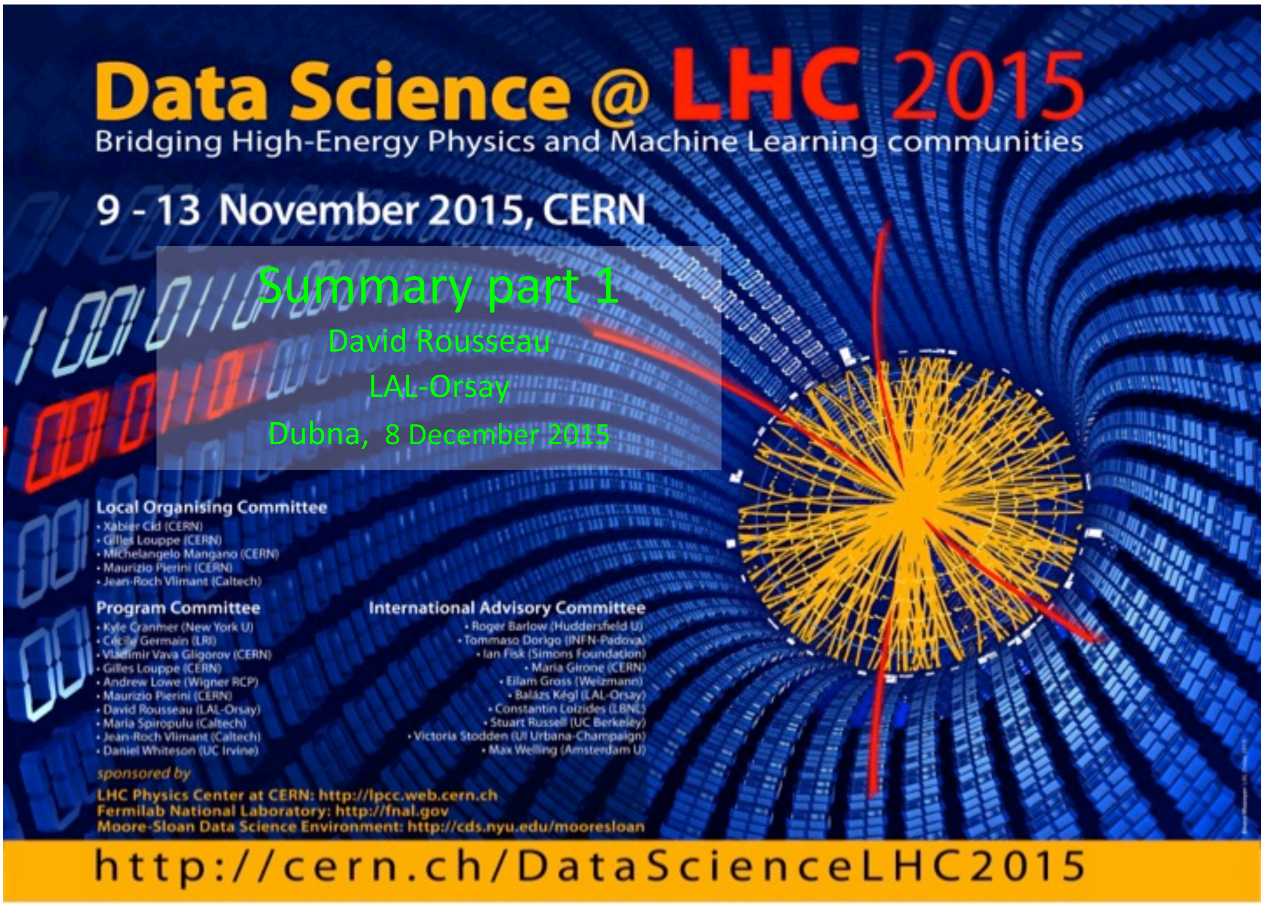
Summary part 1

David Rousseau

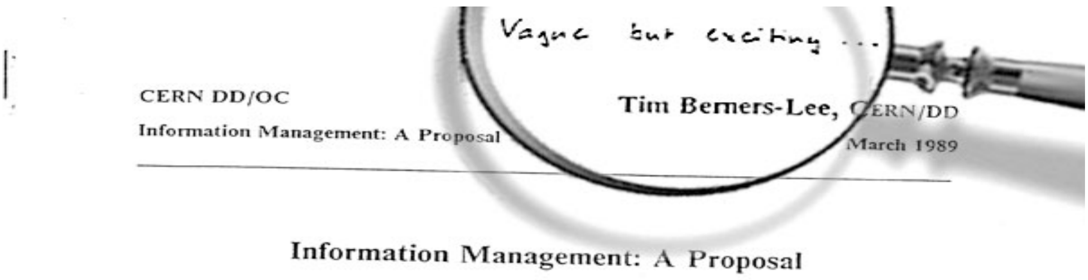
LAL-Orsay

Dubna, 8 December 2015



Context

- We (High Energy Physics) have been doing Big Data/Data Science for decades without knowing it
- then…



Vague but exciting ...

CERN DD/OC

Tim Berners-Lee, CERN/DD

Information Management: A Proposal

March 1989

Information Management: A Proposal

- …then Google, Amazon, Facebook, Yandex…➔big data, big money➔big incentive to develop new algorithms
- ➔we need to catch up!

Foreword

- http://cern.ch/DataScienceLHC2015, transparencies and video (and twitter #DSLHC15)
- Different types of talks
  - HEP talk geared at informing ML people
  - ML talk by ML people
  - ML talk by HEP people
  - Sometimes “answer” from the other community
  - Tutorials
  - Entertaining talks (with no practical direct application to HEP)
- Overall quite dense : this summary by Dirk and I is more an invitation to further reading

Tutorials

Afternoon was tutorials. Some really hands-on (not enough IMHO)

- Monday : TMVA (including new iPython interface)
- Tuesday : ME (Madweight, MemTK)
- Wednesday : deep learning
- Thursday : Scikit-learn (including interface to TMVA through R
- Friday : Caffee (Convolutional Neural Net on GPU)

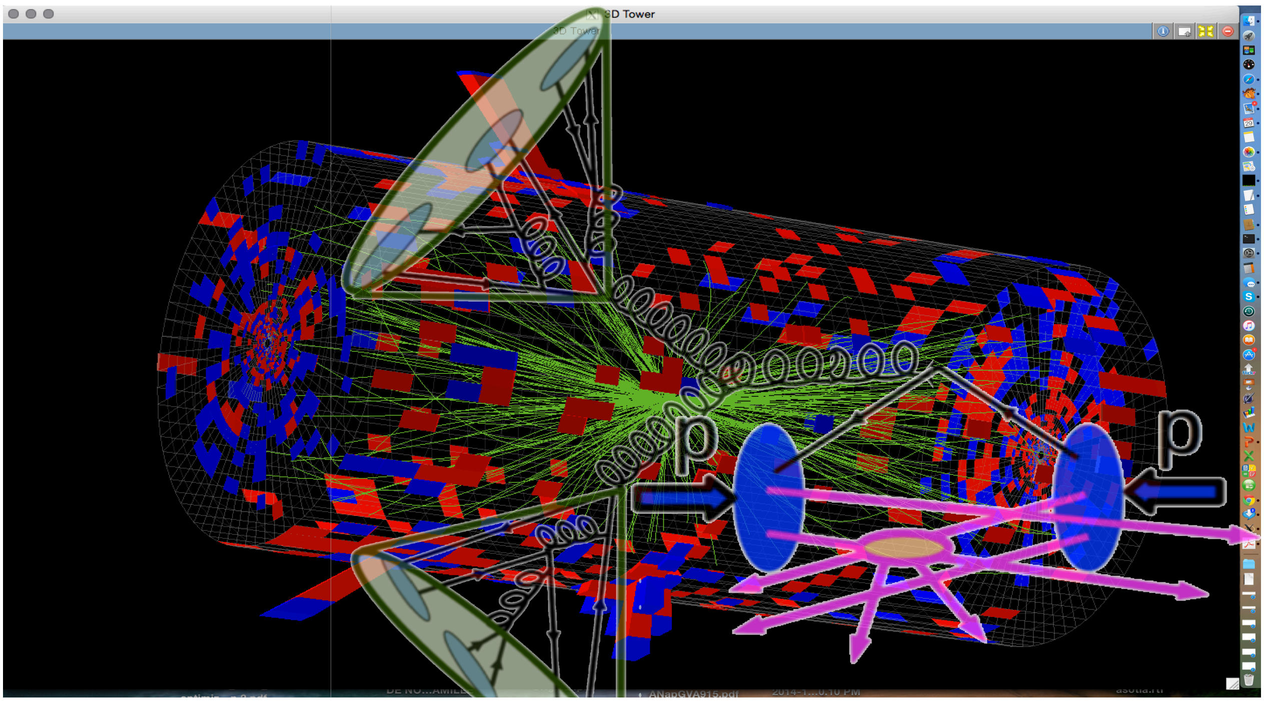
David Rousseau, DS@LHC2015 summary part 1, Dubna, 8 Dec 2015

Monday

- Data and Science in HEP: Vincenzo Innocente
  - #HEP2ML Excellent introduction talk for non HEP people
- Data Science in industry : Ellie Dobson
  - Ellie was CERN fellow in ATLAS. Data Science as a job opportunity for HEP PhD. #entertaining
- ML at ATLAS&CMS : setting the stage : Mauro Donega, Preparing for the future: opportunities for ML in ATLAS & CMS : Tobias Golling
  - #HEP2ML and #MLbyHEP Two talks setting the stages
- Deep Learning RNNaissance : Juergen Schmidhuber
  - #entertaining Historical perspective
- Feature extraction : Sergei Geyzer
  - #MLbyHEP How to select the relevant variables(==features)

Data and Science in HEP: V. Innocente

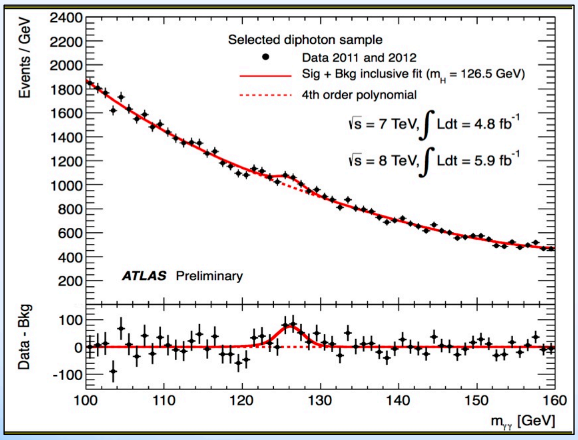
- Excellent introduction to HEP (non distributed) data processing for Machine Learning people



Data Science in industry : E. Dobson

- Experience of an ex Cern fellow on Atlas in Data Science industry. Data Science now a good opportunity for HEP PhD leaving academics.

Theory-driven approach



‘start with the system and work towards the data’

Data-driven approach

‘start with the data and work towards the system’

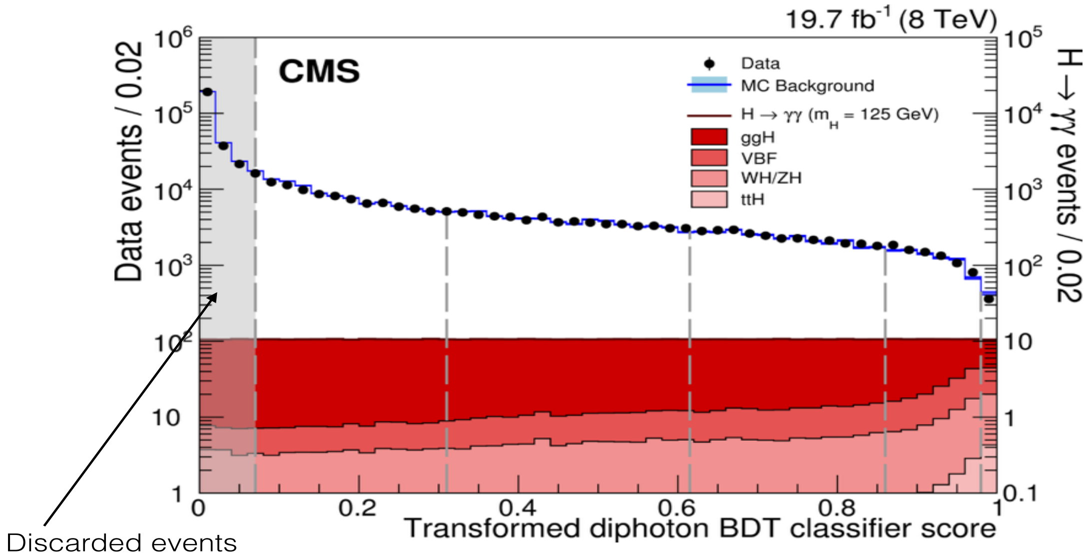
ATLAS&CMS : setting the stage : M. Donega

- What we’ve used ML so far in ATLAS/CMS:
  - Pattern recognition: clustering pixels
  - Tracks classification: duplicate removal, quality selection,…
  - Energy / momentum regressions: photons, electrons, (b-)jets,…
  - Objects identification: select electron, b/c-jet,… form (typically jets) background
  - Entire event classification: separate signal from background(s) events
  - Fisher discriminant, Likelihoods, Neural Networks, BDT, 1D/2D fit MVA outputs
  - Data placement: predict which samples will become hot
  - The vast majority of these application moved from “cut-based” solutions to supervised learning techniques (unsupervised learning at present not used)

Donega (2)

BDT output

Number of classes (5) and boundaries chosen to optimize the S/B.
(discard events in the lowest score bin)



Transformed such that the sum of the signal components is flat

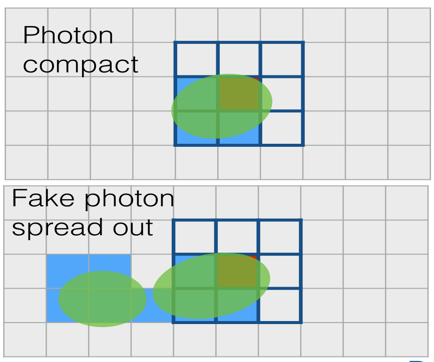
Donega (3)

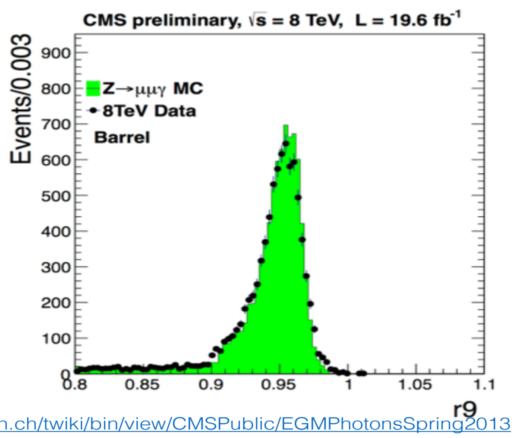
Systematic uncertainties on inputs

Systematics uncertainties typically lead to non-optimal classification/regression.
We know how to set a systematic on the input variables but don't have a standard recipe to assign systematics to BDT outputs.

Example: photon identification
BDT classifier to separate photons from fake photons i.e. jets ($\pi^0 \rightarrow \gamma\gamma$)
o(12) input variables, some of which are correlated, mostly describing the shape of the calorimeter cluster

Use physics driven features not full information





Ref. https://twiki.cern.ch/twiki/bin/view/CMSPublic/EGMPhotonsSpring2013

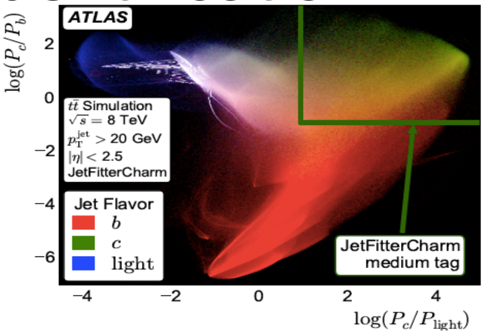
Preparing for the future: opportunities for ML in ATLAS & CMS : T. Golling

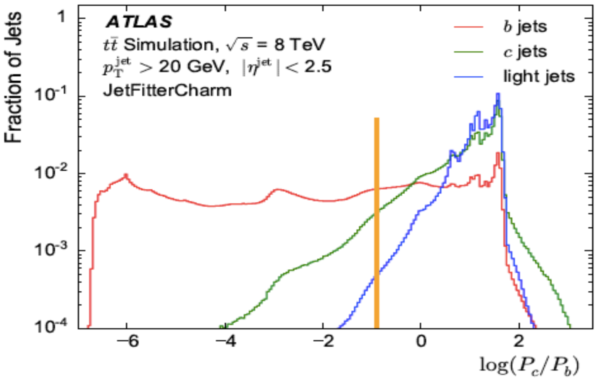
ATL-PHYS-PUB-2015-001

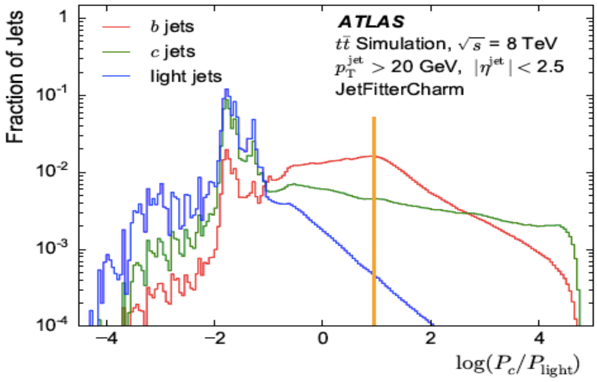
Example: charm-jet Identification

- Define 2 discriminants based on 3 NN outputs:

$$\text{anti-}b \equiv \frac{P_c}{P_b} \qquad \text{anti-light} \equiv \frac{P_c}{P_{\text{light}}}$$





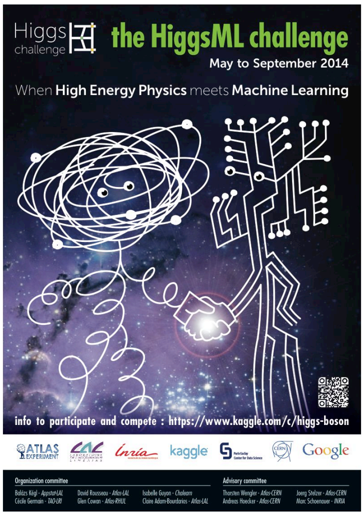


Golling (2)

Loose ends: HEP Particularities

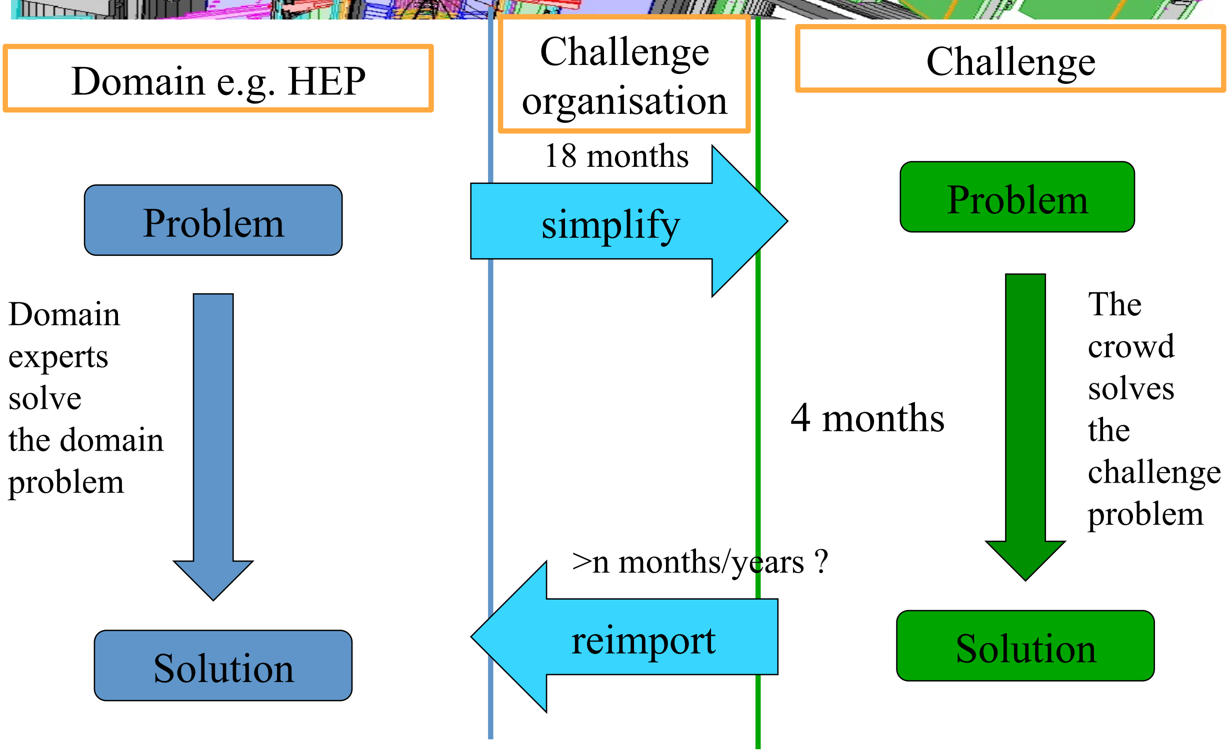
- Mismodeling: data vs. simulation
  - Systematic uncertainties based on mismodeling uncertainty
  - The better the classification the larger the deviation (showstopper, e.g. photon ID)
  - (Limited) possibility to validate and calibrate MC to data
- In MC we use data with a large variation in relative weights / neg weights – problems for training
- Variable-length / non-continuous input feature phase space
- We usually have a model based on our physics knowledge – this leads to two extreme approaches:
  - Matrix Element Method (MEM): rely on “calculable” part of model
  - ML: let machine learn (still model dependence)
  - MEM pros & cons:
    - Pros: no need to train, no need for large statistics, make us of maximum available information
    - Cons: slow for complex final states, many approximations/simplifications of the model needed
- **Can we combine ML and physics input in a smart way?**
- Features may vary significantly e.g. with p_T or eta (analogy: facial expressions in face recognition)

Higgs ML challenge 2014



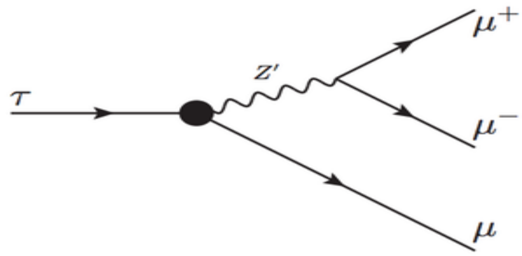
- (started DR meeting Balazs Kegl data scientist at LAL-Orsay cafeteria summer 2012)
- Why not put some ATLAS simulated data on the web and ask data scientists to find the best machine learning algorithm (=MVA) to find the Higgs ?
  - Instead of HEP people browsing machine learning papers, coding or downloading possibly interesting algorithm, trying and seeing whether it can work for our problems
- Challenge for us : make a full ATLAS Higgs analysis simple for non physicists, but not too simple so that it remains useful
- Also try to foster long term collaborations between HEP and ML
- http://jmlr.org/proceedings/papers/v42/

From domain to challenge and back



LHCb : Flavour of physics challenge

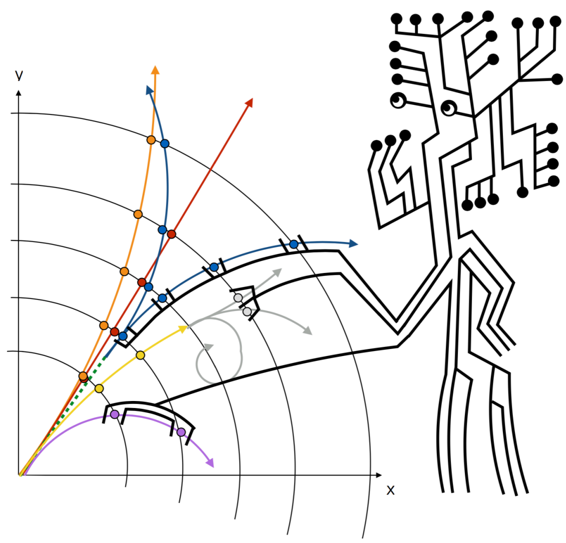
- Wrt HiggsML similar optimisation of significance of a rare signal
- New ingredient : handle data/MC mismodeling



- Turned out to be even more tricky than anticipated

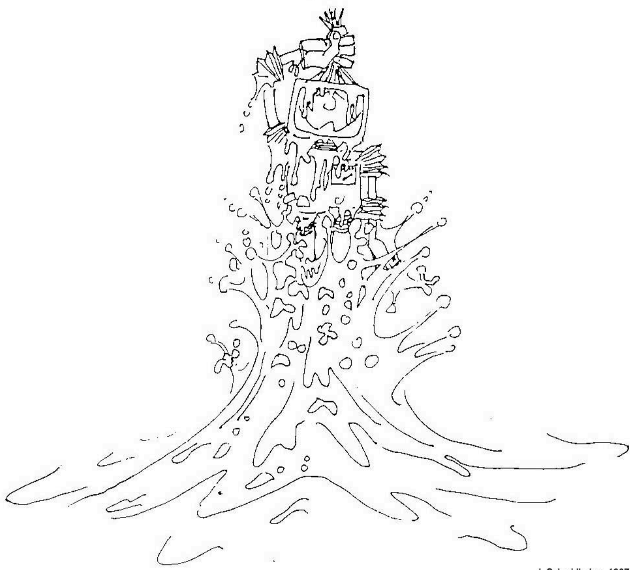
Tracking challenge ?

- Fast and efficient tracking more and more essential as LHC luminosity increases➔tracking challenge?
- Trickier to organise than HiggsML or the like:
  - less “on-the-shelf” algorithms than for classification
  - Figure of merit combination of efficiency/fake rate/CPU time
  - CPU time to be measured in a well defined way
- Goal is to go online in summer 2016



Deep Learning RNNaissance : Juergen Schmidhuber

- Historical perspective



J. Schmidhuber, 1987

My diploma thesis (1987): first concrete design of recursively self-improving AI

Learn & improve learning algorithm itself, and also the meta-learning algorithm, etc…

Schmidhuber (2)

Robot Cars

http://www.idsia.ch/~juergen/robotcars.html

1995: Munich to Denmark and back on public Autobahns, up to 180 km/h, no GPS, passing other cars

Ernst Dickmanns, *the* robot car pioneer, Munich, 80s

2014: 20 year anniversary of self-driving cars in highway traffic

Feature extraction : S. Gleyzer

- While performing data analysis one of the most crucial decisions is which features to use
  - Garbage In = Garbage Out
  - Ingredients:
    - Relevance to the problem
    - Level of understanding of the feature
    - Power of the feature and its relationship with others
- How to:

Select

Assess

Improve

Feature set

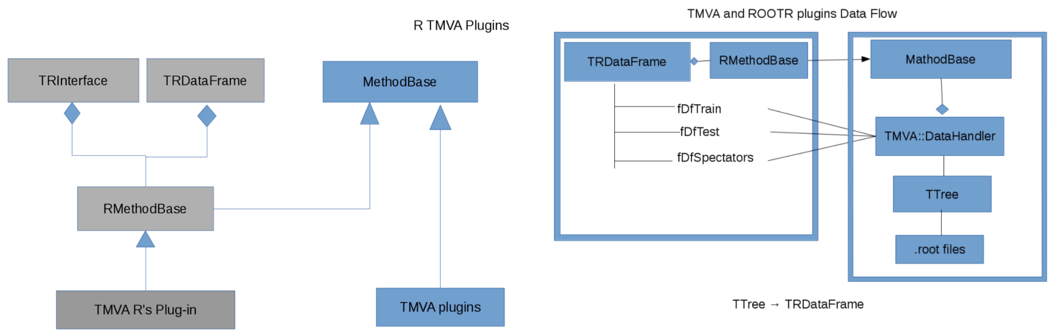
used to solve the problem

Gleyzer (2)

- ...reviewed various tools
- Often in HEP one searches for new phenomena and applies classifiers trained on MC for at least one of the classes (signal) or sometimes both to real data
  - Flexibility is KEY to any search
  - It is more beneficial to choose a reduced parameter space that consistently produces strong performing classifiers at actual analysis time
- Feature selection tool
  - R (CRAN): Boruta, RFE, CFS, Fselector, caret
  - TMVA: FAST algo (stochastic wrapper), Global Loss function
  - Scikit-Learn
  - Bioconductor

TMVA tutorial

- ❑ TMVA is the workhorse ML used in HEP
- ❑ As been somewhat left behind
- ❑ Rejuvenated effort since last summer, for example, interface to R (hence to outside ML world)
- ❑ iPython interface



Tuesday

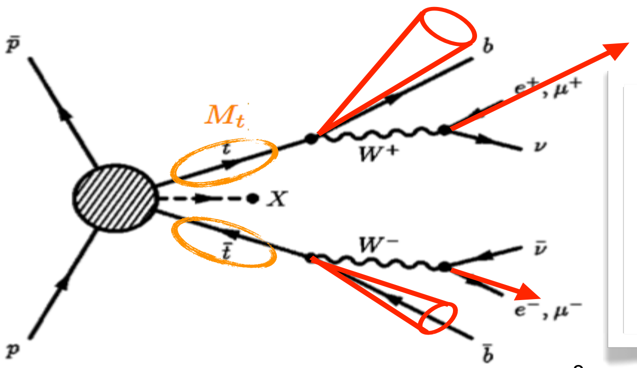
- ❑ Matrix Element technique plus experience ttH : Lorenzo Bianchini
  - ○ #HEP2ML ME is not Machine Learning. Why ME in this workshop ? Why don't we through all 4-vector to a BDT/NN and let it figure out the physics ? Won't work. However possibility for a mixed approach: use ME output as a feature
- ❑ Approximate Bayesian Computation : Richard Wilkinson
  - ○ #ML2HEP ABC widely used outside HEP, little in HEP, probably because we have quite good simulation suite (generators+geant4). Still possible niches, see Josh Bendavid answer
- ❑ Approximate likelihood : Kyle Cranmer
  - ○ #MLbyHEP
- ❑ Stochastic optimization : beyond mathematical programming : Marc Schoenauer
  - ○ #ML2HEP Review of optimisation method for chaotic landscape, of high dimensionality (where Minuit fails)
- ❑ Software R&D for Next Generation of HEP Experiments, Inspired by Theano : Amir Farbin
  - ○ #MLbyHEP Theano : python based symbolic representation and operations, optimized calculation on CPU's and GPU's. Tried out for MEM calculation. New non LHC HEP experiment (e.g. Dune) : tried out DNN reco
- ❑ Better cities through imaging : Gregory Dobler
  - ○ #entertaining : "One picture every 10s of Manhattan skyline for two years". "Video of a busy road crossing". What can you do with this ? A lot!

Matrix Element technique plus experience ttH : L. Bianchini

- ME : compute event probability from first principle (zero "learning"!, but very human/computer time intensive)

Canonical example: top mass

Y = 4-vectors of jets and leptons
M_t = top quark mass
X = 4-vectors of the 2 → 6 scattering (ancillary variables)
W(**Y**|**X**) = detector response



Matrix Element Method
(MEM)

$$dP_X(M_t) \propto f_p(x_1) f_{\bar{p}}(x_2)\, |\mathcal{M}(X; M_t)|^2\, dX$$

$$dP_{Y|X} = W(Y|X)dY$$

$$dP_{X,Y} \propto dP_{Y|X} \times dP_X(M_t)$$

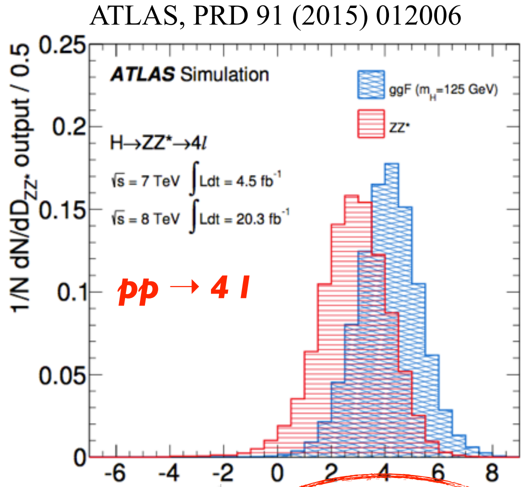
$$dP_Y(M_t) = \left[\int dP_X(M_t) \times W(Y|X)\right] dY$$

9

Tuesday, November 10, 15

Bianchini (2)

ATLAS, PRD 91 (2015) 012006



$$D_{ZZ^*} = \ln\left(\frac{|\mathcal{M}_{\mathrm{sig}}|^2}{|\mathcal{M}_{ZZ}|^2}\right)$$

Bianchini (3)

Summary & outlook

A field where ML can have some complementarity

- higher-order predictions difficult to integrate into the MEM
  - LO vs NLO, parton shower, transfer function
- ML can help where MEM falls short
  - several examples already exist
- squeezing every bit of information out of LHC data is our mandate!

Approximate Bayesian Computation : R. Wilkinson

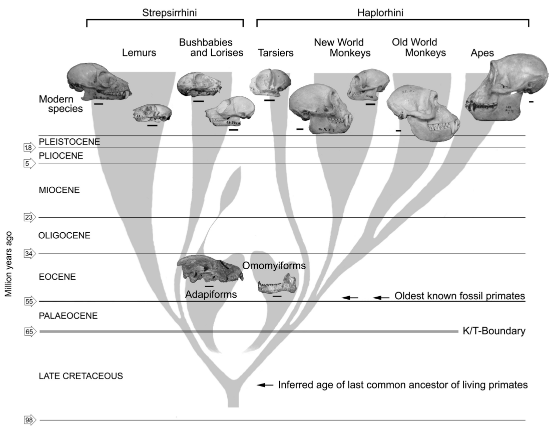
- Introductory course on ABC
- We have a theory/model with parameters θ, we perform experiments yielding data D

- The inverse-problem: observe data D, estimate parameter values θ which explain the data.

The Bayesian approach is to find the posterior distribution

$$\pi(\theta|D) \propto \pi(\theta)\pi(D|\theta)$$

posterior $\propto$ prior $\times$ likelihood



How to evaluate π(θ|D)?

Wilkinson (2)

Rejection Algorithm

- Draw θ from prior $\pi(\cdot)$
- Accept θ with probability $\pi(D \mid \theta)$

Accepted θ are independent draws from the posterior distribution, $\pi(\theta \mid D)$.
If the likelihood, $\pi(D|\theta)$, is unknown:

'Mechanical' Rejection Algorithm

- Draw θ from $\pi(\cdot)$
- Simulate $X \sim f(\theta)$ from the computer model
- Accept θ if $D = X$, i.e., if computer output equals observation

there is an approximate version:

Uniform Rejection Algorithm

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(D, X) \leq \epsilon$

…many more flavours and tricks

Wilkinson
…actually best summarized by J. Bendavid

Some Important Points to Keep In Mind

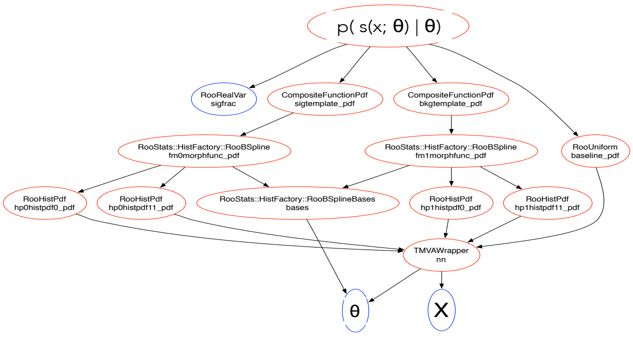
- Tempting to map "computer model" $f(\theta)$ from Richard's talk to ATLAS/CMS full generation + simulation + reconstruction chain
- Worst case scenario: Evaluating metric distance for each set of parameter values requires generating $O(10^6)$ full-sim MC events (tens of thousands of CPU hours)
- A few possible ways this kind of technique can still be useful:
  - Unfold data to generator level (or similarly produce generator $\rightarrow$ reconstructed level response matrices which can be applied quickly to generator level MC)$\rightarrow$
  - Extract reduced set of parameters from data using one or a few full Monte Carlo samples, then perform ABC-type method with a much simpler model (e.g. Bayesian integration over Higgs couplings in Higgs combination)
  - Realize model parameter variations as **reweighting** of one or a few full Monte Carlo samples

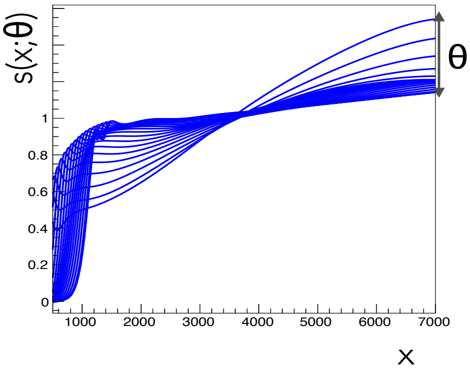
Approximate likelihood with parameterised classifier : K. Cranmer

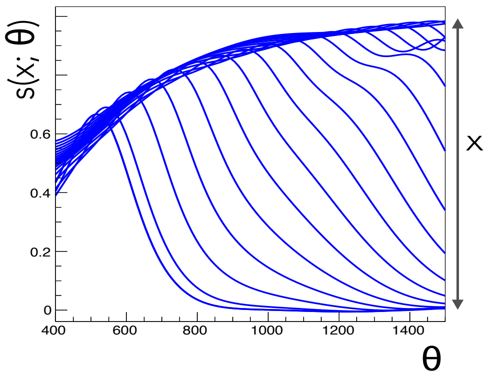
EMBEDDING THE CLASSIFIER IN THE LIKELIHOOD

Postpone evaluation of the classifier to the time when the likelihood is evaluated and a specific value of the parameter θ is being tested

$$T(D;\theta_0,\theta_1) = \prod_e \frac{p(x_e|\theta_0)}{p(x_e|\theta_1)} = \prod_e \frac{p(s(x_e;\theta_0,\theta_1)|\theta_0)}{p(s(x_e;\theta_0,\theta_1)|\theta_1)}$$



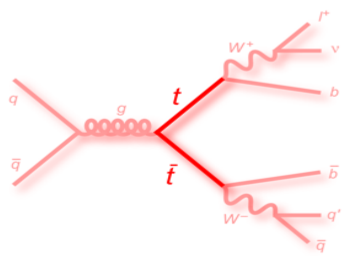




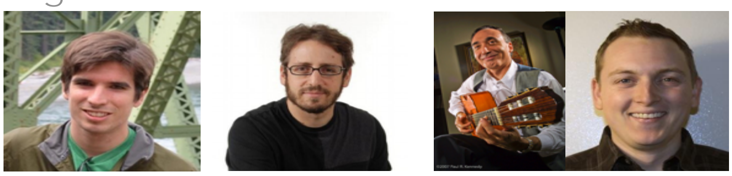
Cranmer (2)

PARAMETRIZED CLASSIFIERS WITH DNN

Example: $Z' \to t\bar{t}$

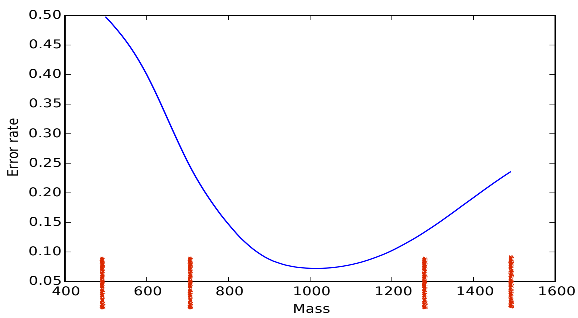


together with:



Peter Sadowski , Daniel Whiteson, Pierre Baldi, Taylor Faucett

The networks were trained on 28 features: 22 low-level, 5 high-level, and the mass



Train at $m_{Z'}$=500,750,1250,1500 GeV

Almost identical performance to dedicated training at $m_{Z'}$=1000 GeV

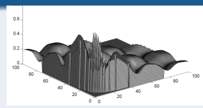
Stochastic optimization: M. Schoenauer

- Thorough review of different algorithms

Stochastic Optimization

Hypotheses

- Search Space Ω — with some topological structure
- Objective function $\mathcal{F}$ — assume some weak regularity



Stochastic (Local) Search

- Randomly draw $\boldsymbol{x}_0 \in \Omega$ and compute $\mathcal{F}(\boldsymbol{x}_0)$ — Initialisation
- Until(happy)
  - $\boldsymbol{y} =$ Random neighbor$(\boldsymbol{x}_t)$ — neighbor structure on Ω
  - Compute $\mathcal{F}(\boldsymbol{y})$
  - If $\mathcal{F}(\boldsymbol{y}) \succ F(\boldsymbol{x}_t)$ then $\boldsymbol{x}_{t+1} = \boldsymbol{y}$ — accept if improvement
    else $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$

Comments

- Find one *close* local optimum — defined by neighborhood structure
- Iterate, leaving current optimum — Iterated Local Search

Schoenauer : summarised by A. David

Stochastic methods

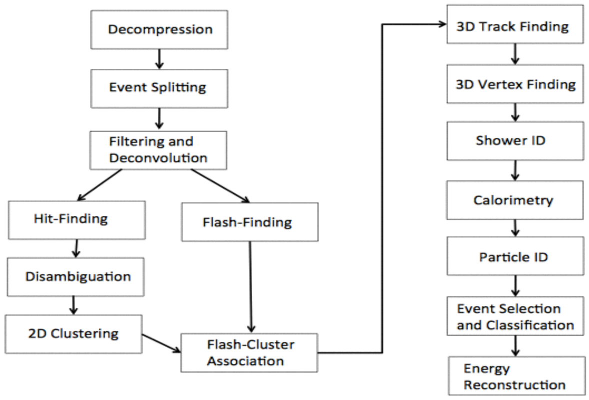
5

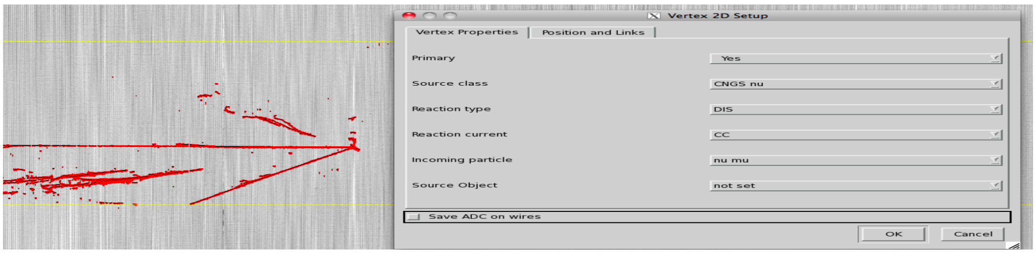
- Guaranteed to converge to best answer…
  - …in infinite time.
- Quickly get a "good enough" answer.
  - Useful in time-constrained systems (L1 or HLT?).
- Robust minimum vs. absolute best.
  - Useful in optimization of analyses with many systematic uncertainties.
- ATLAS+CMS Higgs 4000 parameter likelihood.
  - Is there something as accurate as MINUIT but faster?

- DR: pattern recognition ? data placement ?

LArTPC Reconstruction

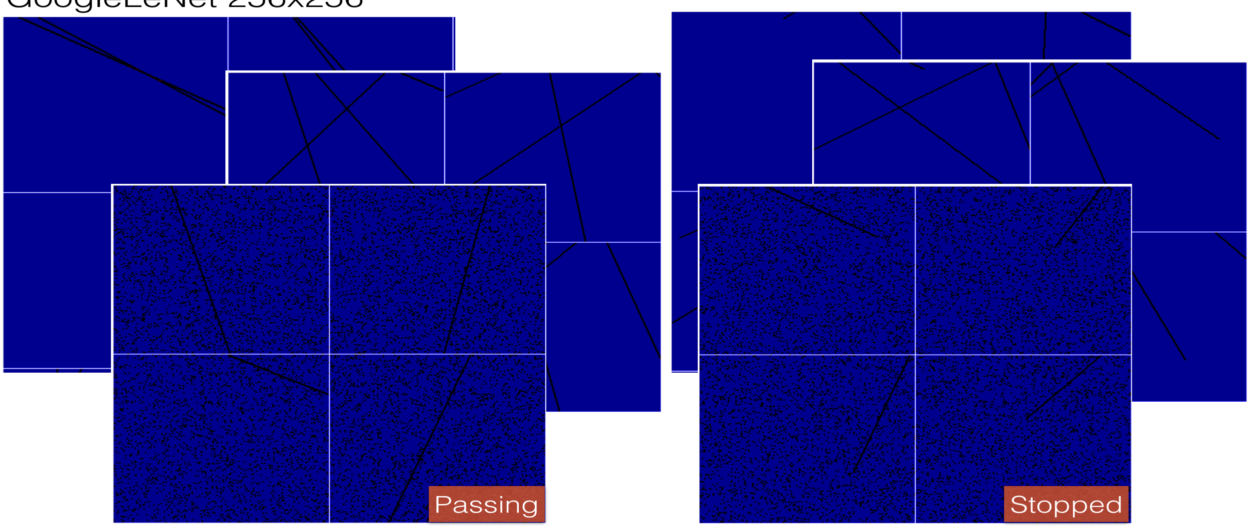
- Neutrino Physics has a long history of *hand scans*.
  - QScan: ICARUS user assisted reconstruction.
- Full automatic reconstruction has yet to be demonstrated.
  - LArSoft project: art framework + LArTPC reconstruction algorithm, started in ArgoNeuT and contributed to/used by many experiments.
- Ideally suited for DNN-based reconstruction
  - Just need to know what type of event (classification) and the energy of the neutrino (regression).





DNN Classification of "Raw" LArTPC Data

GoogleLeNet 256x256



1-4 Tracks With or without noise, DNN correctly classifies ~90-99%

Farbin (3)

- Theano : optimised symbolic computation in python

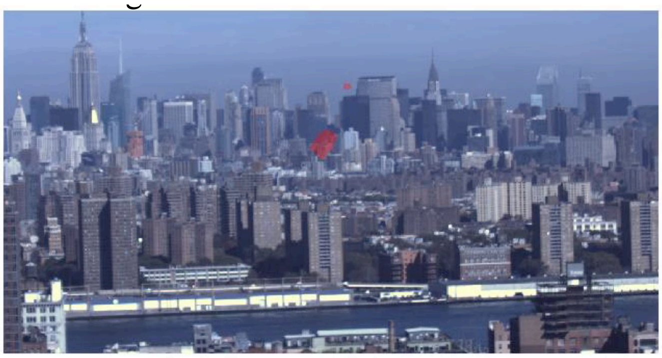
Theano

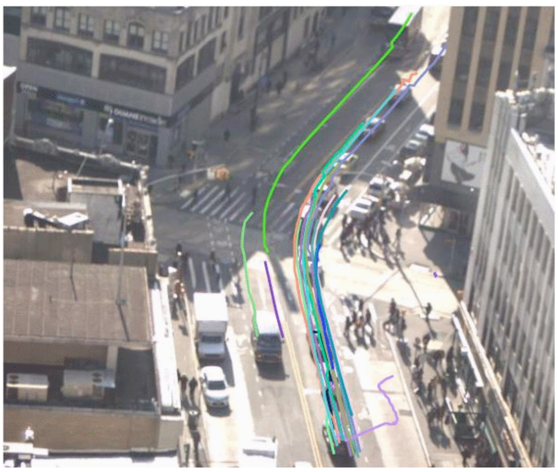
- Might be trivial to implement some algorithms with Theano.
  - Anything you can write as a formula can be easily expressed in Theano and automatically optimized.
  - Many things are already implemented.
  - For example, Kalman Filter (from: http://matthewrocklin.com/blog/work/2013/04/05/SymPy-Theano-part-3/)

```
from sympy import MatrixSymbol, latex
n       = 1000                       # Number of variables in our system/current stat
e
k       = 500                        # Number of variables in the observation
mu      = MatrixSymbol('mu', n, 1)   # Mean of current state
Sigma   = MatrixSymbol('Sigma', n, n) # Covariance of current state
H       = MatrixSymbol('H', k, n)    # A measurement operator on current state
R       = MatrixSymbol('R', k, k)    # Covariance of measurement noise
data    = MatrixSymbol('data', k, 1) # Observed measurement data

newmu   = mu + Sigma*H.T * (R + H*Sigma*H.T).I * (H*mu - data)      # Updated mean
newSigma= Sigma - Sigma*H.T * (R + H*Sigma*H.T).I * H * Sigma       # Updated covariance
inputs = [mu, Sigma, H, R, data]
outputs = [newmu, newSigma]
dtypes = {inp: 'float64' for inp in inputs}

from sympy.printing.theanocode import theano_function
f = theano_function(inputs, outputs, dtypes=dtypes)
import numpy
ninputs = [numpy.random.rand(*i.shape).astype('float64') for i in inputs]
nmu, nSigma = f(*ninputs)
```

Better cities through imaging : Dobler

Symposium





Friday: open data round table

- Open data / replicability is a hot topic in science at large
- Different LHC experiments have different approaches:
  - Release a fraction of reconstructed data, possibly with Monte Carlo
  - Release a software
  - Release of analysis ntuple
- Not clear what will happen in practice (I mean, beyond PR)
- Key question IMHO: how to collaborate on new analysis techniques with people outside ATLAS (data scientists) and even in other LHC collaborations ?
  - Open datasets proposed should be enough to try new ideas. But what about discussions/topical publications
  - Time frame for data scientist is ½ year (next ICML, next NIPS…)
  - Time frame for ATLAS/CMS publication more like two years
  - But should not be an issue for non-analysis stuff, like Data Placement or tracking