Research Opportunities in AutoML

Rich Caruana

Microsoft Research
Motivation
75% of Machine Learning is preparing to do machine learning
and 15% is what you do afterwards…

Preparing to do machine learning

75% of Machine Learning
75% of Machine Learning is preparing to do machine learning and 15% is what you do afterwards...

most ML research about the middle 10%
75% of Machine Learning is preparing to do machine learning and 15% is what you do afterwards.

Most ML research about the middle 10%.
Goals for This Talk

For This Talk
• Foster research on the complete ML pipeline in lab and ML in the field
• How/Where do we start?

Ultimate goal is to make the practice of ML more reliable so you don’t need a Ph.D. in ML + 10 years experience to do ML well

Suggest future challenge/competition problems

Describe a few open problems in AutoML

Foster research on the complete ML pipeline

Start by looking at difference between ML in lab and ML in the field
Real-World Engineering vs. UC-Irvine/CIFAR ML Research
UC-Irvine/CIFAR vs. Real-World
UC-Irvine/CIFAR vs. Real-World

- Sometimes add data
- Until doing well on metric
- Change alg's, params, and coding
- Metric(s) pre-defined
- Know how well others did
- No collection, cleaning, ...
- Download data
UC-Irvine/CIFAR

Real-World

vs.

Download data

• No collection, cleaning, ...

• Metric(s) pre-defined

• Change algs, params, and coding

• Sometimes add data

• Until doing well on metric

• Problem undefined

• Know how well others did

• New features and data feeds

• Add new features and data feeds

• Don’t know how well you can do

• Most effort goes into the data

• Clean, clean, clean

• Coding of data is critical

• Choose practical algorithms

• Debug, debug, debug

• Wash, rinse, repeat

• Month after month after month!
Surprisingly, the research pipeline is complex because we assume the researcher is an expert.
Machine Learning (Research) Pipeline

- Modify Algorithm
- Up-sample Minority Class
- Feature Selection
- Re-code Features
- Discover Leakage
- Check Calibration
- Deal with Missing Values
- Tweak Hyper Parameters

Machine Learning Pipeline
Machine Learning (Engineering) Pipeline

• By real engineers, teams of engineers, ...
• On real data, to real metrics, ...
• On schedule, on budget, ...
• Must be maintainable, repeatable, documentable, ...

Machine Learning (Engineering) Pipeline
Machine Learning (Engineering) Pipeline
Machine Learning (Engineering) Pipeline

1. Problem Definition
2. Data Collection
3. Data Cleaning
4. Data Coding
5. Metric Selection
6. Algorithm Selection
7. Parameter Optimization
8. Post-Processing
9. Deployment
10. Online Evaluation
11. Debug
Each step in the pipeline is an opportunity to do AutoML research.
Future AutoML (Engineering) Pipeline
Goals for this talk

- Foster research on the complete ML pipeline
- Describe a few open problems in AutoML
- Suggest future challenge/competition problems
- So let’s just jump in…
Importance of Hyper-Parameter Optimization

• Hyper-Parameter Optimization is most mature subarea in AutoML

• Manual heuristic search: surprisingly sub-optimal

• Grid search: effective with small number of parameters

• Random search: better than grid with larger number of parameters

• Bayesian Optimization: better than random with very large # parameters

With modern algorithms (boosting, deep neural nets, ...) parameter optimization is much more critical than you might think...

… because modern high-flying algorithms are all low-bias, high variance

How many people here use automatic hyper-parameter optimization?
Importance of Hyper-Parameter Optimization

Around 2000-2005, some thought supervised learning was done

Quiz: they thought best algorithm was:

- SVMs?
- Random Forests?
- Boosting?
- Neural Nets?

• SVMs
• Random Forests
• Boosting
• Neural Nets

Importance of Hyper-Parameter Optimization
Importance of Hyper-Parameter Optimization

Around 2000-2005, some thought Supervised Learning was done:

- Boosting?
- Random Forests?
- Neural Nets?
- SVMs?

Quiz: they thought best algorithm was:
Importance of Hyper-parameter Optimization

- Importance of Hyper-parameter Optimization

- Our results: +1-4% on DNNs by doing careful Bayesian Optimization

- TIMIT benefits from careful hyper-parameter optimization

- Why didn't deep nets get discovered in mid 90's?

- Didn't explore the space and hyper-parameters thoroughly enough?

- Why didn't deep nets get discovered in mid 90's?

- Best DNN on CIFAR-10 and -100 use massive parameter optimization

- Bing Ranker: FastRank vs. NeuralNet Ranker (circa 2010)

- SVMs (circa 2000-2005)

- Bing Ranker: FastRank vs. NeuralNet Ranker (circa 2010)
## ML Algorithm is an Important Hyper-Parameter

- **Threshold Metrics**
  - Probability Metrics
  - Model Accuracy
  - F-Score
  - Lift
  - ROC Area
  - Break Even Point
  - Squared Error
  - Cross Entropy
- **Rank/Ordering Metrics**
  - Average Precision
  - Best
  - Mean

### Table of Performance Metrics

<table>
<thead>
<tr>
<th>Model</th>
<th>F-Score</th>
<th>Lift</th>
<th>ROC Area</th>
<th>Accuracy</th>
<th>Best</th>
<th>Mean</th>
<th>Squared Error</th>
<th>Cross Entropy</th>
<th>Probability Metrics</th>
<th>Threshold Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>0.536</td>
<td>0.161</td>
<td>0.919</td>
<td>0.615</td>
<td>0.786</td>
<td>0.393</td>
<td>0.732</td>
<td>0.733</td>
<td>0.733</td>
<td>0.733</td>
</tr>
<tr>
<td>Logistic Reg</td>
<td>0.696</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>DT</td>
<td>0.742</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>BST-STRMP</td>
<td>0.742</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>KNN</td>
<td>0.742</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>BAG-DT</td>
<td>0.892</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>SVN</td>
<td>0.892</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>ANN</td>
<td>0.892</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>RAND-FOREST</td>
<td>0.892</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>BST-DT</td>
<td>0.892</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
<tr>
<td>BEST</td>
<td>0.892</td>
<td>0.884</td>
<td>0.844</td>
<td>0.719</td>
<td>0.961</td>
<td>0.277</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
<td>0.682</td>
</tr>
</tbody>
</table>
Importance of Hyper-Parameter Optimization

- Hyper-parameter optimization is an example of what AutoML can achieve.
- 20 years ago, selecting hyper-parameters were part of the craft of machine learning.
- Now, multiple papers and algorithms for hyper-parameter optimization exist.
- Knowing how to select hyper-parameters is part of what makes you an expert.
- Makes a significant difference in the accuracy of trained models.
- Thriving research community with multiple workshops.
- Open source code.

Need to view other steps in ML pipeline as new research opportunities.

- Neural nets: number of hidden units, learning rate, momentum, ...
Future AutoML (Engineering) Pipeline

1. Problem Definition
2. Data Collection
3. Data Cleaning
4. Data Coding
5. Metric Selection
6. Algorithm Selection
7. Parameter Optimization
8. Post-Processing
9. Deployment
10. Online Evaluation
11. Debug
Tools to Better Understand Data
NEVER Trust the DB/Data Spec!!
AutoML Tools to Better Understand Data

- Auto variable type determination
  - 0, 1, 2, 3, 4, 5: nominal, ordinal, integer, continuous?
  - Are there dates in fields?
  - Is a field a unique identifier or sequence number?

- Auto anomaly detection
  - Can’t just try everything -- missing variables often cause leakage!

- Auto missing value detector
  - 0, 1, -1, 0, +1
  - 0, 1, 2

- Auto missing value detector
  - Different coding needed for NNs, SVMs, KNN vs. decision tree-based methods

- Auto coding
  - Is a field a unique identifier or sequence number?
  - Are there dates in fields?

- Auto variable type determination
  - 0, 1, 2, 3, 4, 5: nominal, ordinal, integer, continuous?
First Real Data Set I Worked With (1995)

- Pneumonia data from 1992-1995
- 14,199 patients
- < 200 features
- Mix of Booleans, categorical, and continuous variables
- Missing values
- MAR -- Missing At Random
- Missing values correlated with target class (caused leakage?)
- Quickly wrote simple unix utility to help better understand the data

 (> 200 features
 14,199 patients
 Pneumonia data from 1992-1995)
colstats demo...
DataDiff

• Automatically recognize changes in data
  • Changes in DB design, broken sensors, new semantics, new feeds, …
• In real world, DBs and data sources are living, breathing, evolving entities
  • Humans make mistakes, forget what they did 1st time, retire, …
  • In large DBs and data sources are living, breathing, evolving entities
• Changes in DB design, broken sensors, new semantics, new feeds, …
• Don’t care about simple drift if learned model can handle it
• Density estimation is hard in high dimensions (but this is a special case)
• Don’t care about simple drift if learned model can handle it
• Don’t care about simple drift if learned model can handle it
• Don’t care about simple drift if learned model can handle it

Warning flags, default to more robust model, auto-retrain/adapt, …

Care most about changes that affect model accuracy or utility
• C-section 1993-1995 vs. 1996-1998 data (missing values recoded, …)
• 30-day hospital re-admission 2011-2013 vs. new 2014 sample

Density estimation is not as trivial as it might seem:
• Example from 50-50 male-female to 40-60 male-female
• E.g., from 50-50 male-female to 40-60 male-female
• E.g., from 50-50 male-female to 40-60 male-female

Humans make mistakes, forget what they did 1st time, retire, …
• In real world, DBs and data sources are living, breathing, evolving entities

Care most about changes that affect model accuracy or utility
• Warning flags, default to more robust model, auto-retrain/adapt, …

DataDiff
Model Protection

Rappers
• Model trained to predict 30-day readmission was deployed at a children's hospital

Real-world:
• The real-world is always changing
• Test data often looks very different from train data
• Model should know data it was trained on and raise red flags when it detects run-time (test) data looks meaningfully different
• Can make this part of standard practice

Model Protection Wrappers
If you train a model on patient data, and the model is used to change practice of medicine (intervention), next time you collect data it is affected by model… so how do you collect unbiased data 2nd time around?

This is a deep, fundamental problem that in some domains is not easy to solve (ethically, or efficiently). --- problem with non-causal learning features but at labels and relationship between inputs and outputs could approach this as a Diff problem that looks not just at input

And model is used to change practice of medicine (intervention)
Future AutoML (Engineering) Pipeline
Post-Processing: Calibrated Probabilities

- Probabilities make complex systems easier to engineer
- Uniform language that is easy to explain/understand
- Consistent from rev to rev (eliminates threshold effects)
- Uniform language that is easy to explain/understand
- Make complex systems easier to engineer

Where do probabilities come from?
- Careful choice of learning algorithm?
- Most learning algorithms do NOT generate good probabilities
- Even the best can often be improved
Platt Scaling by Fitting a Sigmoid

• Linear scaling of SVM $[-\infty, +\infty]$ predictions to $[0, 1]$ is bad

• Platt's Method [Platt 1999]:
  - Scale predictions by fitting sigmoid on a validation set using 3-fold CV and Bayes-motivated smooth
  - Linear scaling of SVM $[-\infty, +\infty]$ predictions to $[0, 1]$ is bad
Platt Scaling vs. Isotonic Regression

- Platt Scaling:
- Isotonic Regression:
Platt Scaling vs. Isotonic Regression
Auto-Calibrate

• Foster new research on new calibration methods
• Data mining challenge problem on calibration

• Current tools require expertise and careful use
• Easy to use tool for automatic calibration would see widespread use

• Use cross-validation for calibration samples when small data?
• Automatically select sample to be used for post-training calibration?

• Try multiple methods and reliably pick best...
• Probably depends on source model that generated scores in 1st place
• Depends on ROC
• Depends on data skew
• Depends on sample size

• Not as easy to make bulletproof as you might think

Auto-Calibrate
AutoML Open Problems

- **robust attribute typing and coding** — the spec is never right
- **Diff** — because the world never stops changing
- **runtime wrappers** — a model has to know its limitations
- **feedforward cycle detection** — and we never stop changing the world
- **auto calibration** — probabilities are good, but not easy to automate
- **auto leakage detection** — because data is never good enough
- **auto cross-validate** — because cross-validation isn't really as simple as you think
- **auto metric selection** — which metrics are sensitive to changes
- **auto calibration** — because data is never good enough
- **auto leakage detection** — because data is never good enough
- **skewed data expert** — because rare classes are very common
- **auto transfer** — sometimes transfer helps, sometimes transfer hurts
- **auto compression** — make small model as small and fast as possible

---

AutoML Open Problems
Leakage and other "accidents"

- 50% of data mining competitions have leakage!!!
- win data mining competitions
- KDD2011 best paper award:
  - "Leakage in Data Mining: Formulation, Detection, and Avoidance"

Shachar Kaufman, Saharon Rosset, Claudia Perlich, Ori Stitelman

- Automatic leakage detection:
  - sequential analysis, missing value analysis, feature analysis, diff train to real test, …
  - Automatic leakage detection:
  - important to have expectations and know when they are violated
  - pneumonia leakage 2: 4k features (AUC = 0.99)
  - pneumonia leakage 1: missing values

Leakage and other "accidents"
AutoML Open Problems

- Auto transfer -- sometimes transfer helps, sometimes transfer hurts
- Auto compression -- make small model as small and fast as possible
- Auto metric selection -- which metrics are sensitive to changes
- Auto cross-validate -- because cross-validation isn't really as simple as you think
- Skewed data expert -- because rare classes are very common
- Auto leakage detection -- because data is never good enough
- Auto calibration -- probabilities are good, but not easy to automate
- Feedback cycle detection -- and we never stop changing the world
- Runtime wrappers -- a model has to know its limitations
- Diff -- because the world never stops changing
- Robust attribute typing and coding -- the spec is never right

AutoML Open Problems
Want to do New Research that Gets Cited?
Want to do new research that gets cited?

- Pick a part of the ML pipeline that's still largely manual
- Define what it would mean to make it (more) automatic
- Develop and publish methods
- Define what it would mean to make it (more) automatic
- Fully-automatic "robot" that solves problem
- Assistant that helps human recognize and solve problem
- Tools that alert when problem (probably) exists
- Organize a challenge competition on that part of the pipeline
- Make code available for use as a baseline (and possibly open source)
- Make data sets publicly available
- Make data sets publicly available
Summary

• AutoML is a growth research area.
• Community has neglected 85% of the challenges of doing real ML.
• Many independent sub-problems all worthy of attention.
• Every time you stub your toe on a real problem = opportunity for new research.
• Every time you stub your toe on a real problem = opportunity for new research.
• Tools will immediately see widespread use.

• Make ddiff tools Open Source and available in R and Linux.
• ddiff challenge/competition in 1-2 years?
• ddiff workshop in 1-2 years?

Suggest we all do research and write papers on ddiff this year.

• Hyper-Parameter Optimization often critical --- start using it!

• Tools will immediately see widespread use.
• Make ddiff tools Open Source and available in R and Linux.
• ddiff challenge/competition in 1-2 years?
• ddiff workshop in 1-2 years?

Suggest we all do research and write papers on ddiff this year.

• Hyper-Parameter Optimization often critical --- start using it!
Thanks!