Practical Advances in Machine Learning: A Computer Science Perspective

Scott Neal Reilly & Jeff Druce
Charles River Analytics

Prepared for 2017 Workshop on Data Science and String Theory

November 30 – December 1, 2017
Objectives of this breakout session

- Quick review of machine learning “from a CS perspective”
- Review some of the latest advances in machine learning
- Tips for using ML
- Discussion of academic/industrial collaboration opportunities/challenges
- Discussion about all of the above
Introductions

- **Charles River Analytics**
  - 160 people, 30-year history
  - Mostly government contract R&D
  - AI, ML, robotics, computer vision, human sensing, computational social science, human factors

- **Scott Neal Reilly**
  - PhD, Computer Science, Carnegie Mellon University
  - Senior Vice President & Principal Scientist, Charles River Analytics
  - Focus on ensemble machine learning and causal learning

- **Jeff Druce**
  - PhD, Civil Engineering, University of Minnesota
  - BS, Applied Math and Physics, University of Michigan
  - Scientist, Charles River Analytics
  - Focus on deep learning, GANs, signal processing+ML
Question: What can machine learning do for me?
Machine learning is about getting computers to perform tasks that I don’t want to or don’t know how to tell them to do.

What kinds of tasks?

How do they learn if I don’t tell them?
Dimensions of a Machine Learning Problem

• **Dimension #1: Data**
  - What kind of data do I have?
  - What are the properties of the data?

• **Dimension #2: Objective/Task**
  - What is it that is being learned?
  - What are the computational/time constraints on learning/execution?

• These tend to suggest particular techniques
Dimension #1: Data

- Sub-Dimension #1: *What kind of data do I have?*
  - Labeled: supervised
Dimension #1: Data

- Sub-Dimension #1: *What kind of data do I have?*
  - Labeled: supervised
  - **Unlabeled: unsupervised**
Dimension #1: Data

- Sub-Dimension #1: *What kind of data do I have?*
  - Labeled: supervised
  - Unlabeled: unsupervised
  - Partially labeled: semi-supervised
Dimension #1: Data

• Sub-Dimension #1: *What kind of data do I have?*
  • Labeled: supervised
  • Unlabeled: unsupervised
  • **Partially labeled: semi-supervised**
Dimension #1: Data

- Sub-Dimension #1: What kind of data do I have?
  - Labeled: supervised
  - Unlabeled: unsupervised
  - Partially labeled: semi-supervised

- An environment that can label data for you: exploratory
  - Active learning, Reinforcement learning
Dimension #1: Data

- Sub-Dimension #1: *What kind of data do I have?*
  - Labeled: supervised
  - Unlabeled: unsupervised
  - Partially labeled: semi-supervised
- An environment that can label data for you: exploratory
  - Active learning, Reinforcement learning
Dimension #1: Data

• Sub-Dimension #2: *What are the properties of the data?*
  • How much is there?
  • How noisy is it?
  • How many features are there?
Dimension #2: What is the learning task?

- **Classification**
  - Given features of X, what is X?
  - Supervised, unsupervised, semi-supervised, etc.

- **Regression**
  - Given features of X, what is value of feature Y?
  - Linear regression, symbolic regression/genetic programming, etc.

- **Dimensionality reduction**
  - Given features of X, can I describe X with fewer features that are comparably descriptive?
  - Principal component analysis, latent Dirichlet allocation, etc.

- **Anomaly detection**
  - Given features of X, is X unusual given other X’s?
  - Principal component analysis, support vector machines, etc.

- **Process learning**
  - Given task T, how do I decide what action A (or plan P) will accomplish T?
  - Reinforcement learning, genetic programming, RNNs, etc.

- **Structure learning**
  - Given variables V, how do they relate to each other?
  - Statistical relational learning, etc.

- **Model learning**
  - Discriminative vs. generative models
  - Learn $p(\text{class}|\text{features})$ or $p(\text{features}|\text{class})$ respectively.
Some Approaches to ML

- Given what data is available and the task, pick from...
  - Neural Nets / Deep Learning
  - Bayesian Learning
  - Statistical Relational Learning
  - Symbolic/rule learning
  - Reinforcement Learning
  - Genetic programming
  - Other Approaches
    - kNN, svm, logistic regression, decision trees/forests
Question: What are some of the interesting recent advances in machine learning?
Advance #1: Deep Learning

Convolutional Neural Networks
Deep Reinforcement Learning
Generative Adversarial Networks
Convolutional Neural Networks

- In traditional image/signal processing and learning problems, human crafted features are used to transform the images into more informative space.
- However, using human-designed features does not leverage the computational power of modern day computers/GPUs!
- To perform better classification, we let a deep neural network learn optimal features that can best separate the data.
In traditional image/signal processing and learning problems, human crafted features are used to transform the images into more informative space.

However, using human-designed features does not leverage the computational power of modern day computers/GPUs!

To perform better classification, we let a deep neural network learn optimal features that can best separate the data.
Convolutional Neural Networks

- In traditional image/signal processing and learning problems, human crafted features are used to transform the images into more informative space.
- However, using human-designed features does not leverage the computational power of modern day computers/GPUs!
- To perform better classification, we let a deep neural network *learn* optimal features that can best separate the data.
Fully Convolutional Networks

Fully Convolutional Networks for Segmentation

forward/inference

backward/learning

pixelwise prediction

segmentation g.t.
CNNs for non-image problems

Natural Language Processing – Text Classification
CNNs for non-image problems

Natural Language Processing – Text Classification

Signal Processing – Stereotypical Motor Movement Detection in Autism

\[ \text{CNN}(d, c, F) \]
What do all problems where leveraging CNNs is effective have in common?

CNNs mine high dimensional data where *proximal* input features possess some structure which can be exploited to achieve some task.
What do all problems where leveraging CNNs is effective have in common?

CNNs mine high dimensional data where *proximal* input features possess some structure which can be exploited to achieve some task.

- Lots of proximal structure!
- What problems are you facing where subtle, complex, embedded local structures could potentially be exploited?
Advance #1: Deep Learning

Convolutional Neural Networks
Deep Reinforcement Learning
Generative Adversarial Networks
Reinforcement Learning

Policy

\[ \pi : S \times A \rightarrow [0, 1] \]

\[ \pi(a|s) = P(a_t = a|s_t = s) \]
Reinforcement Learning

Policy

\[ \pi : S \times A \rightarrow [0, 1] \]
\[ \pi(a|s) = P(a_t = a|s_t = s) \]

How can we learn an optimal policy to achieve the goal?

Goal

Observed State

Agent

Charles River Analytics
Deep Reinforcement Learning

Episodes

Agent

Environment

state $s_t$

reward $r_t$

action $a_t$

$r_{t+1}$

$s_{t+1}$
Deep Reinforcement Learning

- Learn the best policy through a series of training episodes.
- Training uses an action-value function (aka Q function), or the expected return for following some policy.
Deep Reinforcement Learning (Q learning)

- Learn the best policy through a series of training episodes.
- Training uses an action-value function (aka Q function), or the expected return for following some policy.
- Traditionally, a linear function was used, DRL uses a deep net to approximate Q.
DRL Successes

Bots are now the world champion in...

A variety of Atari games - Mnih

Go - AlphaZero

Dota 2 - Deepmind
DRL Successes

Bots are now the world champion in...

A variety of Atari games - Mnih

Go - AlphaZero

Dota 2 - Deepmind

Is DRL only good for games?
DRL – What can it do?

- Natural Language Processing
- Text Generation
- Understanding Deep Learning: Daniely et al. (2016)
- Deep Probabilistic Programming, Tran et al. (2017)
- Machine Translation: He et al. (2016a)
- Building Compact Networks
DRL – What can it do?

- Natural Language Processing
- Text Generation
- Understanding Deep Learning: Daniely et al. (2016)
- Deep Probabilistic Programming, Tran et al. (2017)
- Machine Translation: He et al. (2016a)
- Building Compact Networks

DRL can be used where a large, diverse state space makes it difficult to explore all possible strategies, and actions may have latent effects, which at some point become very important in achieving a task.
Advance #1: Deep Learning

Convolutional Neural Networks
Deep Reinforcement Learning
Generative Adversarial Networks
The example: We can think of G as a counterfeiter attempting to produce fake money such that they can not be detected by the discriminative false currency detecting agent D.
What are GANs? – Improving G

1. Training Set
   - \( \hat{p}_{data} \)

2. \( D_1 \)
   - Gives probability sample is real
   - \( G_1 \) Produces sample
   - \( p_{g_1} \)

3. \( D_2 \)
   - Gives probability sample is real
   - Update G
   - \( G_2 \) Produces sample
   - \( p_{g_2} \)

4. \( \ldots \)
   - Update G
   - \( G_3 \) Produces sample
   - \( p_{g_3} \)

5. \( \ldots \)

6. \( \ldots \)
   - \( G^F \)
   - \( p_{model} \)
GANS – Successes

Noise Input

GANs

HD Face Generation

Next Frame Prediction

Text to Image Generation

Ground Truth

MSE

Adversarial

charles river analytics
GANS – What can they do?

- In just a short time, GANS have proven to be an extremely ripe area for research
- Image, music, are generation
- Superresolution
- Domain transformation (sketch <-> photo, satellite -> map)
- Advanced malware software training

GANS can be used where one wants to sample from a complex distribution which describes the structure of some training set, but produces novel instances.
Advance #2: Probabilistic Programming
Probabilistic Reasoning: The Gist

Probabilistic model expresses general knowledge about a situation.
Evidence contains specific information about a situation.
Queries express things that will help you make a decision.
Answers to queries are framed as probabilities of different outcomes.
Probabilistic Reasoning: Predicting the Future

Probabilistic Reasoning System

Corner Kick Model

Inference Algorithm

Evidence:
- Tall center forward
- Inexperienced goalie
- Strong wind

Query:
- Will a goal be scored?

Answer:
- Yes 20%
- No 80%
Probabilistic Reasoning: Inferring Factors that Caused Obs.

Evidence:
- Tall center forward
- Inexperienced goalie
- Strong wind
- A goal was scored

Query:
- How good is the goalie?

Answer:
- Poor 50%
- Average 30%
- Good 20%
Probabilistic Reasoning: Using the Past for Prediction

The evidence contains knowledge of:

- Preconditions and outcomes of previous situations
- Preconditions of the current situation

![Diagram](diagram.png)

- **Evidence:**
  - Tall center forward
  - Inexperienced goalie
  - Strong wind
  - A goal was scored last time
  - Weak wind this time

- **Query:**
  - Will a goal be scored this time?

- **Answer:**
  - Yes 15%
  - No 85%
Limitations on Probabilistic Reasoning Systems

- The “Corner Kick Model”
  - Not object oriented
  - No recursion
  - No loops
  - No way to integrate complex simulation models

- The “Inference Algorithm”
  - There is no such thing
  - There are lots of them with different properties

- Hard to use in larger systems
Probabilistic Programming Languages

- Figaro!
  - https://github.com/p2t2/figaro

- Your model is a program
  - Figaro is built on Scala
  - Loops, recursion, objects
  - You can pick an included inference algorithm or let the system pick
  - Integration easy in both directions
    - E.g., deep net integration is an active area of exploration
Advance #3: Ensemble Machine Learning
• Sometimes there is no algorithm that alone does what you need

• Sometimes there is, but you don’t know what it is

• What to do then?
  • Ensemble machine learning has shown that it can often outperform any individual ML technique
  • Think hurricane tracking
Types of Ensembles

(a) Data ensembles

(b) Chain ensembles

(c) Technique ensembles

(d) Nested ensembles
Enhanced Technique Ensembles
Advance #4: Explainable Machine Learning
Advance #4: Explainable Machine Learning

- From before, letting the network pick what features are used leads to enhanced performance.
- However, what guarantees are there that the features learned by the network will be human interpretable?

Automated Feature Extraction (CNN)
Advance #4: Explainable Machine Learning

From before, letting the network pick what features are used leads to enhanced performance.

However, what guarantees are there that the features learned by the network will be human interpretable?

Answer: Nothing!

Automated Feature Extraction (CNN)

This problem is not confined to CNNs, opaqueness is a problem across many areas in DL (and ML/AI in general).

Coming into effect in Europe in 2018: [General Data Protection Regulation (GDPR)](https://gdpr.eu)
How can ML explain itself?

- Visual based methods (for CNNs)

  - Attention Maps
  - Saliency Maps
  - Gradient Maps
CRA approach for explainable ML

- Develop a *causal model* of the network in question
- Probe network for human understandable concepts – look at *activations*
- Use interventions to demonstrate causality
  - In other approaches, it can be easy to “hallucinate” a cause-effect relationship

---

Causal Model - Pipeline

- ML System
- Classification
- Concept Inference
- Concepts
- Causal Learning
- Internal Data
Example Causal Model (Causal Graphical Model)
Pedestrian Detection: Causal Learning

<table>
<thead>
<tr>
<th>Intervention</th>
<th>None</th>
<th>Pedestrian</th>
<th>Outline</th>
<th>Color</th>
<th>Color+Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 3 Activation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>110</td>
<td>182</td>
</tr>
</tbody>
</table>

Average activation for positive images: 197
These are Pedestrians (according to Node 3)

**Activation maximization**

**Adversarial image**
(selected based on activation maximization analysis)
Explainable AI – What can it do?

• Back out information on why a typical “black box” algorithm is doing what it’s doing
• Give augmented example based explanations (typical in imagery)
• For CRA, back out the what human understandable concepts a network is using, and quantifying the importance of those features in some task.

Explainable AI is relevant when not only high performance is desired, but also a testable and explorable framework for understanding what the AI is doing “behind the curtain”.
Question: How do computer scientists determine which techniques to apply to one type of problem vs another?
How to choose your algorithm: Some tips (1 of 2)

- What kind of problem is it? (Classification vs. regression vs. ...)
  - Multiclass? Multilabel?
  - Can your learning method handle this?

- What kind of data do you have available?
  - Could you label some unlabeled data and go semi-supervised?
  - Can you explore the world and use active learning or RL?

- How much data do you have?
  - Deep learning only works with sufficient data
  - Can you augment the data you have?

- How much human expertise is available?
  - Can you quantify the uncertainty in this knowledge?

- What computational horsepower do you have at your disposal?
  - RAM limited? GPUs?
How to choose your algorithm: Some tips (2 of 2)

- How noisy is your data? Do you have missing values? How is the data encoded? Are “semantics” misaligned?
  - Data prep is CRITICAL in machine learning!
  - Sometimes data science techniques help
  - Sometimes machine learning can itself help

- Is your data mixed? E.g., double and categorical
  - Can the data be reasonably converted?

- What kinds of relationships do we expect to find?
  - Linear? Non-linear?
  - What kind of non-linear are plausible?

- Is explainability important or just performance?
  - E.g., decision trees are implicitly somewhat interpretable, CNNs are not

- Remember ensembles. Maybe you don’t have to choose!
Microsoft Azure Machine Learning: Algorithm Cheat Sheet

**ANOMALY DETECTION**
- One-class SVM: >100 features, aggressive boundary
- PCA-based anomaly detection: Fast training

**CLUSTERING**
- K-means: Finding unusual data points
- Discovering structure: Three or more
- Predicting categories: Two

**MULTICLASS CLASSIFICATION**
- Fast training, linear model: Multiclass logistic regression
- Accuracy, long training times: Multiclass neural network
- Accuracy, fast training: Multiclass decision forest
- Accuracy, small memory footprint: Multiclass decision jungle
- Depends on the two-class classifier, see notes below: One-vs-all multiclass

**REGRESSION**
- Ordinal regression: Data in rank ordered categories
- Poisson regression: Predicting event counts
- Fast forest quantile regression: Predicting a distribution
- Linear regression: Fast training, linear model
- Bayesian linear regression: Linear model, small data sets
- Neural network regression: Accuracy, long training time
- Decision forest regression: Accuracy, fast training
- Boosted decision tree regression: Accuracy, fast training

**TWO-CLASS CLASSIFICATION**
- Two-class SVM: >100 features, linear model
- Two-class averaged perceptron: Fast training, linear model
- Two-class logistic regression: Fast training, linear model
- Two-class Bayes point machine: Fast training, linear model

© 2015 Microsoft Corporation. All rights reserved. Created by the Azure Machine Learning Team Email: AzurePoster@microsoft.com Download this poster: http://aka.ms/MLCheatsheet
Question: What are the circumstances under which it benefits industry to partner with academics?
Academic-Industry Collaborations

- Industry-prime / Academic-sub
  - We do this all the time
  - Academics bring cutting-edge ideas we want to build from
  - Academics can bring domain expertise that we lack
    - Note: We have almost no domain expertise in anything our customers care about.
  - Can help us open up new customers, research areas, business

- Academic-prime / Industry-sub
  - Industrial research labs can feel academic in many ways
  - Though tend to be more team-focused than MURIs (Multidisciplinary University Research Initiatives)
  - If the company has relevant capabilities, invite them to the team
  - We are happy to publish research
  - One concern: most companies are for-profit
Question: What do you want to talk about now?
Discussion questions

• Which ML techniques do you currently use?
  • What are the challenges associated with those techniques?
  • How do we know if the technique is working or not?

• How do you know if you have enough / good-enough data?
  • Can the preexisting data be augmented?
  • Can expert knowledge be incorporated?

• Which ML tools do you currently use?
  • What are the challenges associated with those tools?

• What are the biggest challenges associated with applying ML to string theory problems?

• What are the string theory problems (in layman’s terms if possible!) that are most appropriate for ML to help with?

• What kinds of university-industry collaborations have you engaged in? What worked well or didn’t work so well?