Bulldozers 3
Tomášek, Havránek, Hajič, Taufer
Entropy

- gives us the average amount of information in bits in some attribute of an instance
- \(-\sum_i p_i \log_2 p_i\)
Entropy

- approximately 140 boxes $\Rightarrow \log_2(140) \approx 7.129$ bits
- global entropy $\approx 6.006$

https://docs.google.com/spreadsheet/ccc?key=0Av1dVjNSTV1idDIzTjQ2QU9hVkgxdWk1QUN1c0pHbEE#gid=0
Tomasek - decision by parameters
Decision tree

Should I attack?

Have you got a weapon?

- Yes
  - Are you close enough to the enemy?
    - Yes
      - Yes
    - No
      - No

- No
  - Can you tackle the enemy bear-handed?
    - Yes
      - Yes
    - No
      - No
Decision tree - last time TODO

- don't use irrelevant enums
- sql optimizations
- multi core processing
- find some very strong Machine
upgrade - irrelevant enums

- deleted 20 less significant enums
  - filled in less than 30 % records
  - small entropy (fiModelSeries)
sql optimizations

- building additional indexes
- original query time for first iteration ~ 10 min
- with indexes 6 min
- using entropy instead of var
decision tree split

- **count**
  - very fast
  - very wrong

- **var**
  - very slow
  - good

- **entropy**
  - fast
  - good
boxes

- result fix size for $1024
  - small enough
  - $1000 error gives for 24000 rmsle 0.01
  - index can be counted in one division (price /1000)
  - but $1024 is even better
    - division 1024 can be done in one bit shift ( >> 10)
  - box id can be cached in sql ( counted columns )
Decision tree - result last time

- depth 4
- runtime 2:34:02
- result 0.54
- using var
Decision tree - result optimalized

- depth 4
- runtime 0:45:31
- result 0.46
Decision tree - final result

- depth 6
- runtime 6:34:02
- result
  - 0.26 train
  - 0.29 valid
Result overview

- bigger depth leads to over fit
- doing something better in leafs will not help
- this result is good in competition but even the winner results can't be used in real word
Genetic solution overview

- Population member
  - Expression tree
    - Evaluates price
  - Nodes
    - [Price] -> Price
    - Constant, Arithmetic, Sql Aggregation, Switch

- Fit function
  - Challenge official: RMSLE

- Reproduction
  - switching subtrees between father and mother
Genetic solution overview

- Previous results
  - 0.49 in 294 generations
Genetic solution - upgrade

- Objectives
  - More totality, less liberalism
  - More mutation
Genetic solution - upgrade

- Reproduction
  - Before:
    - Randomized subtree switching
  - Now:
    - Node type specific "DNA exchange"
      - subtree taken from node of type T (ex. atrith. plus) is alternated only by partner subtree taken from "T-node"
      - Forced reproduction breaks max-depth constraint
Genetic solution - upgrade

- Changed fit function
  - Considers "dynamism" aspect of tree (population member)
    - Static members have worse fit
Genetic solution - upgrade

- Not mutating pruning
  - Subtrees independent on evaluated data row replaced by constant
    - Scope for softer mutation convergence
      - example...
Genetic solution - upgrade

- More mutation
  - Pruning
    - Random subtree replace by 0 constant
      - simulates "totalitarian" supervision
  - Mutation for SqlAgg nodes
    - Changing agg function
    - Added: extension/reduction filter parameters
      - data columns
  - For EnumSwitch nodes
    - Switching / cloning, replacing subtrees
      - example...
Genetic solution - upgrade

- Startup population
  - Reusing older results
    - Loading serialized trees

- Performance
  - Sql result caching
    - By exactly same parameters (data column values)
  - Lower startup population count
    - Mutation process provides new every generation
      - Not exactly pure mutation
Genetic solution - results

- Weekend run on i7
  - 8 threads (4 cores HT)
  - Run parameters:
    - Threadpool size: 16
    - Pop count: 30
      - + Loaded older members
    - Min depth: 3
      - But partly eliminated changed fit function
    - Max depth: 12
      - But the new reproduction
    - Probabilities: R = 0.6, C = 0.1, M = 0.7
    - Train data count: 100k
      - changing every generation
Genetic solution - results

- Weekend run on i7
  - 180 generations
    - Previous test train data set size was 40k
  - Best result fit
    - train 0.284
    - valid 0.341
Genetic solution - results

- Alpha male example...
Neural networks

Added state, fiProductClassDesc indicators

Better guesswork for YearMade

Classifier accuracy went up to 29.15 on training set

RMSLE on training set 0.425
Actual usability

Actual RMSLE for challenge winner: 0.22910

On a bulldozer priced at 22000 GBP (cca. median value of training set), that indicates an average error of 5664.5 GBP, or >25%
Actual usability

Our best system RMSLE: 0.341

On a bulldozer priced at 22000 GBP, that indicates an average error of 8940 GBP, or > 40%
Actual usability

We believe both these results to be hardly usable for an actual Blue Book

(We think expert judgment would provide a far smaller error for a similar cost)
Q & A
Bonus - Angry Birds AI Competition
why is this interesting?

- everybody knows this game
- best results from last year can not beat human players in score
- you can play angry birds by doing this competition
- planning and physics
Angry Birds AI Competition

- well known game
- there is some Basic Game Playing Software
  - basic java code which can
    - analyse a video game frame
      - get bounding box of components
    - calculates trajectories of birds
    - game playing
What to do

- **planning**
  - what to attack with witch bird
  - special action birds planing
- **physics**
  - "what will happens when I attack here?"
  - "where to attack to destroy this building?"
Discussion