Learning from Data

Marat Freytsis

University of Oregon

Machine Learning for Jet Physics — LBNL, December 13, 2017

Why bother?

In theory there's no difference between theory and practice. In practice there is.
– Yogi Berra

the data is reality
we can only produce approximations
not always good ones

ideally

- avoid spurious features
- exploit correlations where present
- learn features we haven't thought of

Why bother?

> *In theory there's no difference between theory and practice. In practice there is.*
> *– ~~Yogi Berra~~*

the data is reality
we can only produce approximations

not always good ones

ideally

- avoid spurious features
- exploit correlations where present
- learn features we haven't thought of

Why bother?

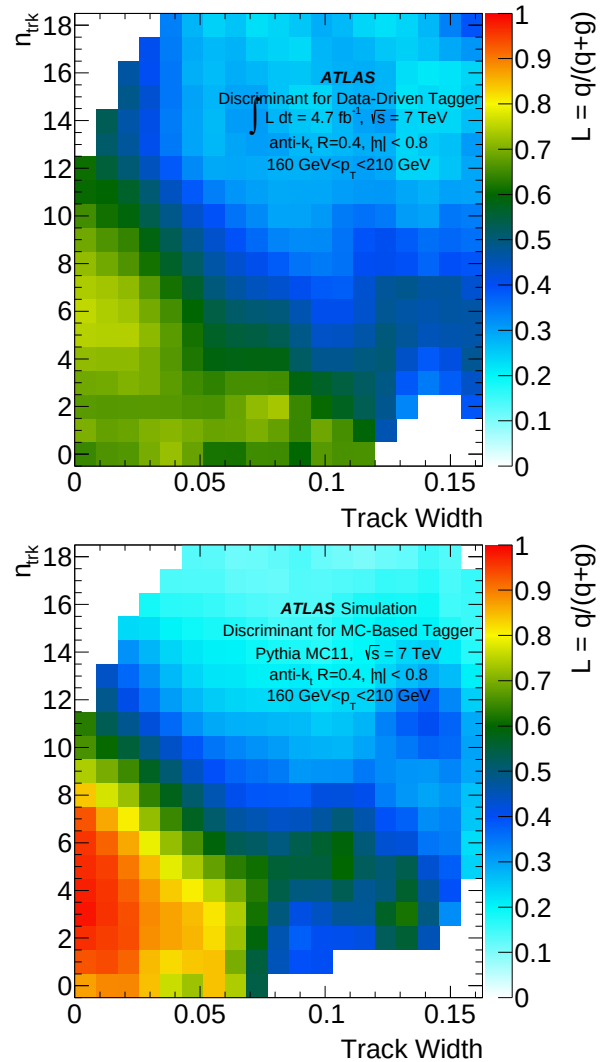
In theory there's no difference between theory and practice. In practice there is.
– ~~Yogi Berra~~

the data is reality
we can only produce approximations

not always good ones

ideally

- avoid spurious features
- exploit correlations where present
- learn features we haven't thought of



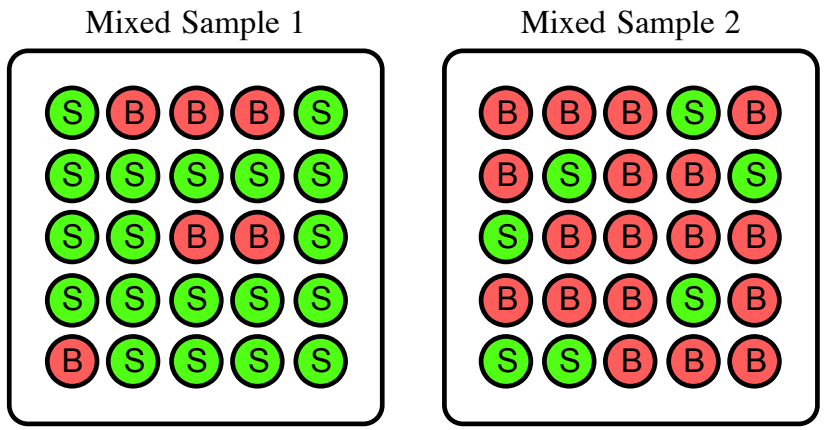
CERN-PH-EP-2014-058

Plan

- Simulation and its discontents
- Letting data drive with weak supervision
- Other features and approaches

Supervising with data

real data: can't assign truth labels, can't create pure samples
what to do? use mixed training events directly!



[arXiv:1708.02949]

only thing known is fractional composition
requires more care than fully curated training data:

- all training sets sample identical distributions
- different mixture fractions needed in each training set

fractional labels are here observables: integrated cross sections

Loss functions

how to identify signal events?

1. direct attack (learning with label proportions):

$$\ell_{\mathrm{LLP}}(\{f_t\},\{y_p\}) = |\langle f_{t,i}\rangle - \langle y_{p,i}\rangle|$$

Dery, Nachman, Rubbo, Schwartzman [arXiv:1702.00414]

   requires new loss function and training algorithm
2. clever trick (classification without labels):

$$\ell_{\mathrm{CWoLa}}(\{f_t\},\{y_p\}) = \sum_i |f_{t,i} - y_{p,i}|$$

Metodiev, Nachman, Thaler [arXiv:1708.02949]

   or your fully-supervised loss function of choice

Both of these have antecedents in the ML literature

Classification without labels

why does the second version work at all? [arXiv:1708.02949]

Theorem

Given mixed samples M_1 and M_2 defined in terms of pure samples S and B with signal fractions $f_1 > f_2$, an optimal classifier trained to distinguish M_1 from M_2 is also optimal for distinguishing S from B.

Proof.

The optimal classifier to distinguish examples drawn from p_{M_1} and p_{M_2} is the likelihood ratio $L_{M_1/M_2}(\mathbf{x}) = p_{M_1}(\mathbf{x})/p_{M_2}(\mathbf{x})$. Similarly, the optimal classifier to distinguish examples drawn from p_S and p_B is the likelihood ratio $L_{S/B}(\mathbf{x}) = p_S(\mathbf{x})/p_B(\mathbf{x})$. Where p_B has support, we can relate these two likelihood ratios algebraically:

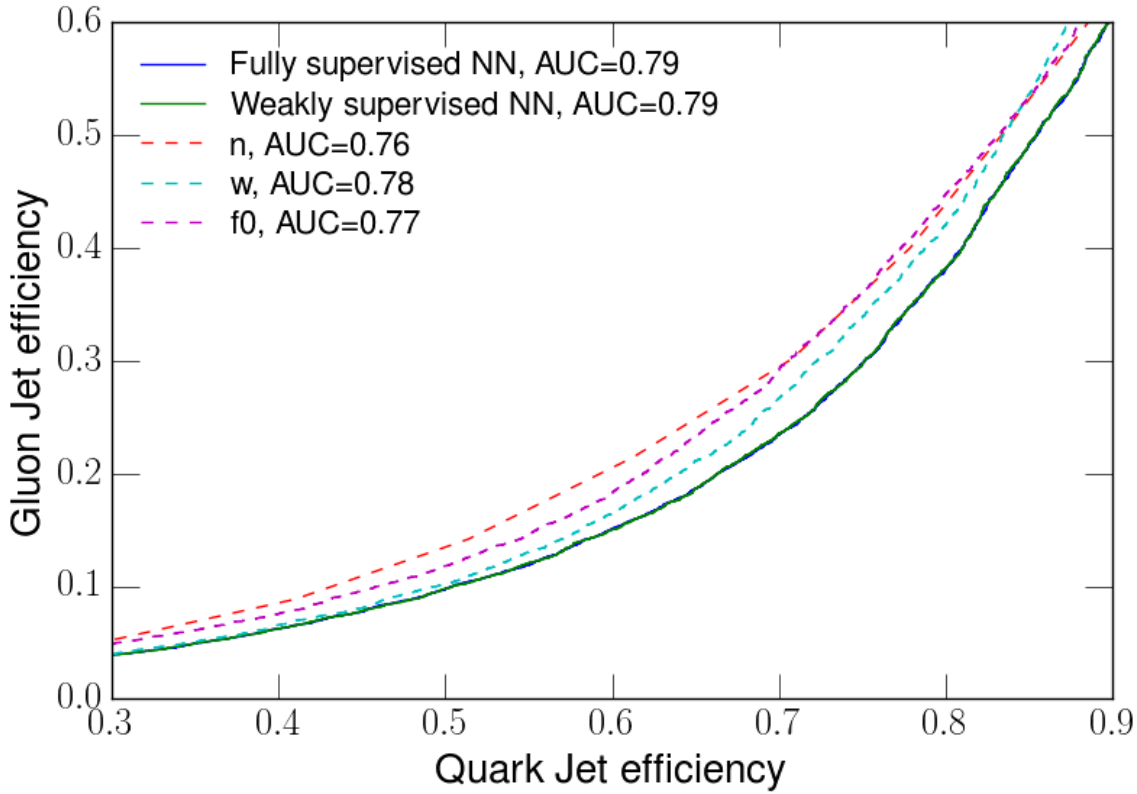
$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1 p_S + (1-f_1) p_B}{f_2 p_S + (1-f_2) p_B} = \frac{f_1 L_{S/B} + (1-f_1)}{f_2 L_{S/B} + (1-f_2)},$$

which is a monotonically increasing rescaling of the likelihood $L_{S/B}$ as long as $f_1 > f_2$, since $\partial_{L_{S/B}} L_{M_1/M_2} = (f_1 - f_2)/(f_2 L_{S/B} - f_2 + 1)^2 > 0$. If $f_1 < f_2$, then one obtains the reversed classifier. Therefore, $L_{S/B}$ and L_{M_1/M_2} define the same classifier. □

still need to know $f_{1,2}$ if you need a particular working point

Performance in simulation

LLP

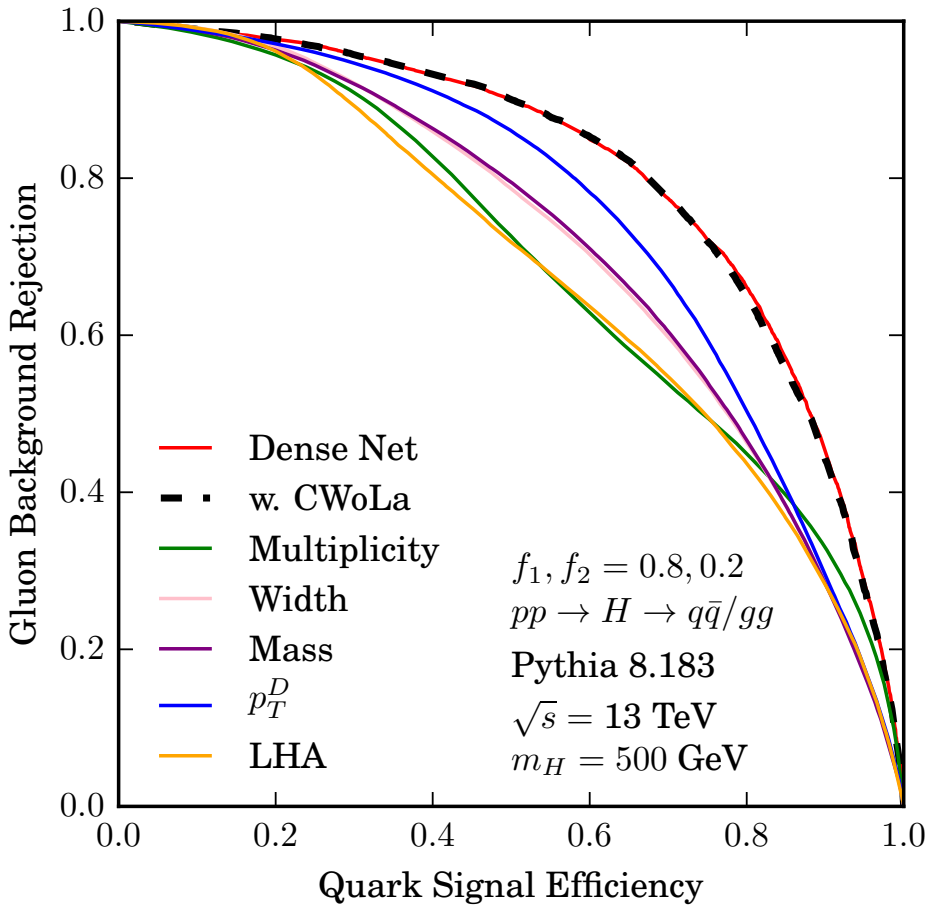


[arXiv:1702.00414]

⚠ full and weak supervision NNs have different architectures here interpret with caution!

Performance in simulation

CWoLa



[arXiv:1708.02949]

Plan

- Simulation and its discontents
- Letting data drive with weak supervision
- Other features and approaches

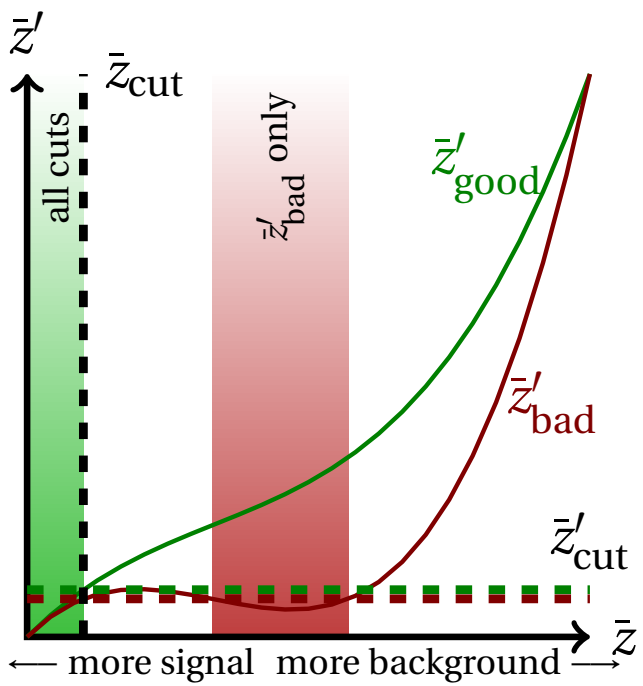
Label insensitivity

easier to understand effect of wrong fractions in LLP than full supervision

$$h_A = f_A h_1 + (1 - f_A) h_0$$
$$h_B = f_B h_1 + (1 - f_B) h_0$$
$$\Longrightarrow$$
$$h_0 = \frac{f_A h_B - f_B h_A}{f_A - f_B}$$
$$h_1 = \frac{(1-f_B) h_A - (1-f_A) h_B}{f_A - f_B}$$

optimal classifier $\bar{z} = \frac{h_1}{h_0 + h_1}$, under $f_A \to f_A + \delta$ mis-reconstructed as $\bar{z}'$

know analytic form of $\bar{z}'$



Cohen, MF, Ostdiek [arXiv:1706.09451]

A BSM example

Technical details

$$pp \to \tilde{g}\tilde{g} \text{ vs. } (Z \to \nu\bar{\nu}) + nj, \quad m_{\tilde{g}} = 2 \text{ TeV}$$

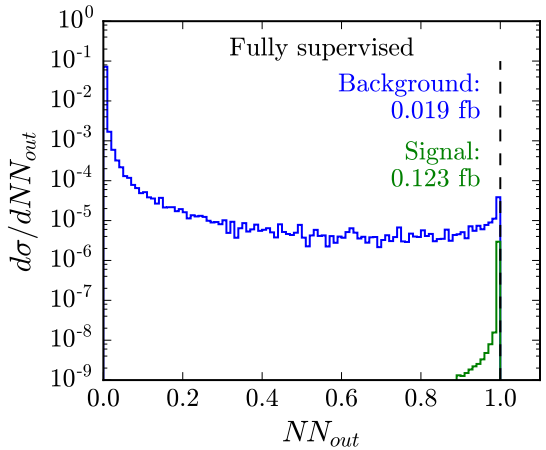
simulate in MADGRAPH5 + PYTHIA6 + DELPHES3
train on p_T of jets
KERAS with TENSORFLOW backend

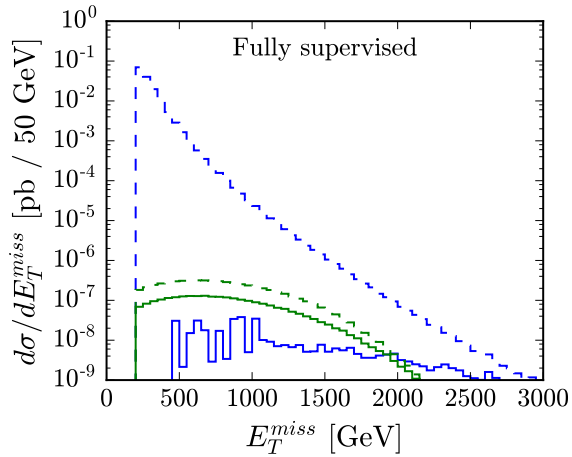
Loss function	BCE
n_{input}	11
Hidden Nodes	30
Activation	Sigmoid
Initialization	Normal
Learning algorithm	SGD
Learning rate	0.01
Batch size	64
Epochs	20

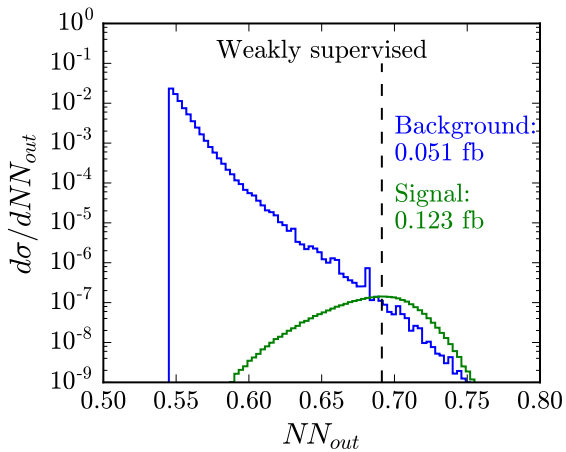
A BSM example

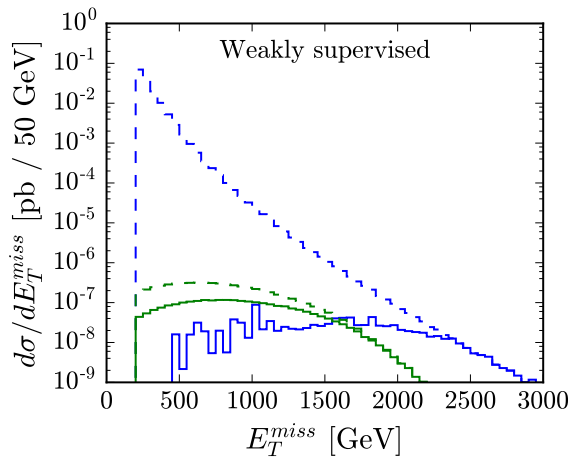
Network performance

Network	AUC	Signal efficiency
Full	0.99992393(31)	0.999373(17)
Weak	0.9998978(35)	0.999286(30)



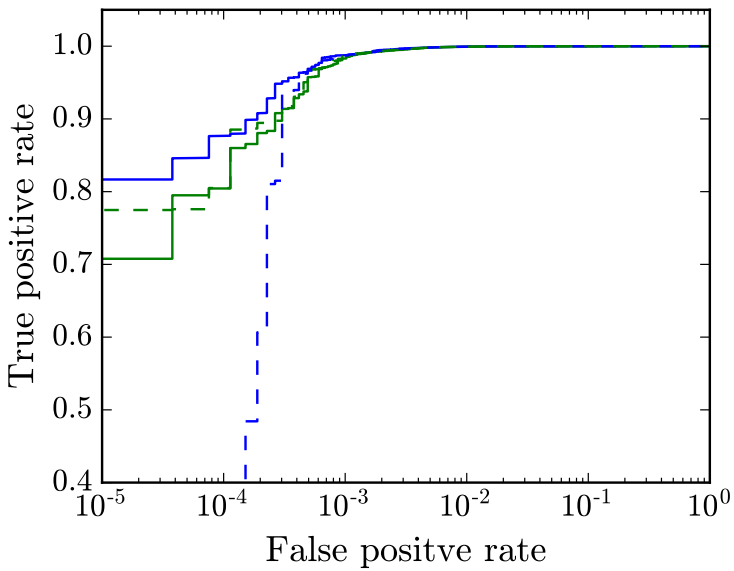




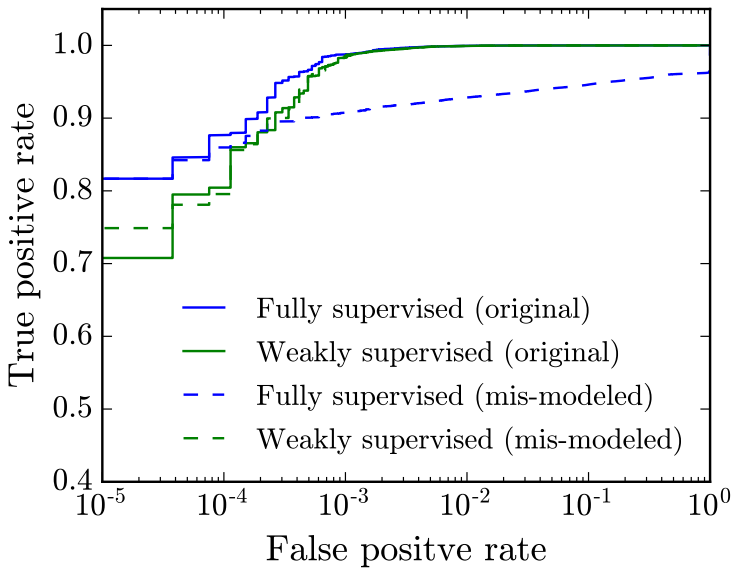


A BSM example

Impact of mismodelling



randomly swap 15% of each class



swap the 10% (15%) most signal-like (background-like)

Open questions, concrete & speculative

- performance for multi-component classification?
  - does CWoLa even have a multi-component generalization?
- how do the optimality arguments change at finite statistics?
- can we propagate uncertainties on inputs through the network?
  - would this be useful?
- can we invert any of these result to see what our models get wrong
- can we go even weaker?
  - *e.g.*, Hopfield networks and generalizations
  - can solve certain classification tasks unsupervised
  - some use in astrophysics, nearly no collider proposals to date
- ...

Now on to the newer stuff!