HEP data: Finding structure in the noise

Tim Salimans

Algoritmica

December 13, 2014
Typical types of data

Statistics / Econometrics

Example: forecasting GDP

- weak relationships
- simple structure
- high noise level

Machine Learning / AI

Example: image recognition

- strong relationships
- deep/complex structure
- low noise level
Typical types of data

Statistics / Econometrics

Example: forecasting GDP
- weak relationships
- simple structure
- high noise level

Machine Learning / AI

Example: image recognition
- strong relationships
- deep/complex structure
- low noise level

HEP data has both high noise & deep structure

https://github.com/TimSalimans/HiggsML/ tim@algoritmica.nl
Sources of noise

- imperfect detectors
  - measurement errors
  - undetected particles (e.g. neutrinos)
Sources of noise

- imperfect detectors
  - measurement errors
  - undetected particles (e.g. neutrinos)

- imperfect simulators
Sources of noise

- imperfect detectors
  - measurement errors
  - undetected particles (e.g. neutrinos)

- imperfect simulators

- $H \rightarrow \tau\tau$ signal quite rare
  - high variance in importance weights
  - small effective sample size

https://github.com/TimSalimans/HiggsML/  
tim@algoritmica.nl
Complex structure

Complex mapping from primitives to signal class

- Particle momenta individually have low correlation with signal class
- Relationship between particle momenta is complex
- Derived variables help some
- Variables like MMC are hard to calculate

https://github.com/TimSalimans/HiggsML/
Complex structure

Complex mapping from primitives to signal class

- Particle momenta individually have low correlation with signal class
- Relationship between particle momenta is complex
- Derived variables help some
- Variables like MMC are hard to calculate

Good model search needed to find correct relationship

- Hard to find correct model by greedy search
- Standard boosted decision trees (e.g. GBM in R) may perform poorly
- XGBoost / RGF are better at model search
- Neural nets excellent at discovering deep relationships
Directly maximizing AMS not a good idea: much too noisy. Better to estimate the odds ratio of signal $s$ to background $b$ and then apply a cutoff.
Statistical efficiency

Directly maximizing AMS not a good idea: much too noisy. Better to estimate the odds ratio of signal $s$ to background $b$ and then apply a cutoff.

**Goal:** build model for $R = \frac{\mathbb{E}[w \| (\text{label}=s) \| x]}{\mathbb{E}[w \| (\text{label}=b) \| x]}$, with $w$ the importance weights, $s, b$ the signal and background identifiers, and $x$ the measured particle momenta. Expectation is taken w.r.t. the simulator distribution.
Directly maximizing AMS not a good idea: much too noisy. Better to estimate the odds ratio of signal $s$ to background $b$ and then apply a cutoff.

**Goal:** build model for $R = \frac{\mathbb{E}[w \mathbb{I}(\text{label}=s) | x]}{\mathbb{E}[w \mathbb{I}(\text{label}=b) | x]}$, with $w$ the importance weights, $s, b$ the signal and background identifiers, and $x$ the measured particle momenta. Expectation is taken w.r.t. the simulator distribution.

**Problem:** The importance weights $w$ are highly variable: small effective sample size.
Statistical efficiency

Decompose the problem to improve efficiency:

\[ R = \frac{\mathbb{E}[w \mathbb{1}(\text{label} = s) | x]}{\mathbb{E}[w \mathbb{1}(\text{label} = b) | x]} = R_1 R_2, \]

with

\[ R_1 = \frac{P(\text{label} = s | x)}{P(\text{label} = b | x)} \]
\[ R_2 = \frac{\mathbb{E}[w | x, \text{label} = s]}{\mathbb{E}[w | x, \text{label} = b]} \]
Statistical efficiency

Decompose the problem to improve efficiency:

\[
R = \frac{\mathbb{E}[w \mathbb{I}(\text{label} = s) | x]}{\mathbb{E}[w \mathbb{I}(\text{label} = b) | x]} = R_1 R_2,
\]

with

\[
R_1 = \frac{P(\text{label} = s | x)}{P(\text{label} = b | x)},
\]

\[
R_2 = \frac{\mathbb{E}[w | x, \text{label} = s]}{\mathbb{E}[w | x, \text{label} = b]}
\]

Advantage: The subproblems are easier (larger effective sample size)
Statistical efficiency

Decompose the problem to improve efficiency:

\[ R = \frac{\mathbb{E}[w \mathbb{I}(\text{label} = s)|x]}{\mathbb{E}[w \mathbb{I}(\text{label} = b)|x]} = R_1 R_2, \]

with

\[ R_1 = \frac{P(\text{label} = s|x)}{P(\text{label} = b|x)} \]
\[ R_2 = \frac{\mathbb{E}[w|x, \text{label} = s]}{\mathbb{E}[w|x, \text{label} = b]} \]

**Advantage:** The subproblems are easier (larger effective sample size)

**Disadvantage:** Solving the subproblems might give a biased solution for the original problem
Gradient boosting machine

Friedman, 2001

Algorithm 1: Gradient Boosted Decision Tree (GBDT) [15]

\[
\begin{align*}
    h_0(x) & \leftarrow \arg \min_\rho \mathcal{L}(\rho, Y) \\
    \text{for } k = 1 \text{ to } K \text{ do} & \\
    \quad \tilde{Y}_k & \leftarrow -\frac{\partial \mathcal{L}(h, Y)}{\partial h}|_{h=h_{k-1}(x)} \\
    \quad \text{Build a } J\text{-leaf decision tree } T_k & \leftarrow A(X, \tilde{Y}_k) \text{ with leaf-nodes } \{b_{k,j}\}_{j=1}^J \\
    \quad \text{for } j = 1 \text{ to } J \text{ do} & \beta_{k,j} \leftarrow \arg \min_{\beta \in \mathbb{R}} \mathcal{L}(h_{k-1}(x) + \beta \cdot b_{k,j}(X), Y) \\
    \quad h_k(x) & \leftarrow h_{k-1}(x) + s \sum_{j=1}^J \beta_{k,j} \cdot b_{k,j}(x) \quad \text{// } s \text{ is a shrinkage parameter} \\
\end{align*}
\]

\[h(x) = h_K(x)\]

- functional gradient descent
- very general, no need to normalize covariates
- popular implementation in R works well for many applications
- greedy model search, does not work well for HEP data

https://github.com/TimSalimans/HiggsML/

15
Regularized greedy forest

Johnson & Zhang, 2014. Variation on gradient boosting that decouples structure search and optimization.

**Algorithm 3: Regularized greedy forest framework**

1. $\mathcal{F} \leftarrow \{\}$.  
   repeat
   2. $\mathcal{F} \leftarrow$ the optimum forest that minimizes $Q(\mathcal{F})$ among all the forests that can be obtained by applying one step of structure-changing operation to the current forest $\mathcal{F}$.  
   3. if some criterion is met then optimize the leaf weights in $\mathcal{F}$ to minimize loss $Q(\mathcal{F})$.  
      until some exit criterion is met  
      Optimize the leaf weights in $\mathcal{F}$ to minimize loss $Q(\mathcal{F})$.  
   return $h_{\mathcal{F}}(x)$

- $L_2$ regularization of leaf coefficients for noise control
- $Q()$ used in structure search can be different from $Q()$ used for optimization of leaf coefficients
- use less regularization in structure search to make the search less greedy, key to make this work for HEP
Model combination by stacking

- How to determine tuning parameters? (nr of leaves, regularization)
- Solve original problem, or decompose into subproblems?
- How to combine models for the subproblems?
Model combination by stacking

- How to determine tuning parameters? (nr of leaves, regularization)
- Solve original problem, or decompose into subproblems?
- How to combine models for the subproblems?

→ Estimate all models and combine through stacking: linear model combination with non-negative weights
Challenges of HEP data

My solution

Going forward

Conclusion

Result

Completed • $13,000 • 1,785 teams

Higgs Boson Machine Learning Challenge

Mon 12 May 2014 – Mon 15 Sep 2014 (58 days ago)

Private Leaderboard - Higgs Boson Machine Learning Challenge

This competition has completed. This leaderboard reflects the final standings.

<table>
<thead>
<tr>
<th>#</th>
<th>Δ1w</th>
<th>Team Name</th>
<th>✅ model uploaded * in the money</th>
<th>Score</th>
<th>Entries</th>
<th>Last Submission UTC (Best – Last Submission)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>Gábor Melis ‡ *</td>
<td></td>
<td>3.80581</td>
<td>110</td>
<td>Sun, 14 Sep 2014 09:10:04 (-0h)</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>Tim Salimans ‡ *</td>
<td></td>
<td>3.78913</td>
<td>57</td>
<td>Mon, 15 Sep 2014 23:49:02 (-40.6d)</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>nhlx5haze ‡ *</td>
<td></td>
<td>3.78682</td>
<td>254</td>
<td>Mon, 15 Sep 2014 16:50:01 (-76.3d)</td>
</tr>
<tr>
<td>4</td>
<td>55</td>
<td>ChoKo Team ¼</td>
<td></td>
<td>3.77526</td>
<td>216</td>
<td>Mon, 15 Sep 2014 15:21:36 (-42.1h)</td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td>cheng chen</td>
<td></td>
<td>3.77384</td>
<td>21</td>
<td>Mon, 15 Sep 2014 23:29:29 (-0h)</td>
</tr>
</tbody>
</table>

https://github.com/TimSalimans/HiggsML/ tim@algoritmica.nl
Incorporating physics knowledge

- Physics knowledge important because of difficulty of model search, and limited sample sizes
Incorporating physics knowledge

- Physics knowledge important because of difficulty of model search, and limited sample sizes
- Variables like MMC and SVFIT suboptimal: We need

\[ \int P(\text{label} = s | m) p(m | x) dm \]

rather than \( \text{arg max}_m p(m | x) \).
Incorporating physics knowledge

- Physics knowledge important because of difficulty of model search, and limited sample sizes
- Variables like MMC and SVFIT suboptimal: We need
  \[ \int P(\text{label} = s|m)p(m|x)dm \]
  rather than \( \text{arg max}_m p(m|x) \).
- Physics knowledge also important to ensure generalization to real data.

https://github.com/TimSalimans/HiggsML/  
tim@algoritmica.nl
“CAKE” variable

- Team CAKE (Thomas Gillam, Christopher Lester, Damien George) came up with a variable modelling

\[
C = \frac{p(x|H \rightarrow \tau \tau)}{p(x|Z \rightarrow \tau \tau)}
\]

which is almost the desired likelihood ratio

\[
p(x|\text{label } = s)/p(x|\text{label } = b)
\]
“CAKE” variable

- Team CAKE (Thomas Gillam, Christopher Lester, Damien George) came up with a variable modelling

\[ C = \frac{p(x|H \rightarrow \tau\tau)}{p(x|Z \rightarrow \tau\tau)} \]

which is almost the desired likelihood ratio \( p(x|\text{label} = s)/p(x|\text{label} = b) \)

- Variable improved result for many teams, but also led to model instability
  - it made my public score worse
  - would have given me winning result on private leaderboard
“CAKE” variable

- Team CAKE (Thomas Gillam, Christopher Lester, Damien George) came up with a variable modelling

\[ C = \frac{p(x|H \rightarrow \tau\tau)}{p(x|Z \rightarrow \tau\tau)} \]

which is almost the desired likelihood ratio

\[ p(x|\text{label} = s)/p(x|\text{label} = b) \]

- Variable improved result for many teams, but also led to model instability
  - it made my public score worse
  - would have given me winning result on private leaderboard

- Reason for instability: variable \( C \) is a model in itself, should restrict \( \partial p(\text{label} = s|x, C)/\partial C \) to be positive as in stacking

https://github.com/TimSalimans/HiggsML/  
tim@algoritmica.nl
“CAKE” variable

- Team CAKE (Thomas Gillam, Christopher Lester, Damien George) came up with a variable modelling

\[ C = \frac{p(x|H \rightarrow \tau\tau)}{p(x|Z \rightarrow \tau\tau)} \]

which is almost the desired likelihood ratio
\[ p(x|\text{label} = s)/p(x|\text{label} = b) \]

- Variable improved result for many teams, but also led to model instability
  - it made my public score worse
  - would have given me winning result on private leaderboard
- Reason for instability: variable \( C \) is a model in itself, should restrict \( \partial p(\text{label} = s|x, C)/\partial C \) to be positive as in stacking
- Alternative: make CAKE model more flexible and optimize parameters on the data
Real data

- Simulated data ≠ reality
- Is our model picking up on errors in the simulation?
- Will results generalize to real data?
Real data

- Simulated data $\neq$ reality
- Is our model picking up on errors in the simulation?
- Will results generalize to real data?
- Need physics knowledge to restrict model structure (a la CAKE)
Real data

- Simulated data $\neq$ reality
- Is our model picking up on errors in the simulation?
- Will results generalize to real data?

- Need physics knowledge to restrict model structure (a la CAKE)
- Can use generative modelling techniques on real (unlabeled) data
Generative modelling

*Bump hunting = unsupervised learning / generative modelling*

- General idea: high density areas define unique physical events
- Prior knowledge: classification boundaries can only occur in low density areas
- We can also do this in higher dimensions
- Combine real unlabeled data with labeled (simulated) data: semi-supervised learning

https://github.com/TimSalimans/HiggsML/ tim@algoritmica.nl
Generative modelling

*Bump hunting = unsupervised learning / generative modelling*

- **general idea:** high density areas define unique physical events
  - prior knowledge: classification boundaries can only occur in low density areas
Generative modelling

*Bump hunting = unsupervised learning / generative modelling*

- general idea: high density areas define unique physical events
  - prior knowledge: classification boundaries can only occur in low density areas
- we can also do this in higher dimensions

https://github.com/TimSalimans/HiggsML/  tim@algoritmica.nl
Generative modelling

*Bump hunting = unsupervised learning / generative modelling*

- general idea: high density areas define unique physical events
- prior knowledge: classification boundaries can only occur in low density areas
- we can also do this in higher dimensions
- combine real unlabeled data with labeled (simulated) data: *semi-supervised learning*

https://github.com/TimSalimans/HiggsML/  
tim@algoritmica.nl
Conclusion

- Machine Learning methods are a powerful tool for analyzing HEP data
  - supervised model faster & more accurate than MMC or SVFIT
  - crucially depends on accuracy of simulated data
Conclusion

- Machine Learning methods are a powerful tool for analyzing HEP data
  - supervised model faster & more accurate than MMC or SVFIT
  - crucially depends on accuracy of simulated data
- real promise lies in semi-supervised methods combining simulated data and real data

https://github.com/TimSalimans/HiggsML/  
tim@algoritmica.nl
Conclusion

- Machine Learning methods are a powerful tool for analyzing HEP data
  - supervised model faster & more accurate than MMC or SVFIT
  - crucially depends on accuracy of simulated data
- real promise lies in semi-supervised methods combining simulated data and real data