"Planing" to expose what the machine is learning

ACME
Multi-variate analyzer
Chang, Cohen, and BO [arXiv:1709.10106]

Bryan Ostdiek
2nd IML Machine Learning Workshop, CERN
April 10, 2018
Review: Linear Regression

How to fit data

1. Plot the data
2. Define the function
   - \( f(x, \vec{a}) = a_0 + a_1 x \)
3. Choose how to know what fits best
   - a.k.a. Loss Function
   - MSE: \( L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i, \vec{a}) - y_i)^2 \)
4. Find the minimum error (loss) (cost)
   - \( a_{\text{best}} = a \) when \( \left( \frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \right|_{x, y = 0} = 0 \)
**Review: Linear Regression**

**How to fit data**

1. Plot the data
2. Define the function
   - $f(x, \tilde{a}) = a_0 + a_1 x$
3. Choose how to know what fits best
   - a.k.a. *Loss Function*
     - **MSE:** $L(x, y, \tilde{a}) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i, \tilde{a}) - y_i)^2$
4. Find the minimum error (loss) (cost)
   - $a_{\text{best}} = a$ when \(\frac{\partial L(x, y, \tilde{a})}{\partial \tilde{a}}\bigg|_{x,y} = 0\)
Review: Linear Regression

How to fit data
1. Plot the data
2. Define the function
   - \( f(x, a) = a_0 + a_1 x + a_2 x^2 \)
3. Choose how to know what fits best
   - a.k.a. Loss Function
     - MSE: \( L(x, y, \bar{a}) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i, \bar{a}) - y_i)^2 \)
5. Find the minimum error (loss) (cost)
   - \( a_{\text{best}} = a \) when \( \left. \frac{\partial L(x, y, \bar{a})}{\partial \bar{a}} \right|_{x,y} = 0 \)
Review: Linear Regression

How to fit data

1. Plot the data
2. Define the function
   - \( f(x, \vec{a}) = a_0 + a_1 x + a_2 x^2 \)
3. Choose how to know what fits best
   - a.k.a. Loss Function
   - MSE: \( L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i, \vec{a}) - y_i)^2 \)
4. Find the minimum error (loss) (cost)
   - \( a_{\text{best}} = a \) when \( \left( \frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \right)_{x, y = 0} = 0 \)

Is that good enough? How many parameters can we add?
Review: Linear Regression

How to fit data

1. Plot the data
2. Define the function
   - \( f(x, \vec{a}) = a_0 + a_1 x + a_2 x^2 \)
3. Choose how to know what fits best
   - a.k.a. Loss Function
   - MSE: \( L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i, \vec{a}) - y_i)^2 \)
4. Find the minimum error (loss) (cost)
   - \( a_{\text{best}} = a \) when \( \left. \frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \right|_{x, y = 0} = 0 \)
5. Is that good enough?
   - How many parameters can we add?
What is the machine learning?

Logistic regression similar, but intuition is off/different

Chang, Cohen, and BO [arXiv:1709.10106]
What is the machine learning?

Logistic regression similar, but intuition is off/different

Does the machine learn a circle?

Chang, Cohen, and BO [arXiv:1709.10106]
What is the machine learning?

Logistic regression similar, but intuition is off/different

Chang, Cohen, and BO [arXiv:1709.10106]
What is the machine learning?

Logistic regression similar, but intuition is off/different

It has learned generically where events are in “any” parameter space. Wrong question to ask.

Chang, Cohen, and BO [arXiv:1709.10106]
What is the machine learning?

Can we find what information the machine is learning?
What is the machine learning?

Can we find what information the machine is learning?

Planing

See also de Oliveria, Kagan, Nachman, Schwartzman [arXiv:1511.05190]

(a) Train machine on low level data
(b) Compute low level AUC
(c) Choose a variable: compute (planing) weights
(d) Train machine on weighed (planed) data
(e) Compute planed AUC
(f) Compare: looking for significant performance drop

Removing information from training samples

Similar to what experiments do with different $p_T$ samples
What is the machine learning?

Can we find what information the machine is learning?

Planing

See also de Oliveria, Kagan, Nachman, Schwartzman [arXiv:1511.05190]

(a) Train machine on low level data
(b) Compute low level AUC
(c) Choose a variable: compute (planing) weights
(d) Train machine on weighed (planed) data
(e) Compute planed AUC
(f) Compare: looking for significant performance drop

Saturation


(a) Train network on low level data
(b) Compute low level AUC
(c) Add high level variable
(d) Train new machine using low + high level info
(e) Compute AUC
(f) No performance gain implies information has been learned
Using saturation to pick the network

![Graphs showing ROC curves for low-level and low-level + high-level networks with different numbers of hidden layers and associated AUC values.](image-url)
Using saturation to pick the network

Minimal improvement
Using saturation to pick the network

Minimal improvement

high-level not adding much
Using saturation to pick the network

Choose 3 layer networks for all deep network examples going forward

high-level not adding much  
Minimal improvement
Example with a toy model

Signal Distribution: \( f(\bar{x}) = [\Theta (r_0 - r) + C_r] \cdot [\bar{z} \cdot B_z + C_z] \)

Background Distribution: Uniform

### Results

<table>
<thead>
<tr>
<th>(x, y, z)</th>
<th>r</th>
<th>PLANED</th>
<th>LINEAR AUC</th>
<th>DEEP AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ ☒ ☒ ☒</td>
<td></td>
<td>☒  ☒</td>
<td>0.61275(01)</td>
<td>0.81243(45)</td>
</tr>
<tr>
<td>✓ ✓ ☒ ☒</td>
<td></td>
<td>☒  ✓</td>
<td>0.79672(01)</td>
<td>0.81388(23)</td>
</tr>
<tr>
<td>✓ ☒ r</td>
<td></td>
<td>☒  ✓</td>
<td>0.61030(01)</td>
<td>0.61026(02)</td>
</tr>
<tr>
<td>✓ ☒ (r, z)</td>
<td></td>
<td>☒  ☒</td>
<td>0.5081(16)</td>
<td>0.49998(03)</td>
</tr>
</tbody>
</table>

![Graph showing correlation](Toy.png)
Example with a toy model

Signal Distribution: \( f(\vec{x}) = \left[ \Theta (r_0 - r) + C_r \right] \cdot \left[ z \cdot B_z + C_z \right] \)

Background Distribution: Uniform

**Check Saturation**

<table>
<thead>
<tr>
<th>((x, y, z))</th>
<th>(r) Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓  ✓  ✓</td>
<td>0.61275(01)</td>
<td>0.81243(45)</td>
<td></td>
</tr>
<tr>
<td>✓  ✓  ✓</td>
<td>0.79672(01)</td>
<td>0.81388(23)</td>
<td></td>
</tr>
<tr>
<td>✓  ✓  (r)</td>
<td>0.61030(01)</td>
<td>0.61026(02)</td>
<td></td>
</tr>
<tr>
<td>✓  ✓  (r, z)</td>
<td>0.5081(16)</td>
<td>0.49998(03)</td>
<td></td>
</tr>
</tbody>
</table>

**Results**

Correlation: \( \text{corr} = 1.00 \)

Linear Response vs. Toy

Bryan Ostdiek
Example with a toy model

Signal Distribution: \( f(\bar{x}) = [\Theta (r_0 - r) + C_r] \cdot [\bar{z} \cdot B_z + C_z] \)

Background Distribution: Uniform

Check Saturation

<table>
<thead>
<tr>
<th>(x, y, z) r PLANED</th>
<th>LINEAR AUC</th>
<th>DEEP AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ ✓ ✓</td>
<td>0.61275(01)</td>
<td>0.81243(45)</td>
</tr>
<tr>
<td>✓ ✓ ✓</td>
<td>0.79672(01)</td>
<td>0.81388(23)</td>
</tr>
<tr>
<td>✓ ✗ r</td>
<td>0.61030(01)</td>
<td>0.61026(02)</td>
</tr>
<tr>
<td>✓ ✗ (r, z)</td>
<td>0.5081(16)</td>
<td>0.49998(03)</td>
</tr>
</tbody>
</table>

Results

Correlation: corr = 1.00

Linear Response
Example with a toy model

Signal Distribution: \[ f(\bar{x}) = [\Theta (r_0 - r) + C_r] \cdot [z \cdot B_z + C_z] \]

Background Distribution: Uniform

Check Saturation

Results

<table>
<thead>
<tr>
<th>(x, y, z)</th>
<th>r</th>
<th>Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>0.61275(01)</td>
<td>0.81243(45)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>0.79672(01)</td>
<td>0.81388(23)</td>
</tr>
<tr>
<td>✓</td>
<td>x</td>
<td>r</td>
<td>0.61030(01)</td>
<td>0.61026(02)</td>
</tr>
<tr>
<td>✓</td>
<td>x</td>
<td>(r, z)</td>
<td>0.5081(16)</td>
<td>0.49998(03)</td>
</tr>
</tbody>
</table>

No discrimination left

corr = 1.00
BSM Models

\[ \mathcal{L} \supset Z'_\mu \sum_f Q_f \left( g_L \bar{f} \gamma^\mu P_L f + g_R \bar{f} \gamma^\mu P_R f \right) \]

Vector couplings

\[ g_L = g_R \]

Left-handed couplings

\[ g_R = 0 \]
What is the machine learning?

How much information is there to learn in a given distribution?

Original distributions

Planed distributions

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>PLANED</th>
<th>LINEAR AUC</th>
<th>DEEP AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓  ✓  x</td>
<td>x</td>
<td>0.746221(01)</td>
<td>0.988510(98)</td>
</tr>
<tr>
<td>✓  ✓  x</td>
<td>x</td>
<td>0.938967(01)</td>
<td>0.989007(03)</td>
</tr>
<tr>
<td>✓  x  m</td>
<td>x</td>
<td>0.50550(29)</td>
<td>0.4942(48)</td>
</tr>
</tbody>
</table>
What is the machine learning?

How much information is there to learn in a given distribution?

![Graphs showing original and planned distributions for $m_{ee}$, $y(e^-)$, and $y(e^+)$](images)

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\checkmark$</td>
<td>$\times$</td>
<td>0.746221(01)</td>
<td>0.988510(98)</td>
</tr>
<tr>
<td>$\checkmark$</td>
<td>$\checkmark$</td>
<td>0.938967(01)</td>
<td>0.989007(03)</td>
</tr>
<tr>
<td>$\checkmark$</td>
<td>$m$</td>
<td>0.50550(29)</td>
<td>0.4942(48)</td>
</tr>
</tbody>
</table>

Something more than $m_{ee}$?
What is the machine learning?

How much information is there to learn in a given distribution?

Original distributions

Planed distributions

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>$m$</th>
<th>Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ ✓ x</td>
<td></td>
<td>0.746221(01)</td>
<td>0.988521(98)</td>
<td></td>
</tr>
<tr>
<td>✓ ✓ x</td>
<td>✓</td>
<td>0.938967(01)</td>
<td>0.989007(03)</td>
<td></td>
</tr>
<tr>
<td>✓ x m</td>
<td></td>
<td>0.505560(29)</td>
<td>0.494242(48)</td>
<td></td>
</tr>
</tbody>
</table>

Something more than $m_{ee}$?
What is the machine learning?

How much information is there to learn in a given distribution?

Original distributions

Planed distributions

<table>
<thead>
<tr>
<th>$(E, \bar{\rho})$</th>
<th>$m$ Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✗</td>
<td>0.746221(01)</td>
<td>0.988510(98)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>0.938967(01)</td>
<td>0.989007(03)</td>
</tr>
<tr>
<td>✓</td>
<td>✗ $m$</td>
<td>0.50550(29)</td>
<td>0.4942(48)</td>
</tr>
</tbody>
</table>

Something more than $m_{ee}$?

NO!
What is the machine learning?

How much information is there to learn in a given distribution?

Original distributions

Planed distributions

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>$m$</th>
<th>Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓  ✔  ✔  ❌</td>
<td>0.746221(01)</td>
<td>0.988510(98)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓  ✓  ✗</td>
<td>0.938967(01)</td>
<td>0.989007(03)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓  ❌  ✗</td>
<td>0.50550(29)</td>
<td>0.4942(48)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>$m$</th>
<th>Planed</th>
<th>Linear AUC</th>
<th>Deep AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓  ✗  ✗</td>
<td>0.763280(05)</td>
<td>0.989353(59)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓  ✓  ✗</td>
<td>0.942004(02)</td>
<td>0.989826(10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓  ✗  $m$</td>
<td>0.626648(28)</td>
<td>0.6258(24)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓  ✗  $(m, \Delta</td>
<td>y</td>
<td>)$</td>
<td>0.52421(15)</td>
<td>0.5320(25)</td>
</tr>
</tbody>
</table>
What is the machine learning?

How much information is there to learn in a given distribution?

Original distributions

Planed distributions

More Information

Linear

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>$m$</th>
<th>PLANTED</th>
<th>LINEAR AUC</th>
<th>DEEP AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.746221(01)</td>
<td>0.988510(98)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.938967(01)</td>
<td>0.989007(03)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>$m$</td>
<td>0.50550(29)</td>
<td>0.4942(48)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$(E, \vec{p})$</th>
<th>$m$</th>
<th>PLANTED</th>
<th>LINEAR AUC</th>
<th>DEEP AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.763280(05)</td>
<td>0.989353(59)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.942004(02)</td>
<td>0.989826(10)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>$m$</td>
<td>0.626648(28)</td>
<td>0.6258(24)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>$(m, \Delta</td>
<td>y</td>
<td>)$</td>
</tr>
</tbody>
</table>
What is the machine learning?
What is the machine learning?

Tool for dimensionality reduction
What is the machine learning?

Tool for dimensionality reduction

Using only mass: AUC = 0.939, ACC = 0.937
Using both: AUC = 0.989, ACC = 0.958
"Planing" to expose what the machine is learning

Conclusion

- Iteratively remove information to see what the machine needs to learn
- Process allows for simple way to see if discriminating power is linear or not

Future Directions

- Apply to more realistic setting
- What if best variables are unknown?
- What if many iterations are needed?