Closing Thoughts on Machine Learning (ML in Practice)
Closing Thoughts on Machine Learning
(ML in Practice)
When someone asks “What is Machine Learning?”

“Learning is any process by which a system improves performance from experience.” - Herbert Simon

Definition by Tom Mitchell (1998):
Machine Learning is the study of algorithms that improve their performance $P$ at some task $T$ with experience $E$.
A well-defined learning task is given by $<P, T, E>$. 
Machine Learning

Unsupervised Learning

Supervised Learning

Reinforcement Learning

What happened?

Predictive Analytics

How can we make it happen?

Prescriptive Analytics

Value

Difficulty

Insight

Optimization

Foresight

Hindsight

Information

Machine Learning
Your Test Data is Sacred!

• Spit data into:
  • Training set
  • Validation (development) set
  • Test set
Your Test Data is Sacred!

• Spit data into:
  • Training set
  • Validation (development) set
  • Test set

• Caution
  • Trajectory (Sequential) Data
  • Groupable data (partial exchangeability)
Your Test Data is Sacred!

• Spit data into:
  • Training set
  • Validation (development) set
  • Test set

• **Caution**
  • Trajectory (Sequential) Data
  • Groupable data (partial exchangeability)

• Rule of thumb:
  • 80% Training
  • 10% Validation
  • 10% Test
Understand Your Data

• Typical Problems
  - All features are zero/constant
  - Some features have large value (due to noise, etc.)

• Visualize the data
  - Histogram each feature
  - Scatter plot (pairs/triplets) of features
  - Perform PCA/MDS first and then scatter plot the projected data
Understand Your Data

- **Typical Problems**
  - All features are zero/constant
  - Some features have large value (due to noise, etc.)
Understand Your Data

• Typical Problems
  • All features are zero/constant
  • Some features have large value (due to noise, etc.)

• Visualize the data
  • Histogram each feature
  • Scatter plot (pairs/triplets) of features
  • Perform PCA/MDS first and then scatter plot the projected data
Don’t Use Labels as Features
Normalize Data

- Mean center, scale variance of each feature
- Min-Max scaling to \([-1,1]\]
- **Whiten** the Data (mean center, identity covariance)
“More Data is Better”
Understand What Features Matter

• Naively: Use all features. (adding features should never hurt)
Understand What Features Matter

• Naively: Use all features. (adding features should never hurt)
  except..
  • Computational problems
  • Overfitting problems
  • Memory problems
  .
  .
  .
Understand What Features Matter

• Use all features. (adding features should never hurt)

Select features (Greedy Backward Selection)
• Measure performance of all combinations of all but one feature on development set
• Remove least important one
• Iterate
Understand What Features Matter

• Use all features. (adding features should never hurt)

Select features one at a time (Greedy Forward Selection)
  • Measure performance of all features individually
  • Include the most important one
  • Iterate
Never Underestimate the Power of a Linear Predictor

Monocular UAV Control

Simulating Video Textures
Never Underestimate the Power of a Linear Predictor

Monocular UAV Control

Simulating Video Textures
Never Underestimate the Power of a Linear Predictor

- Linear Regression
- Logistic Regression
- Linear SVMs
- Linear Dynamical System / Kalman Filter
Never Underestimate the Power of “Well Tuned” Gradient Descent
## Unbalanced Classes

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Class1</th>
<th>Class2</th>
<th>Class3</th>
<th>Class4</th>
<th>LLC*</th>
<th>RLC*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small dataset</td>
<td>15969</td>
<td>87310</td>
<td>525</td>
<td>809</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>Large dataset</td>
<td>311070</td>
<td>293079</td>
<td>10598</td>
<td>10082</td>
<td>213</td>
<td>206</td>
</tr>
</tbody>
</table>
Unbalanced Classes

Approaches to tackle this:

• Downsample the overrepresented classes
  • Downside: Doesn’t use all the data
Unbalanced Classes

Approaches to tackle this:

• Downsample the overrepresented classes
  • Downside: Doesn’t use all the data

• Upsample (duplicate data)
  • Downside: overfitting, computational overhead.
Unbalanced Classes

Approaches to tackle this:

- Downsample the overrepresented classes
  - Downside: Doesn’t use all the data

- Upsample (duplicate data)
  - Downside: overfitting, computational overhead.

- Weight the samples (scale loss)
  - Downside: might not help if using stochastic gradient descent
Overfitting Vs. Underfitting

• Overfitting

  You do great on training data, but relatively poorly on test data
Overfitting Vs. Underfitting

• Overfitting

  You do great on training data, but relatively poorly on test data

  • Test on validation/development set
Overfitting Vs. Underfitting

- Overfitting

  You do great on training data, but relatively poorly on test data

  - Test on validation/development set
  - Regularize more
Overfitting Vs. Underfitting

• Overfitting

You do great on training data, but relatively poorly on test data

• Test on validation/development set
• Regularize more
• Feature selection (fewer features)
  • stepwise
  • PCA
  • think hard!
Overfitting Vs. Underfitting

• Overfitting

You do great on training data, but relatively poorly on test data

• Test on validation/development set
• Regularize more
• Feature selection (fewer features)
  • stepwise
  • PCA
  • think hard!
• Get more data
Overfitting Vs. Underfitting

• Overfitting

You do great on training data, but relatively poorly on test data

• Test on validation/development set
• Regularize more
• Feature selection (fewer features)
  • stepwise
  • PCA
  • think hard!
• Get more data
• Use simpler classifier
Overfitting Vs. Underfitting

• Underfitting

  You perform poorly on both training and test data
Overfitting Vs. Underfitting

• Underfitting

  You perform poorly on both training and test data

  • Use more features
Overfitting Vs. Underfitting

• Underfitting

  You perform poorly on both training and test data
  
  • Use more features
  • More sophisticated classifier
Overfitting Vs. Underfitting

• Underfitting

You perform poorly on both training and test data

• Use more features
• More sophisticated classifier
• Kernelize (infinite features!!)
Overfitting Vs. Underfitting

• Underfitting

You perform poorly on both training and test data

• Use more features
• More sophisticated classifier
• Kernelize (infinite features!!)
• Regularize less
Overfitting Vs. Underfitting

• Underfitting

You perform poorly on both training and test data

• Use more features
• More sophisticated classifier
• Kernelize (infinite features!!)
• Regularize less
• Optimize better
Dima’s Rule:

“Always start by overfitting” - Cris Dima
Building Large Learning Systems

• Avoid premature statistical optimization:
  • Spend time on the parts that matter
  • Unit test learning code
  • Think a lot about features and data
Don’t buy the Hype!

- Rich Caruana Papers
  (large empirical comparisons of different methods)

- Simple models

- SVMs

- Ensemble Methods (e.g. Random Forests)