn't converge well $\implies$ explores more space using the same amount of resources
- iteratively discards the worse half of evaluated configurations

Implementation

- integration of FAKE GAME into H2O framework
- creation of benchmarking environment
  - written in Python
  - supports
    - all supervised machine learning algorithms in H2O
    - H2O Ensemble (implemented in R, based on SuperLearner package)
    - Hyper-Parameter optimization using Random Search and SMAC
  - configurable using YAML and Python

Experiments

- 2 datasets with binomial response class
  - Higgs
  - Airline - 4 different scenarios
- 20+ models benchmarked on each of 5 scenarios
- Hyper-Parameter optimization on each dataset

Overview of Results

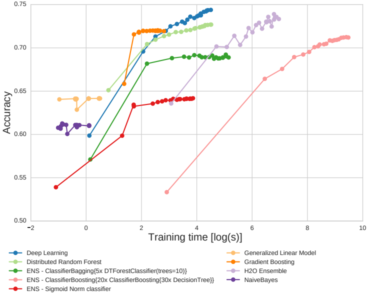


Figure: Higgs dataset

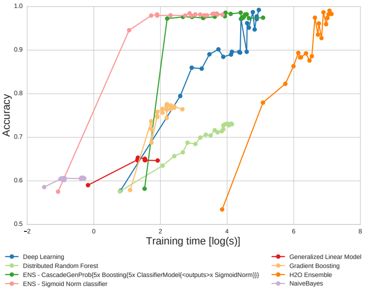
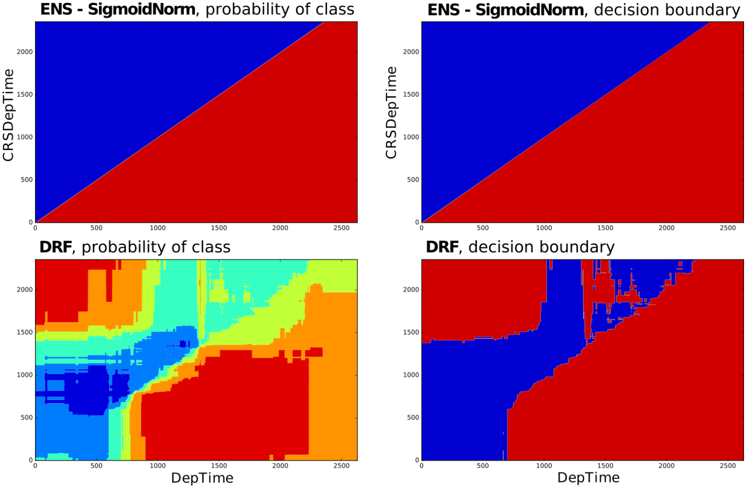
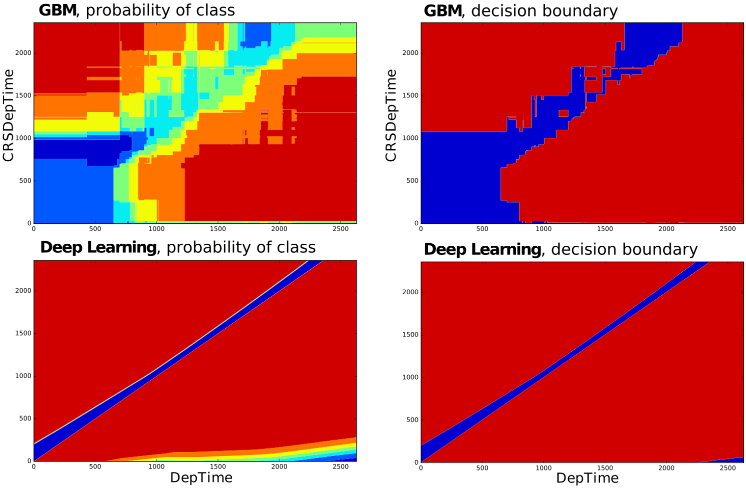


Figure: Airline – predicting IsDepDelayed

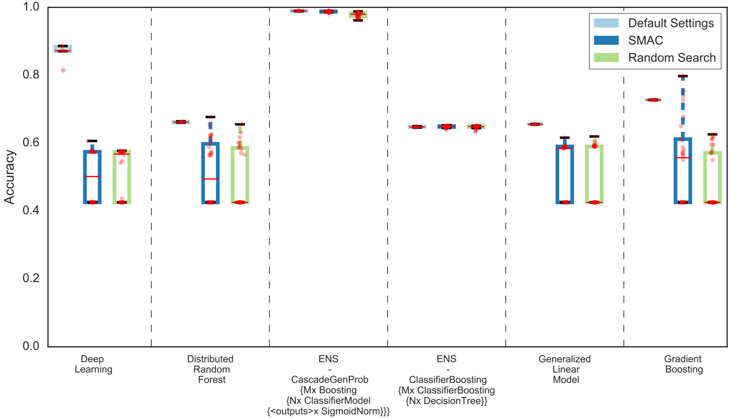
Decision Boundary on Airline dataset



Decision Boundary on Airline dataset



Hyper-Parameter Optimization



Conclusion

- Successfully integrated FAKE GAME into H2O and created benchmarking environment
- Experiments took over 2000 hours (wall clock), used 12 CPUs and 16 GiB of RAM
- Experiments show that
  - newly integrated FAKE GAME can find better models than those previously present in H2O
  - H2O's auto-tuning yields good results by default
- Results of those experiments were submitted, as part of an article, to be published in Machine Learning

0xdata,
https://www.slideshare.net/0xdata/rf-brighttalk
https://www.slideshare.net/0xdata/deep-water-gpu-deep-learning-for-h2o-arno-candel

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl,
"Algorithms for Hyper-Parameter Optimization,"
Adv. Neural Inf. Process. Syst., pp. 2546–2554, 2011.

F. Hutter, H. H. Hoos, and K. Leyton-Brown,
"Sequential model-based optimization for general algorithm configuration",
Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6683 LNCS, pp. 507–523, 2011.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar,
"Efficient Hyperparameter Optimization and Infinitely Many Armed Bandits,"
arXiv Prepr., 2016.