An Agnostic Framework for Uniform ML Experiments

Lorenz Weizsäcker FG Knowledge Engineering TU Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Related Talks at EuroScipy 2010:

- Nicolas Chauvat
  Knowledge management of numerical experiments
- André Espaze
  Pre and post-processing with Salome
- Andrew Davison
  Automated tracking of computational experiments using Sumatra

Agnostic Frameworks

TECHNISCHE UNIVERSITÄT DARMSTADT

Assume an ML research project and you are to prepare code that makes accomplishment of experiments less painful.

Unfortunately, you do not have any idea what project will be such that the code must not cover

- definition of losses
- data or data specifications
- training algorithm
- preprocessing code
- evaluation schemes

No algorithms, no data structures.

Agnostic Frameworks

Peewit

TECHNISCHE UNIVERSITÄT DARMSTADT

No algorithms, no data structures? Such framework cannot be very helpful.

Take positive view point: How helpful can it be? We give it a try:

peewit

- does not provide *any ML engines*
- provides elementary workflow services
- services that help to *avoid a mess*
- relieves ML engines of providing such
- here
  - `http://mloss.org/`
  - `http://www.ke.tu-darmstadt.de/resources/peewit`

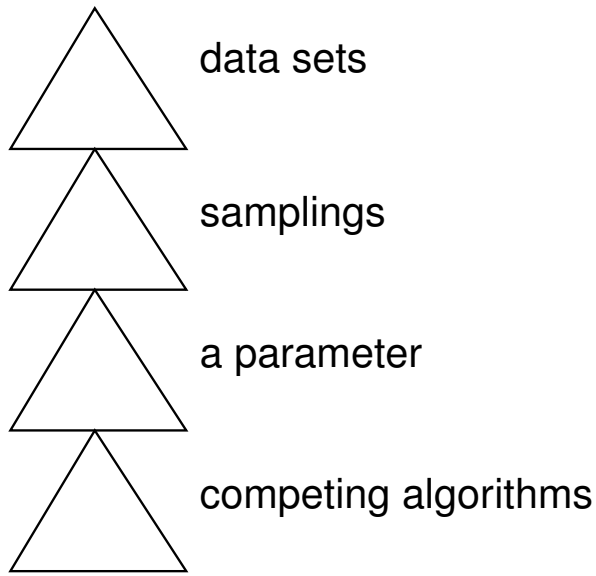
Peewit Perspective

Form of a Simple Experiment

TECHNISCHE UNIVERSITÄT DARMSTADT

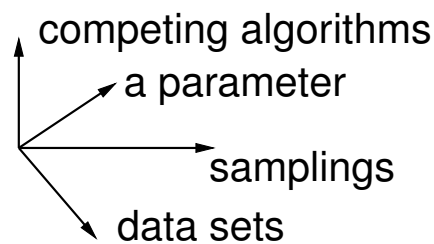
A simple experiment might look like this.

- four *user-defined* components
- each taking different *(descent) values*
- probe all combination of values



Uniformity:

- number of descent values is fixed
- values of last compound (results) can be stort in 4-dim data-cube
- algorithms use same *or corresponding* parameter
- cube element has coordinates that are abstracted descent values



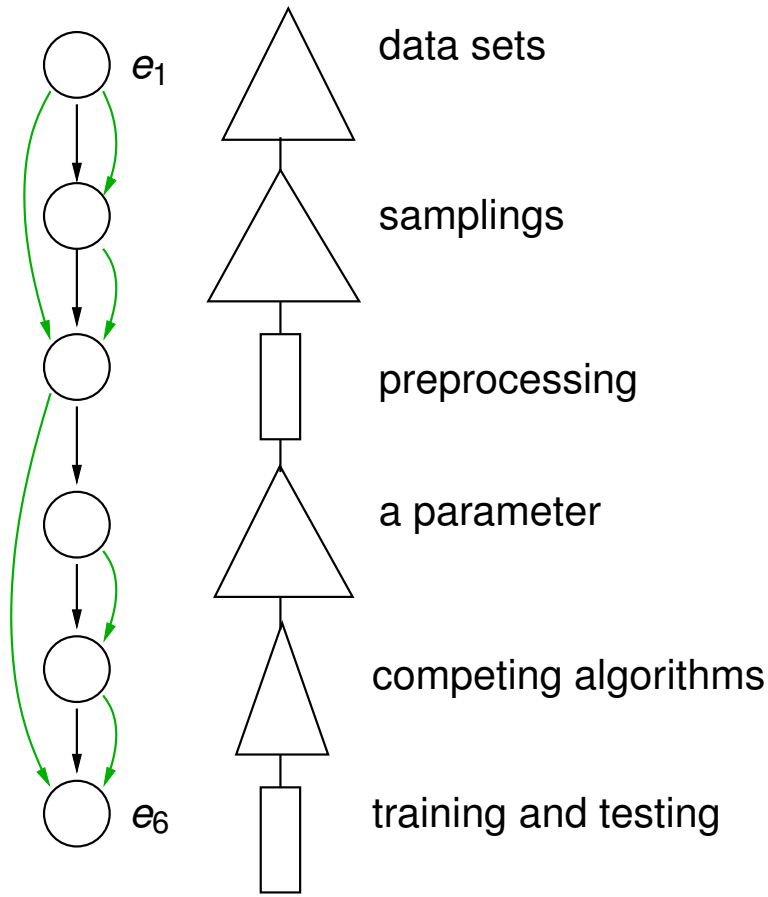
Peewit Perspective

Experiment: Chain of E-Nodes

TECHNISCHE UNIVERSITÄT DARMSTADT

experimental components: *e-nodes*

- *entirely* decompose experiment into e-nodes
- might have single descent value
- can be anything
- put e-nodes in chain (black edges) that respects dependecies (green edges)



Peewit Perspective

Dealing with Non-Uniformity

TECHNISCHE UNIVERSITÄT DARMSTADT

What if value tree is not so regular?

- compare two algorithms
- one takes a parameter the other does not

Answer:

- this not an *experiment*
- there are two similar experiments
- differing only in two e-nodes
- let peewit facilitate handling similar experiments

Peewit Perspective

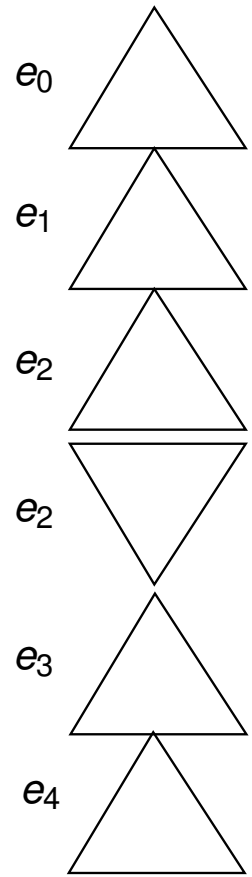
In-Line Aggregation

TECHNISCHE UNIVERSITÄT DARMSTADT

What about calibration on hold-out calibration set?

Aggregation over descent values as input for further descents:

- any e-node *e* produces *ascent value*
- computed after descent has completed
- input: child ascent values and proper descent values



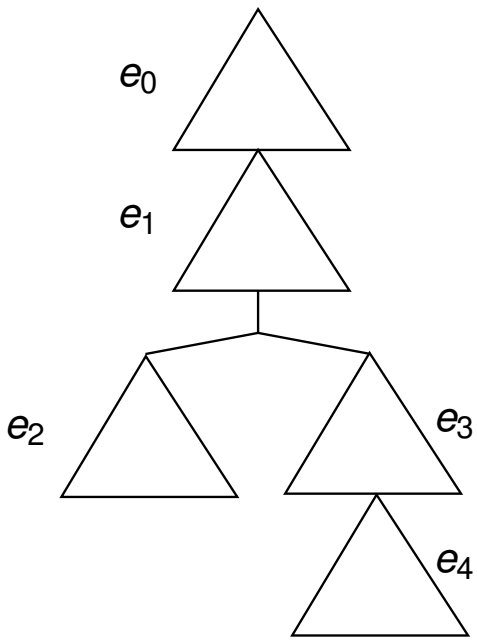
Peewit Perspective

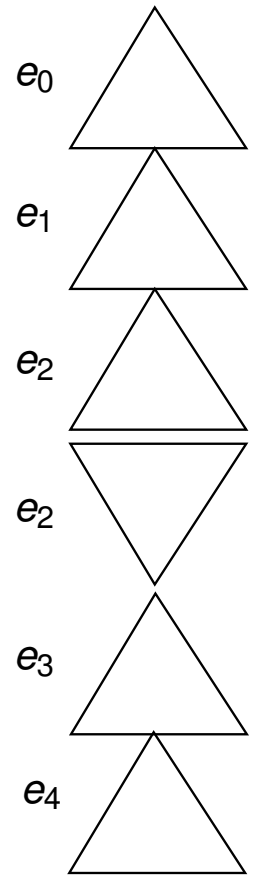
Brackets for In-Line Aggregation

TECHNISCHE UNIVERSITÄT DARMSTADT

experiment: tree of e-nodes

- descent of subtree (e_2)
- ascent of subtree
- descent of next subtree (e_2, e_4) with ascent values from former as input





Peewit Perspective

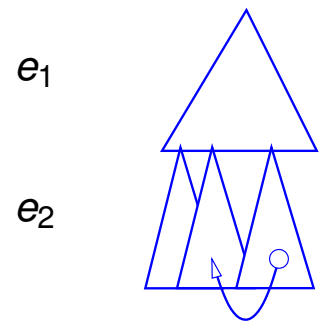
Iterative Computation

TECHNISCHE UNIVERSITÄT DARMSTADT

What about reusing computed values?

Descent production provides *sibling value*.

- next sibling descent can access it
- also next sibling of higher degree
- desired sibling is referenced trough name lowest common ancestor e-node



Peewit

TECHNISCHE UNIVERSITÄT DARMSTADT

Potential advantages:

- not restricted to specific kind of experiment
- still, machine gets grip on structure of experiment
- several usefull services possible on that basis
- concise experiment specification

Implemented services:

- name space persistence
  - user provide names
  - valid for coding, and for result queries and displays
- aggregations of result cubes

Thinkable:

- automatic parallelization on 2-20 nodes
- simple but usefull version management integration
- estimation of remaining run-time, and other

Peewit

No?

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Examples shiped with prototype explain it much better.

`http://www.ke.tu-darmstadt.de/resources/peewit`