Thanks Eilam, Schwartz/Reisman & Weizmann Institute!
COLLABORATORS

Gilles Louppe

Kyunghyun Cho

Joan Bruna

Daniel Whiteson

Michael Kagan

Juan Pavez
Physics motivation
## Cosmology: 6 Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega_B H^2$</td>
<td>Physical Baryon Density Parameter</td>
<td>0.02230 ± 0.00014</td>
</tr>
<tr>
<td>$\Omega_C H^2$</td>
<td>Physical Dark Matter Density Parameter</td>
<td>0.1188 ± 0.0010</td>
</tr>
<tr>
<td>$T_0$</td>
<td>Age Of The Universe</td>
<td>13.799 ± 0.021 × 10⁹ Years</td>
</tr>
<tr>
<td>$N_S$</td>
<td>Scalar Spectral Index</td>
<td>0.9667 ± 0.0040</td>
</tr>
<tr>
<td>$\Delta_2$</td>
<td>Curvature Fluctuation Amplitude</td>
<td>2.441 ± 0.09 × 10⁻⁹</td>
</tr>
<tr>
<td>$T$</td>
<td>Reionization Optical Depth</td>
<td>0.066 ± 0.012</td>
</tr>
</tbody>
</table>

The Cosmic Microwave Background
A Gaussian Process in the Sky

[Image of the Cosmic Microwave Background]

- Big Bang Expansion: 13.7 billion years
- Dark Ages
- Afterglow Light Pattern 400,000 yrs.
- Quantum Fluctuations
- Development of Galaxies, Planets, etc.
- Inflation

The timeline shows:
- 1st Stars about 400 million yrs.
- Dark Energy Accelerated Expansion
- WMAP

The multipole moment $\ell$ vs. angular scale plot shows temperature fluctuations [u K²] for different angular scales.
THE "DARK MATTER" OF PHYSICS

**Ordinary Matter** *(cherry)*
- stars, dust, planets,…
- **5% of the Universe**

**Dark Matter** *(icing)*
- we know where it is, but not what it is
- **27% of the Universe**

**Dark Energy** *(cake)*
- cosmological constant, or something more exotic?
- **68% of the Universe**
THE "DARK MATTER" OF PHYSICS

THE UNIVERSE: (A PIE CHART)

5% STUFF WE KNOW (INCLUDING PIES)

68%  27%

"DARK MATTER"

NO CLUE
SIGNS OF NEW PHYSICS

1) Dark matter!

2) The matter / antimatter asymmetry in the Universe

3) Massive neutrinos

4) The “hierarchy problem”

5) The “strong CP problem”
Primordial Era
Logarithmic Scale
PARTICLE PHYSICS: 19 PARAMETERS

\[ \mathcal{L}_{SM} = \frac{1}{4} W^\mu_{\nu} \cdot W^\nu_{\mu} - \frac{1}{4} B^\mu_{\nu} B^\nu_{\mu} - \frac{1}{4} G^a_{\mu \nu} G^{a \mu \nu} \]

\[ + \left( \bar{L} \gamma^\mu (i \partial_\mu - \frac{1}{2} g Y B^\mu) L + \bar{R} \gamma^\mu (i \partial_\mu - \frac{1}{2} g' Y B^\mu) R \right) \]

\[ + \frac{1}{2} \left| (i \partial_\mu - \frac{1}{2} g Y B^\mu) \phi \right|^2 - V(\phi) \]

\[ + g''(\bar{q} \gamma^\mu T_a q) G^a_{\mu \nu} + \left( G_1 \bar{L} \phi R + G_2 \bar{L} \phi_s R + h.c. \right) \]

interactions between quarks and gluons

\[ \text{Higgs vacuum expectation value} \]

\[ v = 246 \text{ GeV} \]

\[ m_H = 125 \text{ GeV} \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$</td>
<td>Electron mass</td>
<td>511 keV</td>
</tr>
<tr>
<td>$m_\mu$</td>
<td>Muon mass</td>
<td>105.7 MeV</td>
</tr>
<tr>
<td>$m_\tau$</td>
<td>Tau mass</td>
<td>1.78 GeV</td>
</tr>
<tr>
<td>$m_u$</td>
<td>Up quark mass</td>
<td>1.9 MeV</td>
</tr>
<tr>
<td>$m_d$</td>
<td>Down quark mass</td>
<td>4.4 MeV</td>
</tr>
<tr>
<td>$m_s$</td>
<td>Strange quark mass</td>
<td>87 MeV</td>
</tr>
<tr>
<td>$m_c$</td>
<td>Charm quark mass</td>
<td>1.32 GeV</td>
</tr>
<tr>
<td>$m_b$</td>
<td>Bottom quark mass</td>
<td>4.24 GeV</td>
</tr>
<tr>
<td>$m_t$</td>
<td>Top quark mass</td>
<td>172.7 GeV</td>
</tr>
<tr>
<td>$\theta_{12}$</td>
<td>CKM 12-mixing angle</td>
<td>13.1°</td>
</tr>
<tr>
<td>$\theta_{23}$</td>
<td>CKM 23-mixing angle</td>
<td>2.4°</td>
</tr>
<tr>
<td>$\theta_{13}$</td>
<td>CKM 13-mixing angle</td>
<td>0.2°</td>
</tr>
<tr>
<td>$\delta$</td>
<td>CKM CP-violating Phase</td>
<td>0.995</td>
</tr>
<tr>
<td>$g_1$</td>
<td>U(1) gauge coupling</td>
<td>0.357</td>
</tr>
<tr>
<td>$g_2$</td>
<td>SU(2) gauge coupling</td>
<td>0.652</td>
</tr>
<tr>
<td>$g_3$</td>
<td>SU(3) gauge coupling</td>
<td>1.221</td>
</tr>
<tr>
<td>$\theta_{QCD}$</td>
<td>QCD vacuum angle</td>
<td>~0</td>
</tr>
<tr>
<td>$v$</td>
<td>Higgs vacuum expectation value</td>
<td>246 GeV</td>
</tr>
<tr>
<td>$m_H$</td>
<td>Higgs mass</td>
<td>125 GeV</td>
</tr>
</tbody>
</table>
600 million collisions every second

https://cds.cern.ch/record/1541893
600 million collisions every second
What is interesting about physics problems ...  
...from an machine learning point of view?
Reductionist

mechanistic models
clear causal structure

nuclear & particle physics
astrophysics
climate
connectome
protein folding
lattice simulations

Emergent

descriptive models
unclear causal structure

ecology
health
language
perception
psychology
systems biology
Maybe AI should start with problems where causal structure is clear and mechanistic models are available?

Reductionist
- mechanistic models
- clear causal structure

Emergent
- descriptive models
- unclear causal structure

Nuclear & particle physics
- astrophysics
- cosmology
- epidemiology
- perception
- psycholoy
- language
- ecology
- health

Protein folding
- systems biology
- lattice simulations
PHYSICS PROBLEMS HAVE A LOT OF STRUCTURE

causal structure (we take it for granted)

hierarchical / compositional structure

rich symmetries (in data & internal to generative process)

we can compare vast array of experiments in context of theoretical model ("transfer learning on steroids")

well understood correlations

non-trivial "noise models" (aka detector response)

- ambiguous factorization between "physics" and "noise model", particularly since Quantum Mechanics is probabilistic
OTHER INTERESTING ASPECTS OF PHYSICS PROBLEMS

Traditionally, our tasks cast in a probabilistic terms

• connections to statistical side of machine learning

• it’s not good enough to outperform in ideal case if it isn’t robust and/or we can’t provide reliable uncertainty quantification

Not easy to beat “human-level performance” with engineered features & heuristics

• particularly when scope of problem expands from ideal case to realistic case with scientific notion of “systematic uncertainty”
Just back from 2.5 days with physicists working with the Large Hadron Collider at CERN.

They have vast amounts of data (many Terabytes per second, generated 24/7), and very big clusters to deal with it. The growth in the amount of data is clearly outstripping the growth in computing power.
One very interesting aspect for me was that they rely very heavily on a spectrum of simulations, going from extremely detailed (and time consuming) to much cruder. They trust these simulations to the degree that they train their ML models with them, with some caveats of course.

The great importance of simulations raised the possibility of applications of generative models. They are very interested in GANs and VAEs, for the following purposes:

1. Accelerating simulations
2. Learning simulation parameters
3. Learning new physics

In my mind this is one of the most compelling applications for generative models I've seen in a while.

They have a dataset ready of 2,000,000 3D images of size 20X30X30. They have some regression/classification tasks for this data, but I think the most interesting aspect is using it to evaluate generative models the same way we now use MNIST or SVHN.
They were interested in our work on Deep Kalman Filters. They routinely use classic Kalman filters in realtime, even though there are some well-known non-linearities in their processes. I think DKF might end up being useful by:

1. Fitting the model better to the non-linearities in their data, some of which can't be modelled a-priori because they stem from unknown imperfections in the experimental apparatus.

2. "Compiling inference" [amortized inference] which has the potential of speeding up their computation. This last part will be a challenge since they have super-optimized their system, but could definitely be made to work eventually.
They are very interested in **domain adaptation** and **high-dimensional re-weighting**. For example, when they learn over the simulation data but have to test over real data. One of the speakers was trying to adapt domain-adversarial networks for one of their routine tasks.

Many of the problems which are of highest interest to them occur in "the tail" of some distribution. They might have interesting events that happen once every $10^6$ samples. That means that on a high-level, modelling the bulk of the distribution is often not as interesting as getting the tails right.
Simulation & Reconstruction
=
Generation & Inference of Latent Variables
THE PLAYERS

- **θ**: parameters of interest
- **ν**: nuisance parameters

**PREDICTION**

- forward modeling
- generation
- simulation
- (z: latent variables)

**INFERENCE**

- inverse problem
- measurement
- parameter estimation

- p(x|θ, ν)

- x: observed data, covariates, simulated data
Monte Carlo (Simulation) = (Implicit) Generative Model

Monte Carlo Truth = Latent Variables
= unobserved random variables
**THE FORWARD MODEL**

\( \mathcal{L}_{SM} = \)

\( \frac{1}{4} W_{\mu \nu} \cdot W^{\mu \nu} - \frac{1}{4} B_{\mu \nu} B^{\mu \nu} - \frac{1}{4} G_{\mu \nu}^a G^{a \mu \nu} \)

\( + \left[ \bar{L} \gamma^\mu (i \partial_\mu - \frac{1}{2} g \tau \cdot W_\mu - \frac{1}{2} g' Y B_\mu) L + \bar{R} \gamma^\mu (i \partial_\mu - \frac{1}{2} g' Y B_\mu) R \right] \)

\( + \frac{1}{2} \left[ (i \partial_\mu - \frac{1}{2} g \tau \cdot W_\mu - \frac{1}{2} g' Y B_\mu) \phi \right]^2 - V(\phi) \)

\( W^\pm, Z, \gamma, \text{and Higgs masses and couplings} \)

\( + g''(\bar{q} \gamma^{\mu} T_\alpha q) G^{a \mu} \)

\( \text{interactions between quarks and gluons} \)

\( + (G_1 \bar{L} \phi R + G_2 \bar{L} \phi q R + \text{h.c.}) \)

\( \text{fermion masses and couplings to Higgs} \)

1) We begin with Quantum Field Theory.
**THE FORWARD MODEL**

\[ \mathcal{L}_{SM} = \frac{1}{4} W^{\mu \nu} \cdot W_{\mu \nu} - \frac{1}{4} B_{\mu \nu} B^{\mu \nu} - \frac{1}{4} G_{\mu \nu} G^{\mu \nu} \]

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions

hierarchical: \( 2 \rightarrow \text{O}(10) \rightarrow \text{O}(100) \) particles
THE FORWARD MODEL

\[ \mathcal{L}_{SM} = \frac{1}{4} W_{\mu\nu} \cdot W^{\mu\nu} - \frac{1}{4} B_{\mu\nu} B^{\mu\nu} - \frac{1}{4} G^a_{\mu\nu} G^{a\mu\nu} \]

kinetic energies and self-interactions of the gauge bosons

\[ + \bar{L} \gamma^\mu (i \partial_\mu - \frac{1}{2} g \gamma^5 W^\mu - \frac{1}{2} g' Y B^\mu) L + \bar{R} \gamma^\mu (i \partial_\mu - \frac{1}{2} g' Y B^\mu) R \]

kinetic energies and electroweak interactions of fermions

\[ + \frac{1}{2} \left[ (i \partial_\mu - \frac{1}{2} g \gamma^5 W^\mu - \frac{1}{2} g' Y B^\mu) \phi \right]^2 - V(\phi) \]

W^\pm, Z, \gamma, and Higgs masses and couplings

\[ + g'^\mu (\bar{q} \gamma^\mu T^a q) G^a_{\mu} \]

interactions between quarks and gluons

\[ + (G_1 \bar{L} \phi R + G_2 \bar{L} \phi R + h.c.) \]

fermion masses and couplings to Higgs

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions

hierarchical: \( 2 \rightarrow O(10) \rightarrow O(100) \) particles
THE FORWARD MODEL

\[ \mathcal{L}_{SM} = \frac{1}{4} W_{\mu\nu} \cdot W^{\mu\nu} - \frac{1}{4} B_{\mu\nu} B^{\mu\nu} - \frac{1}{4} G_{\mu\nu}^a G^{a\mu\nu} \]

1) We begin with Quantum Field Theory

2) Theory gives detailed prediction for high-energy collisions

hierarchical: 2 \rightarrow O(10) \rightarrow O(100) particles

3) The interaction of outgoing particles with the detector is simulated.

>100 million sensors
We begin with Quantum Field Theory

The interaction of outgoing particles with the detector is simulated.

Finally, we run particle identification and feature extraction algorithms on the simulated data as if they were from real collisions.

hierarchical: $2 \rightarrow O(10) \rightarrow O(100)$ particles
“Of course, particle physicists are among the first to realize that nature is compositional.”

– YANN LECUN

“The world is compositional, or there is a god”

– JASON EISNER
SUB-ATOMIC SCALE
pencil & paper calculable from first principles
\[ p(z_1 | \theta) \]
pencil & paper calculable from first principles

\[ p(z_1 | \theta) \]

controlled approximation of first principles

\[ p(z_2 | z_1, v_1) \]
pencil & paper calculable from first principles
\[ p(z_1 | \theta) \]

controlled approximation of first principles
\[ p(z_2 | z_1, \nu_1) \]

phenomenological model
\[ p(z_3 | z_2, \nu_2) \]
Detector Simulation $p(x \mid z_3, v_3)$:
- detailed engineering (e.g., CAD)
- in situ measurements of temperature, magnetic field, alignment, calibration constants
- first-principles description of interaction of particles with matter
- measured interaction of particles with matter
FULL SIMULATION

Events \( \sim 10^{15} \)

**partons ~10**
- momenta, particle type

**hadrons ~100**
- momenta, particle type

**sensors ~10^8**
- energy deposit

**sensor readout ~10^8**
- raw data

Legend:
- **parameter of Interest**
- **nuisance parameter**
- **latent variable**
- **observed covariate**
- **derived quantities**

\[ N \quad z \quad w \quad z_1 \quad z_2 \quad \ldots \quad z_N \quad N \]

\[ M \quad w \quad w_1 \quad w_2 \quad \ldots \quad w_M \quad M \]
FULL SIMULATION

- theory parameters
- Events \( \sim 10^{15} \)
- partons \( \sim 10 \)
  - momenta, particle type
- hadrons \( \sim 100 \)
  - momenta, particle type
- sensors \( 10^8 \)
  - energy deposit
- sensors readout \( 10^8 \)
  - raw data
- raw data

Legend:
- parameter of interest
- nuisance parameter
- latent variable
- observed covariate
- derived quantities

\[ \theta \]

\[ \nu \]

\[ z_1, z_2, \ldots, z_N \]

\[ w_1, w_2, \ldots, w_M \]
Reconstruction $\approx$ point-estimate inversion of generative process (at some level of granularity)

Particle Candidates $\approx$ Results of Clustering / Segmentation

Object Identification $\approx$ Classification

Momentum Measurement $\approx$ Regression

interestingly, usually point estimates, not probabilistic at this stage
FULL SIMULATION + RECONSTRUCTION

Events $\sim 10^{15}$

partons $\sim 10$

hadrons $\sim 100$

sensors $10^8$

theory parameters

nuisance parameters

measured parton density functions, etc.

detector design, alignment

measured interactions with matter

calibration constants

momenta, particle type

momenta, summary stats

momenta, particle type

momenta, impact parameter

energy, summary stats

energy deposit

raw data

sensor readout $10^8$

parameter estimates, likelihood, posterior

event-level features

jets $\sim 10$

reconstructed particles $\sim 100$

clusters $\sim 100$

tracks $\sim 100$

Legend:

- Parameter of Interest
- Nuisance Parameter
- Latent Variable
- Observed Covariate
- Derived Quantities

$N$

$w_1, w_2, ..., w_M$

$z_1, z_2, ..., z_N$

$M$

$z_1^w, z_2^w, ..., z_N^w = N$
Physics Measurements & Searches for new particles = Likelihood-free Inference with Simulation-based Implicit Models
DETECTOR SIMULATION

**Conceptually:** $\text{Prob}(\text{detector response} \mid \text{particles})$

**Implementation:** Monte Carlo integration over micro-physics

**Consequence:** Evaluation of the likelihood is intractable
**DETECTOR SIMULATION**

**Conceptually:** \( \text{Prob(detector response | particles)} \)

**Implementation:** Monte Carlo integration over micro-physics

**Consequence:** evaluation of the likelihood is intractable

This motivates a new class of algorithms for what is called **likelihood-free inference**, which only require ability to generate samples from the simulation in the “forward mode”
Most measurements and searches for new particles at the LHC are based on the distribution of a single variable or feature

- choosing a good variable (feature engineering) is a task for a skilled physicist and tailored to the goal of measurement or new particle search

- likelihood $p(x|\theta)$ approximated using histograms (univariate density estimation)
$H \rightarrow ZZ \rightarrow 4l$
**TERMINOLOGY**

**Reconstructed Object:** hypothesized particle responsible for a subset of the raw sensor data

- **object identification (ID):** classification of reconstructed objects where class labels are usually particle types (electron, muon, photon, jet, b-jet, …)

- **reconstructed variables:** estimates of the particle’s energy, momentum, production location

**Event:** one collision (& associated sensor data and reconstructed quantities)

- **event selection:** a filter / binary classifier based on event-level data

- **discriminating variable:** a summary statistic for an event
A PHYSICALLY MOTIVATED FEATURE

Don’t believe the media:

\[ E \neq mc^2 \]
A PHYSICALLY MOTIVATED FEATURE

Don’t believe the media:

\[ E \neq mc^2 \]

What Einstein really said:

\[ E^2 = (mc^2)^2 + (|\vec{p}|c)^2 \]
A Physically Motivated Feature

Don’t believe the media:

\[ E \neq mc^2 \]

What Einstein really said:

\[ E^2 = (mc^2)^2 + (|\vec{p}|c)^2 \]

Every physics student knows energy and momentum are conserved

\[ E_{\text{Higgs}} = E_{\text{before}} = E_{\text{after}} = \sum_{i} E_i \]

\[ \vec{p}_{\text{Higgs}} = \vec{p}_{\text{before}} = \vec{p}_{\text{after}} = \sum_{i} \vec{p}_i \]
A PHYSICALLY MOTIVATED FEATURE

Don’t believe the media:

\[ E \neq mc^2 \]

What Einstein really said:

\[ E^2 = (mc^2)^2 + (|\vec{p}|c)^2 \]

Every physics student knows energy and momentum are conserved

\[ E_{\text{Higgs}} = E_{\text{before}} = E_{\text{after}} = \sum \limits_i E_i \]

\[ \vec{p}_{\text{Higgs}} = \vec{p}_{\text{before}} = \vec{p}_{\text{after}} = \sum \limits_i \vec{p}_i \]

Thus, we can estimate the mass of the Higgs with

\[ m_H = \sqrt{E_{\text{after}}^2 / c^4 - |\vec{p}_{\text{after}}|^2 / c^2} \]
Figure 9: Distribution of the four-lepton reconstructed mass in the full mass range for the sum of the 4e, 2e2μ and 4μ channels. Points with error bars represent the data, shaded histograms represent the backgrounds, and the unshaded histogram the signal expectation for a mass hypothesis of \(m_H = 126\) GeV. Signal and ZZ background are normalized to the SM expectation, Z+X background to the estimation from data. The expected distributions are presented as stacked histograms. No events are observed with \(m_{4l} > 800\) GeV.

Table 3: The number of observed candidate events compared to the mean expected background and signal rates for each final state. Uncertainties include statistical and systematic sources. The results are given integrated over the full mass measurement range \(m_{4l} > 100\) GeV and for 7 and 8 TeV data combined.

<table>
<thead>
<tr>
<th>Channel</th>
<th>4e</th>
<th>2e2μ</th>
<th>4μ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZZ</td>
<td>77±10</td>
<td>191±25</td>
<td>119±15</td>
</tr>
<tr>
<td>Z+X</td>
<td>7.4±1.5</td>
<td>11.5±2.9</td>
<td>3.6±1.5</td>
</tr>
</tbody>
</table>
Collaborative Statistical Modeling

\[ f_{\text{tot}}(D_{\text{sim}}, G|\alpha) = \prod_{c \in \text{channels}} \left[ \text{Pois}(n_c|\nu_c(\alpha)) \prod_{e=1}^{n_c} f_c(x_{ce}|\alpha) \right] \cdot \prod_{p \in S} f_p(a_p|\alpha_p) \]
Collaborative Statistical Modeling
MIXTURES OF INHOMOGENEOUS POISSON PROCESSES

**ATLAS**

$\gamma S = 8 \text{ TeV}, 20.3 \text{ fb}^{-1}$

- $p_T > 20 \text{ GeV}, |\eta| < 2.4$
- ee-channel

**ATLAS**

$\gamma S = 8 \text{ TeV}, 20.3 \text{ fb}^{-1}$

- Data
- $Z \rightarrow ee$
- $\gamma \gamma \rightarrow \mu \mu/ee$
- WW, WZ, ZZ
- $W \rightarrow l \nu$
- multi-jet
- $Z \rightarrow \tau \tau$
- $t \bar{t} + \text{Single top}$

**CMS Preliminary, 8 TeV, 20.6 \text{ fb}^{-1}**

- $\gamma Z \rightarrow \mu^+\mu^-$
- $t \bar{t}$, $tW$, WW, WZ, ZZ, $\tau \tau$
- jets (data)

$\int L \, dt = 1.4 \text{ pb}^{-1}$

**Data 2010 ($S = 7 \text{ TeV}$)**

- $b\bar{b}+c\bar{c}$ - FONLL
- $W \rightarrow \mu \nu$ - MC@NLO
- $Z \rightarrow \mu \mu$ - MC@NLO
- Drell Yan - PYTHIA
- $W \rightarrow \tau \nu$ - PYTHIA
- $Z \rightarrow \tau \tau$ - PYTHIA
- $t \bar{t}$ - PYTHIA

$| \eta | > 2.4 \text{ TeV!}$

2.4 TeV!
Most measurements and searches for new particles at the LHC are based on the distribution of a single variable or feature

- choosing a good variable (feature engineering) is a task for a skilled physicist and tailored to the goal of measurement or new particle search

- likelihood $p(x|\theta)$ approximated using histograms (univariate density estimation)

### Figure 9: Distribution of the four-lepton reconstructed mass in the full mass range for the sum of the 4e, 2e2µ and 4µ channels. Points with error bars represent the data, shaded histograms represent the backgrounds, and the unshaded histogram the signal expectation for a mass hypothesis of $m_H = 126$ GeV. Signal and ZZ background are normalized to the SM expectation, $Z^+X$ background to the estimation from data. The expected distributions are presented as stacked histograms. No events are observed with $m_4` > 800$ GeV.
Most measurements and searches for new particles at the LHC are based on the distribution of a single variable or feature

- choosing a good variable (feature engineering) is a task for a skilled physicist and tailored to the goal of measurement or new particle search

- likelihood $p(x|\theta)$ \textit{approximated} using histograms (univariate density estimation)

\[ 10^8 \text{ SENSORS } \rightarrow 1 \text{ REAL-VALUED QUANTITY} \]

This doesn’t scale if $x$ is high dimensional!
Bosons are produced, and a background process with the state momentum of the quarks is not known. Such as neutrinos (momentum transverse to the beam axis, therefore any conservation of momentum. The initial state has zero inferred in the plane transverse to the beam by requiring particles. This cannot be directly measured, but can be the amount of momentum carried away by the invisible angles.

Momentum is determined by three measurements: the momentum of the quarks (collimated streams of particles originating from quarks direction and momentum to be measured. When a particle enters a detector, it interacts with them in a way that allows its identity to be discovered, though new particles cannot directly into other particles. Though new particles cannot be directly into other particles, the internal representation of a NN is notoriously difficult to interpret. They are too complex to completely understand the insight contained in the high-level features, suggesting that they are equivalent performance using the low-level features and not capture all of the information contained in the low-level features. This is a well-known problem with shallow models.

We demonstrate that recent developments in deep learning mechanisms by which the deep network (DN) is improving the performance of the deep network with low-level features to the point of giving equivalent rejection of 90% of the background by a minimum threshold on the NN or DN output, choosing it as the best algorithm for the benchmark case here.

Benchmark Case for Higgs Bosons (HIGGS)

The first benchmark classification task is to distinguish Observable decay products include electrically-charged particles (polar) and pseudoscalar and a final state (azimuthal). For convenience, at a right angle, $\tan^{-1}(\frac{\Delta E}{\Delta \phi})$ is used instead of $\Delta E/\Delta \phi$ for distinguishing between jets from heavy quarks (or gluons, denoted $(b_j)$ and $(g_j)$) and a dijet process, which mimics which leads to jets from heavy electrically-charged Higgs bosons. The Higgs boson decays to a heavy electrically-charged Higgs bosons and two identical decay products but distinct kinematic features. There are two categories involving top-quarks $(t_j)$ and $(\bar{b}_j)$, see Figure 1.

Simulated events are generated with the delphes Collider, with showering and hadronization performed by a minimum threshold on the NN or DN output, choosing it as the best algorithm for the benchmark case here.

Figure 7 and Table I show the signal detection efficiency and background rejection for varying thresholds on the output of the neural network for 100 signal events and 1000 background events.

### Table I: Comparison of Performance

<table>
<thead>
<tr>
<th>Technique</th>
<th>Low-level</th>
<th>High-level</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN lo-level</td>
<td>0.73 ± 0.07</td>
<td>0.81 ± 0.01</td>
<td>0.80 ± 0.00</td>
</tr>
<tr>
<td>NN hi-level</td>
<td>0.78 ± 0.00</td>
<td>0.88 ± 0.00</td>
<td>0.88 ± 0.00</td>
</tr>
<tr>
<td>DN lo-level</td>
<td>0.73 ± 0.07</td>
<td>0.80 ± 0.00</td>
<td>0.80 ± 0.00</td>
</tr>
<tr>
<td>DN hi-level + lo-level</td>
<td>0.80 ± 0.00</td>
<td>0.88 ± 0.00</td>
<td>0.88 ± 0.00</td>
</tr>
</tbody>
</table>

The mean Area Under the Curve (AUC) of the signal-rejection curve in Figure 7, with standard deviations in parentheses as shown in Table I.
signal vs. background:

- $y=1$ & $y=0$ labels for binary classification
- $y=0$ background typically already studied and null hypothesis.
  - labeled training data comes either from simulation or region of the data assumed to be dominated by $y=0$ samples
- $y=1$ often corresponds to a hypothetical particle or interaction that has never been observed
  - thus labeled training data has to come from simulation
- occasionally, multi-class with a small set of “signal” categories and “background” categories
FIXED VS. VARIABLE LENGTH, SPARSE VS. DENSE

Low-level detector-level sensor data:

- fixed length $\sim 10^8$ real-valued sensor measurements
- raw sensor data is often very sparse (no energy deposited in many of the sensors)

Mid-level “reconstructed objects”:

- variable number of them, typically sorted by “$p_T$” / “transverse momentum”
- like attention mechanism / segmentation + engineered features
- relatively modest data associated to each object

High-level event quantities

- typically fixed length summary statistics / engineered features
- dimensionality of data manifold at this level is often same as dimensionality of features (eg. “dense and full rank”)
In probabilities:

- we are often looking for small deviations in the tails of distributions.
- Anomaly detection hard because small mismodelling of core probably less interesting than bigger relative change in tail
- labeled training data may be roughly balanced, but class proportions in real data may be very different (importance sampling & weighted data sets)
  - moreover, class proportions in real data may be uncertain or parameter of interest

In data:

- relevant energies range from 0.1 GeV — 10,000 GeV with features a steeply falling spectrum
  - compare with ~Uniform(0,256) pixel intensity in images
- Can log-transform individual components, but that doesn’t help with some complicated combinations of variables
A COMMON THEME

Home

This website keeps track of developments in approximate Bayesian computation (ABC) (a.k.a. likelihood-free), a class of computational statistical methods for Bayesian inference under intractable likelihoods. The site is meant to be a resource both for biologists and statisticians who want to learn more about ABC and related methods. Recent publications are under Publications 2012. A comprehensive list of publications can be found under Literature. If you are unfamiliar with ABC methods see the Introduction. Navigate using the menu to learn more.

ABC in Montreal

Approximate Bayesian computation (ABC) or likelihood-free (LF) methods have developed mostly beyond the radar of the machine learning community, but are important tools for a large and diverse segment of the scientific community. This is particularly true for systems and population biology, computational neuroscience, computer vision, healthcare sciences, but also many others.

Interaction between the ABC and machine learning community has recently started and contributed to important advances. In general, however, there is still significant room for more intense interaction and collaboration. Our workshop aims at being a place for this to happen.
Probabilistic models are an important tool in machine learning. They form the basis for models that generate realistic data, uncover hidden structure, and make predictions. Traditionally, probabilistic models in machine learning have focused on prescribed models. Prescribed models specify a joint density over observed and hidden variables that can be easily evaluated. The requirement of a tractable density simplifies their learning but limits their flexibility --- several real world phenomena are better described by simulators that do not admit a tractable density. Probabilistic models defined only via the simulations they produce are called implicit models.

Arguably starting with generative adversarial networks, research on implicit models in machine learning has exploded in recent years. This workshop’s aim is to foster a discussion around the recent developments and future directions of implicit models.

Implicit models have many applications. They are used in ecology where models simulate animal populations over time; they are used in phylogenetics, where simulations produce hypothetical ancestry trees; they are used in physics to generate particle simulations for high energy processes. Recently, implicit models have been used to improve the state-of-the-art in image and content generation. Part of the workshop’s focus is to discuss the commonalities among applications of implicit models.

Of particular interest at this workshop is to unite fields that work on implicit models. For example:

- **Generative adversarial networks** (a NIPS 2016 workshop) are implicit models with an adversarial training scheme.
- Recent advances in **variational inference** (a NIPS 2015 and 2016 workshop) have leveraged implicit models for more accurate approximations.
- **Approximate Bayesian computation** (a NIPS 2015 workshop) focuses on posterior inference for models with implicit likelihoods.
- Learning implicit models is deeply connected to **two sample testing, density ratio and density difference** estimation.

We hope to bring together these different views on implicit models, identifying their core challenges and combining their innovations.
Our simulators implicitly define a distribution $p(x,z | \theta)$

- usually relatively few parameters $\theta$, which are interpretable
- but simulating one event can involve $10^8$ random numbers $z$
  - thus, latent space $z \in \mathbb{Z}$ is horribly complicated with no fixed dimensionality
- in principle, can reparametrize $x = g(z|\theta)$ with deterministic $g$, but simulators are written in low-level programming languages like C++ with complicated control flow
  - thus, non-differentiable & intractable density

Typically we aren’t very interested in inferring latent variables $p(z|x)$, but we are interested in inference in parameters $p(\theta|x)$
| Model       | Goal is to estimate | likelihood-free | θ inference               | Generator p(x|θ) |
|------------|---------------------|----------------|--------------------------|----------------|
| ABC        | p(θ | x₀)              | yes            | approximate              | —              |
| BBVI       | p(θ,z | x)             | no             | —                        | —              |
| AEVB       | p(φ,z | x)             | yes            | approximate              | surrogate      |
| c-GAN      | p(x|θ)               | yes            | —                        | surrogate      |
| NVP/IAF    | p(x)                 | yes            | —                        | surrogate      |
| CARL       | p(x | θ)/p(x | θ₁)       | yes            | exact                    | simulation@ θ₁ x importance sampling to θ |
| “c-NVP”    | p(x|θ) via bijections x(z|θ) | yes            | exact                    | surrogate      |
**carl module**

*carl* is a toolbox for likelihood-free inference in Python.

The likelihood function is the central object that summarizes the information from an experiment needed for inference of model parameters. It is key to many areas of science that report the results of classical hypothesis tests or confidence intervals using the (generalized or profile) likelihood ratio as a test statistic. At the same time, with the advance of computing technology, it has become increasingly common that a simulator (or generative model) is used to describe complex processes that tie parameters of an underlying theory and measurement apparatus to high-dimensional observations. However, directly evaluating the likelihood function in these cases is often impossible or is computationally impractical.

In this context, the goal of this package is to provide tools for the likelihood-free setup, including likelihood (or density) ratio estimation algorithms, along with helpers to carry out inference on top of these.

*This project is still in its early stage of development. Join us on GitHub if you feel like contributing!*

[build passing coverage 91% DOI 10.5281/zenodo.47798]

**Likelihood-free inference with calibrated classifiers**


**Installation**

The following dependencies are required:

- Numpy >= 1.11
Unifying generative models and exact likelihood-free inference with conditional bijections

By Kyle Cranmer, Gilles Louppe

Recent work in density estimation uses a bijection $f : X \rightarrow Z$ (e.g. an invertible flow or autoregressive model) and a tractable density $p(z)$ (e.g. [1] [2] [3] [4]).

$$p(x) = p(f_\phi(x)) \left| \det \left( \frac{\partial f_\phi(x)}{\partial x} \right) \right|,$$

where $\phi$ are the internal network parameters for the bijection $f_\phi$. Learning proceeds via gradient ascent $\nabla \phi \sum_i \log p(x_i)$ with data $x_i$ (i.e. maximum likelihood wrt. the internal parameters $\phi$).

Since $f$ is invertible, then this model can also be used as a generative model for $X$.

This can be generalized to the conditional density $p(x|\theta)$ by utilizing a family of bijections $f_{\theta} : X \rightarrow Z$ parametrized by $\theta$ (e.g. [5] [6]).

$$p(x|\theta) = p(f_{\theta_\phi}(x)) \left| \det \left( \frac{\partial f_{\theta_\phi}(x)}{\partial x} \right) \right|$$

Here $\theta$ and $x$ are input to the network (and its inverse) and $\phi$ are internal network parameters.

Again, learning proceeds via gradient ascent $\nabla \phi \sum_i \log p(x_i|\theta_i)$ with data $x_i, \theta_i$.

We observe that not only can this model be used as a conditional generative model $p(x|\theta)$, but it can also be used to perform asymptotically exact, amortized likelihood-free inference on $\theta$.

This is particularly interesting when $\theta$ is identified with the parameters of an intractable, non-differentiable computer simulation or the conditions of some real world data collection process.

Comments

Many thanks to Durk Kingma, Max Welling, Ian Goodfellow, and Shakir Mohamed for enlightening discussions at NIPS2016.

Kyle Cranmer · 9 Dec, 2016
Generative Models

“What I cannot create, I do not understand.”

—RICHARD FEYNMAN
Auto-Encoding Variational Bayes

[Kingma and Welling, 2013/2014]
[Rezende et al, 2014]

- $q_\phi(z|x) = N(\mu, \sigma^2)$
  $[\mu, \sigma^2] = f^{z|x}(x, \phi) = \text{multilayer neural net}$

- Objective: lower bound of log $p(x)$.
  - Jointly optimized w.r.t. $\phi$ and $\theta$
  - This is approx. maximum likelihood
  - Simple SGD:
    - Sampling small minibatches of data
    - Sampling from approx. posterior

- This also minimizes an expected KL divergence
  $D_{KL}(q_\phi(z|x) \parallel p(z|x))$
  -> gives us cheap approx. inference for new datapoints

Conv. net as encoder/decoder, trained on faces

Kingma and Welling, Auto-encoding Variational Bayes, ICLR 2014
Rezende, Mohamed and Wierstra, Stochastic back-propagation and variational inference in deep latent Gaussian models, ICML 2014
FULL SIMULATION + RECONSTRUCTION

Events $\sim 10^{15}$

- partons $\sim 10$
- hadrons $\sim 100$
- sensors $10^8$
- sensor readout $10^8$
- jets $\sim 10$
- reconstructed particles $\sim 100$
- clusters $\sim 100$

- theory parameters
- parameter estimates, likelihood, posterior
- event-level features
- jets
- reconstructed particles
- clusters
- tracks

- momenta, particle type
- energy, summary stats
- momenta, impact parameter

- raw data
- energy deposit

- nuisance parameters
- measured parton density functions, etc.
- detector design, alignment
- measured interactions with matter
- calibration constants

Legend
- parameter of Interest
- nuisance parameter
- latent variable
- observed covariate
- derived quantities

Computationally expensive! $\sim 10$ min per event
LEARNING THE (SIMULATED) DATA DISTRIBUTION

Noise ~ N(0,1)

Generative Model

http://torch.ch/blog/2015/11/13/gan.html
WAVENET: A GENERATIVE MODEL FOR RAW AUDIO
WAVENET: A GENERATIVE MODEL FOR RAW AUDIO
WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

Output

Hidden Layer

Hidden Layer

Hidden Layer

Input

1 Second
CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks

Michela Paganini\textsuperscript{a,b}, Luke de Oliveira\textsuperscript{a}, and Benjamin Nachman\textsuperscript{c}
\textsuperscript{a}Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA, 94720, USA
\textsuperscript{b}Department of Physics, Yale University, New Haven, CT 06520, USA
\textsuperscript{c}E-mail: michela.paganini@yale.edu, lukedeoliveira@lbl.gov, bnachman@cern.ch

Creating Virtual Universes Using Generative Adversarial Networks

Mustafa Mustafa\textsuperscript{a,d}, Deborah Bard\textsuperscript{a}, Wahid Bhimji\textsuperscript{e}, Rami Al-Rfou\textsuperscript{f}, and Zarija Lukić\textsuperscript{g}
\textsuperscript{a}Lawrence Berkeley National Laboratory, Berkeley, CA 94720
\textsuperscript{b}Google Research, Mountain View, CA 94043

Figure 9: Five randomly selected $e^+$ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

Figure 10: Five randomly selected $\gamma$ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.

Figure 11: Five randomly selected $\pi^+$ showers per calorimeter layer from the training set (top) and the five nearest neighbors (by euclidean distance) from a set of CaloGAN candidates.
Systematic Uncertainty

Given model $p(x|\theta, \nu)$, we want inference on $\theta$ that is robust to uncertainty in nuisance parameters $\nu$

(Domain Adaptation, Fairness, ...)
Training DANN then parallels the single layer case and consists in optimizing

\[
E(\theta_f, \theta_y, \theta_d) = \frac{1}{n} \sum_{i=1}^{n} L_y^i(\theta_f, \theta_y) - \lambda \left( \frac{1}{n} \sum_{i=1}^{n} L_d^i(\theta_f, \theta_d) + \frac{1}{n} \sum_{i=n+1}^{N} L_d^i(\theta_f, \theta_d) \right),
\]

by finding the saddle point \( \hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d \) such that

\[
(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d),
\]

\[
\hat{\theta}_d = \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d).
\]
COMMENT ON DOMAIN ADAPTATION

Problem with domain adaptation approach:

• assumes class proportions / prior for class labels $p(y)$ is the same for both domains (ie. labeled simulated data and unlabeled real data)

• otherwise domain adversary would just exploit differences in $p(x|y=0)$ vs. $p(x|y=1)$ not mis-modeling / domain differences

• but in searches for new particles we don’t know proportion of $y=1$ (signal) labels in real data (it’s probably 0)

We could apply domain adversary to auxiliary training samples if we are willing to assume known class proportions & auxiliary sample is representative of the same mis-modeling / domain differences
AN ALTERNATIVE TO DOMAIN ADAPTATION

In Domain Adaptation, no attempt to identify source of differences between the two domains.

In HEP, often we have some understanding of the source of modeling (i.e. the parts of the causal simulation that would need to be adjusted in order have data and simulation agree)

- or “reweight” so that the reweighed sample agrees with the data
- we often parametrize the distribution of these simulated samples with continuous nuisance parameters

Can restate goal as inference that is robust to nuisance parameters

- In statistics terminology, want a pivotal quantity
- works with continuous nuisance parameter / family of domains
Typically classifier $f(x)$ trained to minimize loss $L_f$.

- want classifier output to be insensitive to systematics (nuisance parameter $\nu$)
- introduce an adversary $r$ that tries to predict $\nu$ based on $f$.
- setup as a minimax game:
  $$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f} \max_{\theta_r} E(\theta_f, \theta_r).$$
  $$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r)$$

LEARNING TO PIVOT WITH ADVERSARIAL NETWORKS

Typically classifier \( f(x) \) trained to minimize loss \( L_f \).

- want classifier output to be insensitive to systematics (nuisance parameter \( v \))
- introduce an adversary \( r \) that tries to predict \( v \) based on \( f \).
- setup as a minimax game:

\[
\hat{\theta}_f, \hat{\theta}_r = \arg \min_{\theta_f} \max_{\theta_r} E(\theta_f, \theta_r).
\]

\[
E_\lambda(\theta_f, \theta_r) = L_f(\theta_f) - \lambda L_r(\theta_f, \theta_r)
\]

By assumption, the lower bound is active, thus we have

\[
f(x)(0) = \frac{1}{2} \text{large value will preferably enforces}
\]

because the nuisance parameter directly shapes the decision.

There may exist distinct but equally good solutions of convergence towards that solution by Algorithm 1 in the presence of the adversary.

The continuous nuisance parameter \( Z \) is invariant with respect to the nuisance parameter. Setting \( Z \) to a specific value will preferentially enforces a classifier because of the second condition.

This model learns to produce samples AS a way to select among the class of all optimal classifiers a function \( f \).

Assuming a gaussian prior, \( \mu \) remains a question open for future works. There may exist distinct but equally good solutions.

The dynamics of adversarial training is illustrated in Fig. 3, allowing each iteration of Algorithm 1. In the first iterations, we see an update in the parameter \( \theta_f \) corresponding to the means, with tanh and ReLU activations, followed by an output layer of 15 nodes corresponding to the means, respectively with tanh and ReLU activations, followed by a linear activation.

Output nodes for the mean values come with standard deviations and mixture coefficients implement differential definitions of entropy.

As with generative adversarial networks, we propose the following toy example. Let us consider the binary case our uncertainty about the exact location of the mean.

Figure 1 illustrates a concrete case where a classifier trained only from data generated imposing with respect to independent random variables.

Therefore pivotal.

As stated in Eqn. 1, the pivotal quantity criterion is shown here, a classifier trained only from data generated according to the former theoretical characterization of the minimax problem. When running Algorithm 1 the adversary \( r \) of the nuisance parameters as observed only through the output corresponds to the estimated value of the nuisance parameter. Setting \( \mu \) to a specific value will preferably enforces a classifier because of the second condition. Training a large value will preferably enforces a classifier for predicting \( v \) based on \( f \).
Physics-Aware Machine Learning
They were interested in our work on Deep Kalman Filters. They routinely use classic Kalman filters in realtime, even though there are some well-known non-lineararities in their processes. I think DKF might end up being useful by:

1. Fitting the model better to the non-lineararities in their data, some of which can't be modelled a-priori because they stem from unknown imperfections in the experimental apparatus.

2. "Compiling inference" [amortized inference] which has the potential of speeding up their computation. This last part will be a challenge since they have super-optimized their system, but could definitely be made to work eventually.
"There is great promise at the interface of physical models and data driven models."

–NEIL LAWRENCE

SEARCHING OVER SPACE OF MODELS

Using a class of models known as Gaussian Processes to model data

- physics goes into the construction of a "Kernel" that describes covariance of data

Vocabulary of kernels + grammar for composition

Structure Discovery in Nonparametric Regression through Compositional Kernel Search

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani
International Conference on Machine Learning, 2013
pdf | code | poster | bibtex

Exploiting compositionality to explore a large space of model structures

Roger Grosse, Ruslan Salakhutdinov, William T. Freeman, Joshua B. Tenenbaum
Conference on Uncertainty in Artificial Intelligence, 2012
pdf | code | bibtex

Mauna Loa atmospheric CO₂

Residuals
Instead of fitting the dijet spectrum with an ad hoc 3-5 parameter function, use GP with kernel motivated from physics

Final Kernel =

Poisson stats
+ Mass Resolution
+ Parton Density Functions
+ Jet Energy Scale

\[
\mu(x) = p_0 \times (1 - \frac{x}{\sqrt{S}})^{p_1} \times (\frac{x}{\sqrt{S}})^{p_2}
\]
Many scenarios for physics Beyond the Standard Model include highly boosted W, Z, H bosons or top quarks.

Identifying these rests on subtle substructure inside jets.

- an enormous number of theoretical effort in developing observables and techniques to tag jets like this.
Last year deep learning algorithms applied to “jet images”

- based on fast simulation & idealized uniform calorimeter
- preprocessed to recenter ($\eta, \phi$) & rotated

Average Boosted W Jet

Average QCD Jet
NON-UNIFORM GEOMETRY
Most “pixels” are empty. The data is sparse

- Would like to incorporate domain knowledge
- Engineered features work well & have nice theoretical properties that physicists care about
Recursive Neural Networks showing great performance for Natural Language Processing tasks

- neural network’s topology given by parsing of sentence!
FROM IMAGES TO SENTENCES

Recursive Neural Networks showing great performance for Natural Language Processing tasks

- neural network’s topology given by parsing of sentence!

**Analogy:**
word → particle
parsing → jet algorithm
QCD-INSPIRED RECURSIVE NEURAL NETWORKS

Work with Gilles Louppe, Kyunghyun Cho, Cyril Becot (arXiv:1702.00748)

- Use sequential recombination jet algorithms to provide network topology (on a per-jet basis)
- path towards ML models with good physics properties
- Top node of recursive network provides a fixed-length embedding of a jet that can be fed to a classifier
EVENT EMBEDDINGS

Jointly optimize jet embedding → event embedding → classifier

It scales!
EXPLOITING SYMMETRY

Physics is ripe with symmetries, we should incorporate that knowledge into our models

- difficulty: often detector breaks symmetries

Symmetry in Deep Learning

- What makes CNNs so effective?
  - Weight sharing: exploits translation symmetry
  - Depth: exploits equivariance

Network design principle: 
Equivariance to symmetry transformations

Conv vs G-Conv

Planar Convolution

“translate filter and compute inner product”

Translation

\[ T_s f(x) = f(x - s) \]

\[ T_{(2,1)} \begin{bmatrix} F \\ \end{bmatrix} = \begin{bmatrix} F' \end{bmatrix} \]

\[ Z^2\text{-Convolution} \]

\[ [f \ast \psi](s) = \sum_{x \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(x) [T_k \psi]_{k}(x) \]

Group Convolution

“transform filter and compute inner product”

Transformation

\[ T_r f(x) = f(r^{-1}x) \]

\[ T_{r} \begin{bmatrix} F \end{bmatrix} = \begin{bmatrix} F' \end{bmatrix} \]

\[ G\text{-Convolution} \]

\[ [f \ast \psi](g) = \sum_{x \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(x) [T_g \psi]_{k}(x) \]
Graph Convolutional Networks

- A small number of function primitives, i.e., convolution kernel, pooling, ...
- A hierarchical composition of such primitive functions guided by geometry and prior knowledge
- Generalizable to non-planar input

[Bruna et al., 2013; Henaff et al., 2015] and many follow-up work
EXPLOITING COMPOSITIONAL STRUCTURE

Spectrum of function approximation

<table>
<thead>
<tr>
<th>Classes</th>
<th>Compositional Structure</th>
<th>Function Primitives</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Known</td>
<td>Known</td>
<td>Known</td>
<td>No learning necessary</td>
</tr>
<tr>
<td>Partially Known</td>
<td>Known</td>
<td>Some known</td>
<td></td>
</tr>
<tr>
<td>Structure only</td>
<td>Known</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>Primitives only</td>
<td>Unknown</td>
<td>Known</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>1. Compositional structure may be expected, vs. 2. Nothing assumed</td>
</tr>
</tbody>
</table>

NN Approximation of a primitive function

- Wrap a (un)known function
- The wrapper contains both the known function and a NN that approximates it
- Intermediate loss is defined as the discrepancy between the known function and the NN
- Output may be either the known function’s and NN’s output

Slides from Kyunghyun Cho at DataScience @ HEP 2017
How do these fit together?

Combine many of these ideas:

- **Large model, but sparsely activated**
- **Single model to solve many tasks** (100s to 1Ms)
- **Dynamically learn** and grow pathways through large model
- **Hardware specialized for ML supercomputing**
- **ML for efficient mapping** onto this hardware
WHAT IS THE OBJECTIVE?

ML: What is the problem you are trying to solve?

Physicist: [eventually describes problem and formalizes objective]

ML: Ok, well let’s optimize this directly …

Physicist: but, I also want…..

Used to criticize physicists for constantly changing problem statement, but traditional approach to physics problems has many advantages

- modular, reusable components (facilitates transfer learning, “ML2.0”)
- interpretable & individually validated
- a form of structural regularization

Reminds me of concept of Meta-Heuristics
A COMMON VISION

Events \( \sim 10^{15} \)

- Partons \( \sim 10 \)
  - Momentum, particle type

- Hadrons \( \sim 100 \)
  - Momentum, particle type

- Sensors \( \sim 10^{18} \)
  - Energy deposit

- Nuisance parameters

- Nuisance parameter of Interest

- Derived quantities

- Latent variable

- Observed covariate

- Detector design, alignment

- Measured interactions with matter

- Calibration constants

Outputs

- Single large model, sparsely activated

Tasks

- ...
Backup
Metaheuristic

From Wikipedia, the free encyclopedia

In computer science and mathematical optimization, a metaheuristic is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity. Metaheuristics sample a set of solutions which is too large to be completely sampled. Metaheuristics may make few assumptions about the optimization problem being solved, and so they may be usable for a variety of problems.

Compared to optimization algorithms and iterative methods, metaheuristics do not guarantee that a globally optimal solution can be found on some class of problems. Many metaheuristics implement some form of stochastic optimization, so that the solution found is dependent on the set of random variables generated. In combinatorial optimization, by searching over a large set of feasible solutions, metaheuristics can often find good solutions with less computational effort than optimization algorithms, iterative methods, or simple heuristics. As such, they are useful approaches for optimization problems. Several books and survey papers have been published on the subject.

Most literature on metaheuristics is experimental in nature, describing empirical results based on computer experiments with the algorithms. But some formal theoretical results are also available, often on convergence and the possibility of finding the global optimum. Many metaheuristic methods have been published with claims of novelty and practical efficacy. While the field also features high-quality research, unfortunately many of the publications have been of poor quality; flaws include vagueness, lack of conceptual elaboration, poor experiments, and ignorance of previous literature.
WARNING: Possible communication pitfalls
“more statistics” / “high-statistics” / “Monte Carlo Statistics"

- horrible jargon
- refers to number number of observations
- “Monte Carlo statistics” refers to number of samples / examples in data sets generated by our simulations (which use Monte Carlo techniques)
Basic probability concepts:

- probability distributions (pdfs), prior, likelihood

- we often think in terms of histograms (rarely use term “density”)
  - use mixture models all the time, often called “stacked histograms”
  - not familiar with term Poisson point process (”Extended Maximum Likelihood fit”)

- Bayes' theorem & maximum likelihood estimation

- Neyman-Pearson hypothesis testing, confidence intervals, Wilks’s theorem (asymptotics)

- our distributions are usually predicted by theory, so we don’t deal with common named distributions other than Poisson, Gaussian, log-normal, chi-square.

Not very familiar:

- classical statistics concepts like conjugate models, exponential family, ..

- somewhat familiar with bootstrapping (use parametric bootstrap heavily, but not under that name)

- Not very familiar with hierarchical Bayes, empirical Bayes, EM algorithm, …
FAMILIAR CONCEPTS FOR PHYSICISTS

Basic Machine Learning concepts:

• Use of ML was dominated by classification (eg. object id or event selection)
  • “Boosted Decision Trees” have dominated HEP usage for a decade
  • <2003 shallow neural networks were used for classification
  • not familiar with discriminative vs. generative classifiers … rarely hear term “Bayes optimal”

• More recently ML used for regression (eg. to estimate particle’s momentum)
  • not familiar with logistic, ridge, lasso, etc. regression techniques / properties

• huge surge in interest & familiarity with deep learning
  • recurrent neural networks, LSTMs for variable length input is new
  • idea of “embedding” or representation is fairly new not widespread

• Use of ML for inference, anomaly detection, generative models is recent
  • RBMs, wake/sleep, VAE, variational inference generally not familiar
UNFAMILIAR CONCEPTS FOR MANY PHYSICISTS

Graphical Models

Latent variables (known under a different name)

Gaussian Processes and Kernel Machines

Variational inference

Regularization

Bayesian Optimization

Automatic differentiation
Ironically, there are several statistical physics concepts that are key to machine learning that are not widely appreciated by particle physicists.

- eg. Gibbs sampling of energy based models / unnormalized models
- Grand canonical ensemble vs. micro-canonical ensemble
  - eg. intractable partition function / normalizing constant
- Hamiltonian Monte Carlo & Langevin dynamics (perhaps more classical physics than statistical physics)

We (HEPhysicists) need to recalls some of our statistical physics :-)

- We need a different workshop with more theoretical physicists :-)
- Looking forward to Naftali Tishby’s talk “The statistical physics of Deep Learning” & Yoav Levine & Amnon Shashua on “Deep Learning and Quantum Entanglement”
Final Thoughts

We have excellent & interested people in a wonderful setting

We have time to discuss!

Please ask questions. Don’t be afraid to ask “dumb questions”. Interrupt if you don’t understand jargon.
**TERMINOLOGY**

\( p(x|\theta) \) conditional density / pdf / likelihood / statistical model

In machine learning, \( \nabla_x \log p(x|\theta) \) often called “score” as in “score matching”, but this is non-standard terminology in statistics

In statistics, score usually means \( \nabla_\theta \log p(x|\theta) \)

Physicists almost never use the term “score”
QCD-INSPIRED RECURSIVE NEURAL NETWORKS

- down-sampling by projecting into images loses information
- RNN needs much less data to train!
ADVERSARIAL KITTEN DESKTOP COMPUTER
### COMPARISON

| Method       | Goal is to estimate | likelihood-free | $\theta$ inference | Generator $p(x|\theta)$ |
|--------------|---------------------|----------------|-------------------|------------------------|
| ABC          | $p(\theta | x_0)$      | yes            | approximate       | —                      |
| BBVI         | $p(\theta, z | x)$ | no             | —                 | —                      |
| AEVB         | $p(\phi, z | x)$    | yes            | approximate       | surrogate              |
|              |                     |                | on $\phi$ not $\theta$ |                       |
| c-GAN        | $p(x|\theta)$       | yes            | —                 | surrogate              |
| NVP/IAF      | $p(x)$              | yes            | —                 | surrogate              |
| CARL         | $p(x | \theta)/p(x | \theta_1)$ | yes            | exact             | simulation@ $\theta_1$ x importance sampling to $\theta$ |
| "c-NVP"     | $p(x|\theta)$ via bijections $x(z|\theta)$ | yes            | exact             | surrogate              |

**Exact = asymptotically consistent in infinite capacity limit**
The intractable likelihood ratio based on high-dimensional features $x$ is:

$$\frac{p(x|\theta_0)}{p(x|\theta_1)}$$

We can show that an equivalent test can be made from 1-D projection

$$\frac{p(x|\theta_0)}{p(x|\theta_1)} = \frac{p(s(x; \theta_0, \theta_1)|\theta_0)}{p(s(x; \theta_0, \theta_1)|\theta_1)}$$

if the scalar map $s: X \to \mathbb{R}$ has the same level sets as the likelihood ratio

$$s(x; \theta_0; \theta_1) = \text{monotonic}[\frac{p(x|\theta_0)}{p(x|\theta_1)}]$$

Estimating the density of $s(x; \theta_0, \theta_1)$ via the simulator calibrates the ratio.
Binary classifier on balanced $y=0$ and $y=1$ labels learns

$$s(x) = \frac{p(x|y = 1)}{p(x|y = 0) + p(x|y = 1)}$$

Which is one-to-one with the likelihood ratio

$$\frac{p(x|y = 0)}{p(x|y = 1)} = 1 - \frac{1}{s(x)}$$

Can do the same thing for any two points $\theta_0$ & $\theta_1$ in parameter space. I call this a **parametrized classifier**

$$s(x; \theta_0, \theta_1) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)}$$
In practice \( \hat{r}(\hat{s}(x; \theta_0, \theta_1)) \) will not be exact. Diagnostic procedures are needed to assess the quality of this approximation.

1. For inference, the value of the MLE \( \hat{\theta} \) should be independent of the value of \( \theta_1 \) used in the denominator of the ratio.

2. Train a classifier to distinguish between unweighted samples from \( p(x|\theta_0) \) and samples from \( p(x|\theta_1) \) weighted by \( \hat{r}(\hat{s}(x; \theta_0, \theta_1)) \).

\[
\frac{p_1(s^*)}{p_0(s^*)} = \frac{p_1(x)}{p_0(x)} \int d\Omega_{s^*} \frac{p_0(x)}{|\hat{n} \cdot \nabla s|} = \frac{p_1(x)}{p_0(x)}
\]