Experimenting with rule induction algorithms in HEP data analysis

Nikita Stepanov HIP, Helsinki

(on leave from ITEP, Moscow)

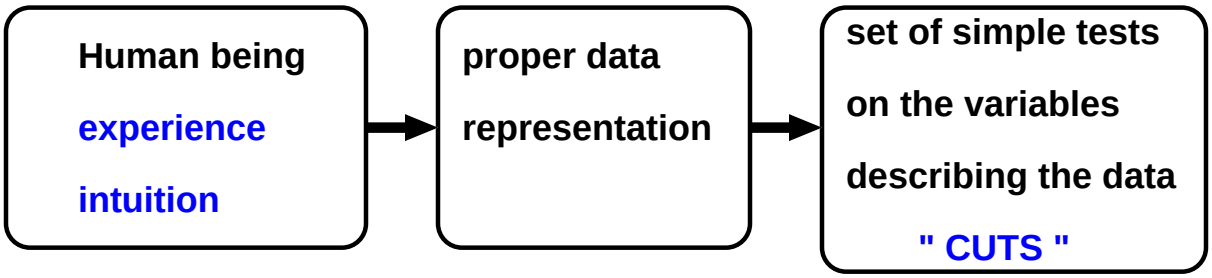
ACAT 2K, FNAL, 16-20 October- 2000

MOTIVATION

Standard HEP data analysis task:

Given data, provide the best possible signal evidence

evidence := statistics + intuition + ...



Once some useful structure in data is captured, one can think about the possible improvement using more sophisticated discriminators or classifiers. NN are proven to be very useful.

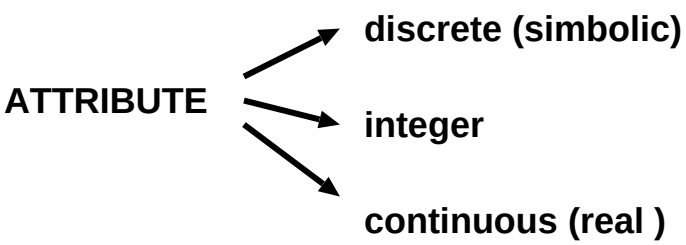
The main objection is the " black boxing " of the knowledge representation.

There are knowledge-oriented algorithms capable of developing descriptions understandable to the user directly manipulating simbolic structures, i.e. RULE INDUCTION ALGORITHMS

Rules etc

Data := vector<Example>

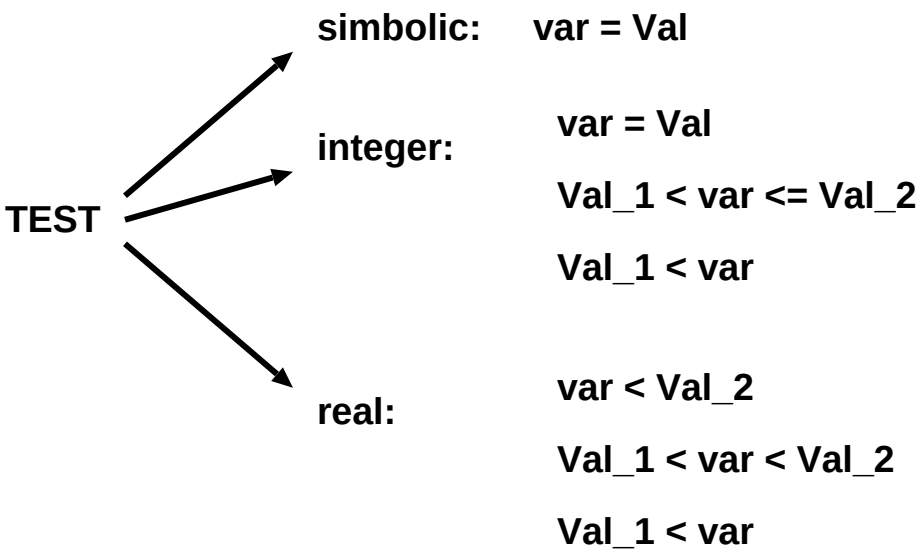
Example := vector<Attribute> + classID



RULES : 0 - order propositional (or attributional) logic

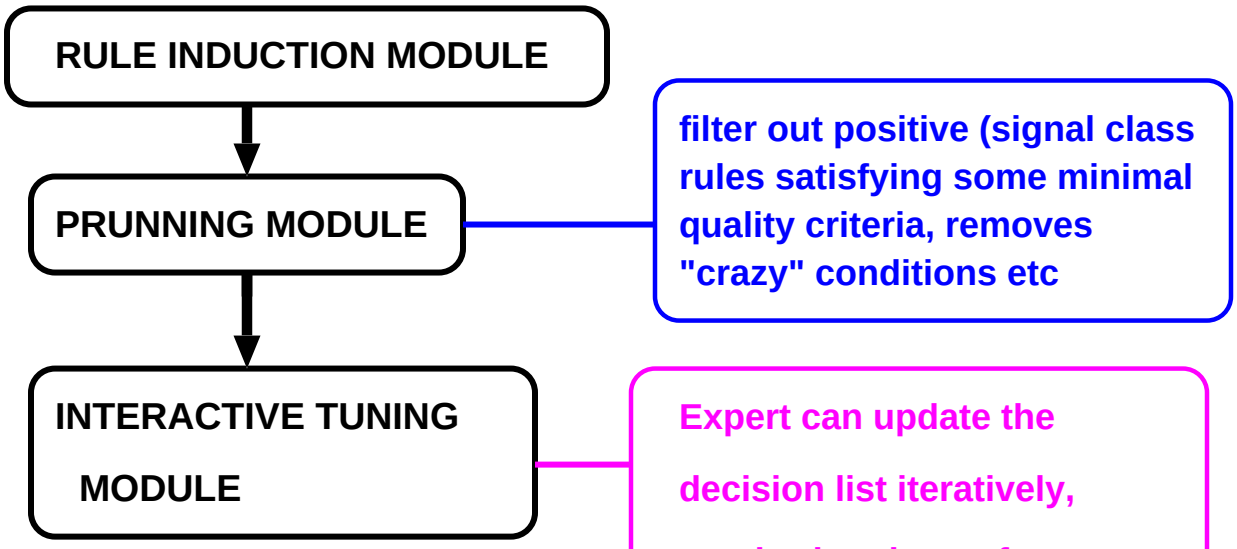
IF (COND) |-> class

Cond = Test_1 V Test2 ..... V Test_i



Decision list = ordered rule set

Induction chain



Last two modules are very task dependent. The first one is quite general.

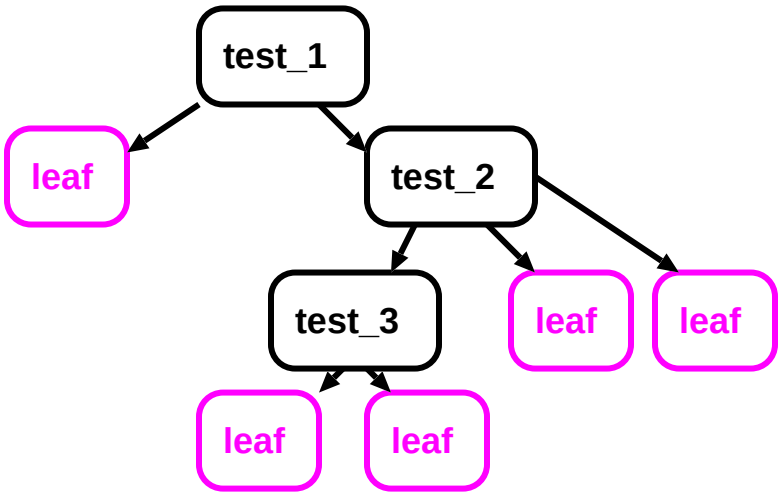
It can be any rule induction algorithm invented so far.

Divide and conquer	Separate and conquer	Genetic
(C4.5 rule)	(AQ, CN2)	Stochastic
TDIDT	Top down induction	etc
	Bottom up induction	

I experimented both with TDIDT and AQ - like algorithms. The results look quite similiar. Both have the same problems.

My present filling is that TDIDT is a bit more efficient, simpler and easy to control.

TDIDT



TDIT Algorithm

S ... the set of examples

1. **Find the "best" attribute At and the "best" test T with mutually exclusive outcomes T_i.**
2. **Split the set S into the subsets S_i, so that all examples in S_i have T_i(At) = true. Each S_i constitutes new node in DT.**
3. **For each S_i tests whether the Stopping Criterion is satisfied. If yes then create a leaf otherwise, perform the same procedure for S_i.**

Splitting heuristics used: Gain Ratio (a la Quinlan C4.5)

Stopping criterion: max Gain Ratio < GRmin || signal class probability > Pmin || number of examples in the node partition < Nmin

Splitting criterion: best pair(attribute,test) has to have at least average Info Gain and maximal Gain Ratio.

Tests on the real-valued attributes allow multiple splits

Info, gain, gain ratio etc

Given example set S, let Pi be the i-th class probability, then

$$\text{Info}(S) = -\sum_i P_i * \text{Log}_2 P_i$$

Given a split S -> {Sk}, then

$$\text{Info Gain} = \text{Info}(S) - \sum_k \frac{|Sk|}{|S|} * \text{Info}(Sk)$$

for the continuous attribute the Info Gain is reduced by $\text{Log}_2 (N-1)/|S|$ where N is the number of different attribute values

$$\text{Split} = \text{Log}_2 |S| - \sum_k \frac{|Sk|}{|S|} * \text{Log}_2 |Sk|$$

$$\text{Gain Ratio} = \frac{\text{Info Gain}}{\text{Split}}$$

Case study

TASK:= improve b-tagging performance for the particular channel:

PP -> tt (H -> bb) -> l, 4b, jj + Et_miss

Require ultimate detector performance, in particular, b-tagging efficiency ~ 50-60% (four b-tagged jets!)

Main (about the only one) background is tt + X

Event samples:

ttH and tt+X events generated by PYTHIA with at least 6 jets with Et > 15 GeV and |eta| < 2.4 and isolated lepton with Pt > 15 GeV were selected.

To learn the classifier, three classes of jets were introduced: b-jet (jet with the b-quark in a core dR < 0.2), c-jet and gq jets. 5000 signal events and 7000 background ones

Detector response and reconstruction:

Fast MC for the Calorimetry, detailed GEANT simulation for the Tracker; track finding and vertex finding using dedicated CMS sw packages.

Lepton, jets, missing Et, all reconstructed tracks and vertices were stored event per event for learning and testing.

Minimal track selection criteria: Pt > 0.7 GeV; chsq / ndf < 5; at least 6 hits / track and 2 pixel hits / track

Example structure

track significance: SIG = | tr. ip | / err_tr_ip, cos(ip, Pt_jet) > 0

secondary vertex significance SIV defined similarly

Only the tracks with SIG > 2 and | tr. ip | < 2 mm were selected

EXAMPLE (i.e. jet)

classID = 0, 1, 2

Et (integer)	1 : Et < 30 GeV; 2 : 30 - 60; 3 : 60 - 120; 4 : 120 - 200; 5 : 200 < Et
$\|\eta\|$ (integer)	1 : |eta| < 0.7 (barrel); 2 : 0.7 - 1.2; 3 : 1.2 - 1.6; 4 : 1.6 - 2.0; 5 : 2.0 - 2.4
NTR (integer)	number of " good " tracks in a cone 0.4
TSS (real)	total significance of associated tracks
TPT (real)	total Pt
MST (real)	effective mass of associated tracks
NSV (integer)	number of secondary vertices associated with a jet
TSV (real)	total vertices significance
MSV (real)	effective mass of all tracks trapped

In total 8 attributes, most of them strongly correlated

Strategy and features

- 3-fold cross validation test : 1000 + 1000 examples to learn TDIDT; 2000 + 3000 to evolve and tune decision list; 2000 + 3000 to validate. Best system was tested using whole set.

  Yet statistics is too small for the total test including mass reconstruction

- TDIDT was learned just to classify jets.

  No surprises - no specific tests to distinguish b and c jets

  Best results were obtained with just two classes.

  Special care needed to avoid ambiguties:

  if (BAD_FOR_SURE(example)) exampleClass = 1 (gq)

  Still problem with continuous attributes

- PRUNNING: rather trivial (more sophisticated did not improve)

- INTERACTIVE TUNING: user monitors the overall b-tagging performance and purity but also the number of 4-tagged signal and background events

Results

selected signal events contain just ~ 0.529 " good " ones

i.e. events with at least 4 b-jets

Performance monitoring:

BE (probability to b-tagg b-jet (per jet))

GQI (probability to b-tagg gq - jet)

CI (probability to b-tagg c-jet)

ES4 (fraction of " good " signal events 4-tagged)

EGL (fraction of 4-tagged signal events)

TTI (fraction of 4-tagged tt background events)

RESULTS

"Straw man" (NTR > 1 V NSV > 0)

BE = 0.569 ES4 = 0.1446

GQI = 0.0194 EGL = 0.089

CI = 0.144 TTI = 0.00392

TDIDT (leaf votes b-class provided Pb > 0.9 and Nex > 10)

BE = 0.5878 ES4 = 0.172

GQI = 0.01751 EGL = 0.104

CI = 0.1436 TTI = 0.00249

Ugly! (no prunning)

~ 150 nodes

~ 80 leafs

Decision list (averaged over all 3 tests)

BE = 0.623 ES4 = 0.2174

GQI = 0.0198 EGL = 0.134

CI = 0.163 TTI = 0.0032

Best decision list

R1 : NTR > 1 V TPT > 4 V TSV > 3
R2 : NTR > 2 V TPT > 8 V TSS > 8
R3 : NTR > 1 V TSV > 2 V MSS > 1
R4 : NTR > 1 V TPT > 15 V TSS > 20

with 4 rules: BSE = 0.584 ES4 = 0.1588
GQI = 0.0160 EGL = 0.0975
CI = 0.141 TTI = 0.00143

R5 : NTR > 1 V TPT > 9 V TSS > 9 V ET <= 3 V ETA <= 2
R6 : NTR > 1 V TPT > 5 V TSS > 7 V NSV > 0 V ET <= 2
R7 : NTR > 0 V TPT > 1 V TSV > 6 V NSV > 1 V ET <= 2
R8 : NTR > 1 V TPT > 5 V TSS > 7 V ET = 1
R9 : NTR > 1 V TPT > 4 V MSS > 0.5 V ET = 1 V ETA <= 3

with 9 rules: BSE = 0.606 ES4 = 0.191
GQI = 0.0194 EGL = 0.118
CI = 0.154 TTI = 0.00214

R10 : NTR = 1 V TSV > 6 V 0.8 < MSV < 3 V TPT > 3 V ET <= 3 V ETA <= 2

R11 : NTR = 1 V TSV > 8 V TPT > 5 V ET <= 3 V ETA <= 2

R12 : NTR = 1 V TPT > 2 V TSV > 5 V TSS > 6 V ET <= 4
V ETA <= 2 V 1 < MSV < 5

R13 : NTR = 1 V TSV > 6 V TPT > 8 V ET <= 3 V ETA <= 3

PERFORMANCE: BSE = 0.628 ES4 = 0.220
GQI = 0.0205 EGL = 0.136
CI = 0.167 TTI = 0.00356

In conclusion

\+

In total, the algorithm behaves like expected:

no dramatic discoveries, but rather sizable perfomance improvement.

It is fast (concerning non interactive part).

The case choosen is difficult, however the rules "discovered" are simple and understandable.

—

At the moment the algorithm relies too heavily on the human help at the prunning stage, it creates additional "subjective" bias. For me, it's some how unavoidable - general prunning algorithms are too general and very computationally expensive.

The hybrid model which can accommodate the background domain knowledge is under the development.

?

I touched just the tiny fraction of this ML domain.

A lot of room for the further development!

Physics is indeed simple and internally logical.