Building a Platform for Efficient Collaborative Research

Percy Liang

Microsoft Faculty Summit
July 16, 2013

Objective

*To create a **collaborative** ecosystem for conducting computational research in an **efficient** and **reproducible** manner.*

The current research process

Lack of reuse

Step 1: come up with a good idea

Step 2:

- Find data, clean it, convert between formats
- Find code, compile it, email authors, reimplement
- Run experiments, keep track of multiple versions

Non-exhaustive comparisons

	Previous method	Our method
Dataset 1	88% accuracy	92% accuracy
Dataset 2	72% accuracy	77% accuracy
Dataset 3	?	?
Dataset 4	?	?
Dataset 5	?	?
Dataset 6	?	?
...	?	?

Uncontrolled comparisons

Previous method	Our method
88% accuracy	92% accuracy
using sampling	using optimization
L_2 regularization	L_1 regularization
5-fold cross-validation	10-fold cross-validation
one set of bugs	another set of bugs

Lack of good broad overview

Question: Which algorithms work well on what types of datasets?

An outsider's perspective

Difficult to understand the **problems**:

classification
regression
ranking
structured prediction
statistical relational learning
...

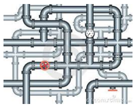
Difficult to find reliable **solutions**:

logistic regression
kernel methods
topic models
conditional random fields
hidden Markov models
...

Outline

MLcomp: code, data, comparison

CodaLab: complex workflows



MLcomp: code, data, and comparison

A meeting place

People with programs:

How well does my method work compared to others?

People with datasets:

What is the best method for my problem?

Components

Programs:

SVMlight
C implementation of support vector machines for classification by Thorsten Joachims.

Datasets:

thyroid
Task is to predict whether a patient has thyroid disease given attributes (age, gender, I131 treatment, etc.)

Runs:

Program	:	**SVMlight**
Dataset	:	**thyroid**
Error	:	2.6%
Time	:	1 second

Usage

- Users upload **programs**

- Users upload **datasets**

- System **runs** programs on datasets

Everything happens in a distributed/asynchronous manner.

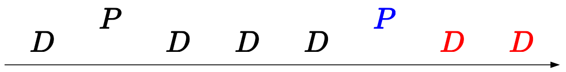
Result

Datasets

Programs

5.6	6.2	2.0	5.6	4.7	3.0	8.1	7.5	7.2	7.0	2.1	5.2	4.6
1.1	1.4	5.0	10.0	1.2	7.8	1.1	5.7	8.6	9.1	6.2	0.9	4.8
5.5	4.5	0.1	0.6	7.3	1.7	0.8	0.6	7.2	9.2	0.1	1.8	1.7
0.4	1.5	2.7	0.4	7.5	5.7	8.2	3.3	9.0	8.3	5.1	0.8	9.5
9.0	8.9	3.1	9.5	9.6	6.0	6.3	3.1	4.4	7.8	0.7	6.6	3.9
3.5	5.2	1.6	4.6	9.3	7.0	7.0	2.0	2.2	4.1	6.1	2.5	9.5
1.9	7.4	2.9	1.5	1.2	9.7	6.3	0.0	6.4	1.3	2.3	1.0	0.9
3.3	9.5	9.8	7.1	8.3	6.4	1.1	3.4	8.9	2.5	9.5	2.2	3.9

Generalization


D P D D D P D D

Evaluation:

- Program P run on dataset D, get some accuracy
- Only meaningful if D is **independent** of P
- In papers, this is never true!

In MLcomp:

- People upload program P
- People upload new dataset D **afterwards**
- Guarantees that P is not overfit to D

Design decisions

- Program is an arbitrary Linux binary (support C++, Java, Python, R, etc.)

- User-uploaded programs and datasets conform to standard interfaces/formats

- All runs executed on Amazon EC2 (initiated by user or system)

- Users can download any programs/datasets/runs (not marked by user as restricted)

Related projects

Code repositories (mloss.org): only code

Data repositories (UCI, mldata.org): only data

Machine learning as a service (BigML, Google Prediction API): provide fixed set of programs, people submit (private) data; doesn't encourage development of new methods

Competitions (Kaggle): provide fixed set of datasets, people submit predictions; doesn't promote general/clean solutions

Status

- Development started in 2009 [with Jake Abernethy, Alex Simma, Ariel Kleiner]

- Today: 2129 users, 686 datasets, 390 programs, 19083 runs

- Website: mlcomp.org

- Open-source on GitHub: https://github.com/percyliang/mlcomp

CodaLab: collaborative workflows

Delving deeper

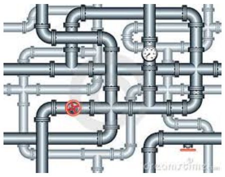
MLcomp's primitive:

accuracy = run(program, dataset)

Want fuller analysis:

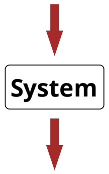
- Hyperparameter tuning / sensitivity analysis
- Learning curves (varying amounts of data)
- Error analysis: ROC, confusion matrices, predictions
- Visualization: plot all these statistics

AI problems require complex workflows



Text understanding

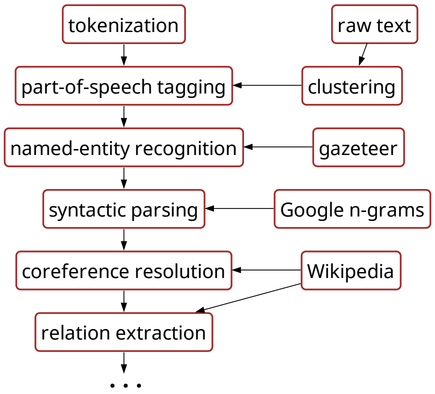
Peter Houser, a 68-year-old man with a history of hypertension, has an aortic aneurysm repaired. On the first postoperative day, he is transferred from the intensive care unit (ICU) to the medical-surgical unit. Mr. Houser has a midline incision, a nasogastric tube connected to low intermittent suction, and a left subclavian triple lumen catheter. On receiving Mr. Houser from the ICU, the nurse notes that he has edema of both lower extremities, and his pedal pulses are not palpable in either foot. Which of these actions should the nurse take?



System

Use a hand-held Doppler ultrasound device to reassess his pulses.

Stages of an NLP workflow



tokenization

raw text

part-of-speech tagging

clustering

named-entity recognition

gazeteer

syntactic parsing

Google n-grams

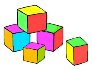
coreference resolution

Wikipedia

relation extraction

. . .

Three principles



Modularity

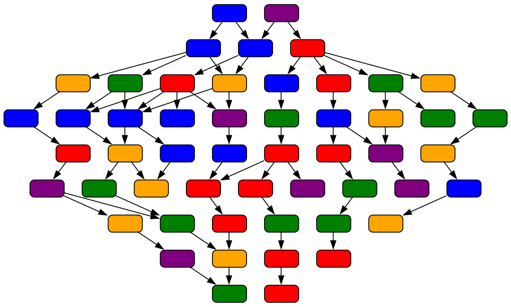
Immutability

Literacy

Principle 1: modularity

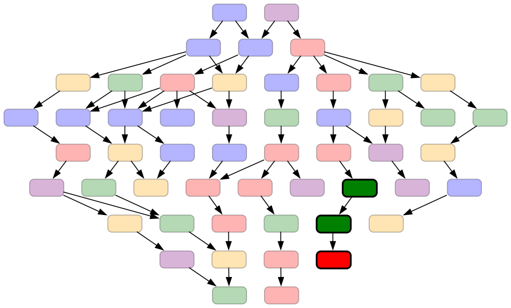
AI problems require efforts of entire community

People specialize, contribute in decentralized way

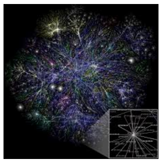


Intermediate tasks

- Old: use intermediate metrics, rhetoric
- New: plug in and see ramifications **automatically**



A collaborative ecosystem



Modules interoperate via standard interfaces (Internet)

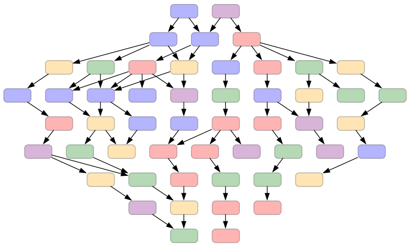
Individual benefits:

- Avoid duplicate work
- Publicize tools/datasets

Commununity benefits:

- Serial combination of components
- Parallel ensembles for better predictions (Netflix Prize)

Principle 2: immutability



Inspiration: Git version control system

- All programs/datasets/runs are write-once
- Enable collaboration without chaos
- Capture the research process in a **reproducible** way

Principle 3: literacy

MLcomp is about **truth**; what about **interpretation**?

Inspiration:

- Mathematica notebook, IPython notebook: interleave code with text descriptions

CodaLab worksheet

We now train the classifier with more data.

Program	:	**SVMlight**
Arguments	:	-n 2000
Dataset	:	**thyroid**
Error	:	2.6%
Time	:	1 second

Notice that the error remains the same, suggesting that we've saturated our model.

Use cases:

- Informal blog posts
- Formal executable papers

Related projects

- runmycode.org, myexperiment.org, Weka require specific formats
- Matlab/R/Perl/Python/Ruby provide code modules, but no data; data is a resource

Challenges

Inertia:

- Problem: People have personal setup, takes effort to port to foreign environment
- Solution: Easy to contribute, benefits of online sharing (Dropbox + execution)

Search:

- Problem: CodaLab is general repository, how to search?
- Solution: Smart autocomplete, ranking, recommendation

Status

Collaboration with Microsoft Research Connections

Project hosted by the Outercurve Foundation

People:

- Development: Christophe Poulain, Beau Hargis, Justin Carden, Dan O'Donnell
- Program/community: Evelyne Viegas, Erick Watson, Lori Ada Kilty, Ivan Judson, Simon Mercer
- Design: Chrisopher Rampey

Final remarks

*To create a **collaborative** ecosystem for conducting computational research in an **efficient** and **reproducible** manner.*

Questions/feedback?