Machine Learning for Humans

Ryan Marcus
Brandeis University
Los Alamos National Lab, HPC-5

@RyanMarcus RyanMarcus RyanCMarcus
http://ryanmarc.us
rmarcus@lanl.gov

July 28, 2015

Machine Learning

Optimize

$$\min_{\vec{d}} \left(\left(f(\vec{p}, \vec{d}) -_{\delta} \vec{L} \right)^2 - \alpha \sum (\vec{d}\vec{\omega}) \right)$$

Subject to

$$P(\vec{d}|\vec{D}) \leq \beta \int_x P(x|\vec{D})$$

- $\vec{d}$, model parameters
- f, model function
- $\vec{p}$, normalized feature vector
- $-_{\delta}$, expectation loss function
- $\vec{L}$, normalized sample labels
- α, complexity penalty coefficient
- $\vec{\omega}$, parameter importance vector
- $\vec{D}$, domain parameter space
- β, parameter penalty
- x, feature space

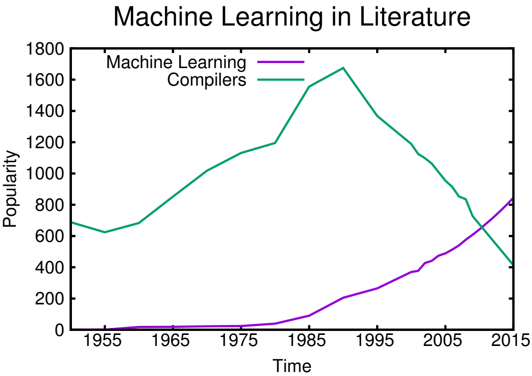
What is ML?

How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?

- Professor Tom Mitchell, Carnegie Melon University

What is ML?

- 1950: Turing Test
- 1952: Checkers
- 1990: Statistical Machine Learning
- 1997: Deep Blue
- 2014: Watson



Examples

- Spam detection
- Credit card fraud detection
- Digit recognition
- Adversary analysis
- Medical diagnostic
- Stock prediction
- Customer segmentation

Models of Learning

What do we mean by “learning”?

Intuition Extraction System

”You don’t know it until you can teach it.”

- Find basic patterns
- Extract general rules
- Simplicity valued over correctness

Models of Learning

What do we mean by "learning"?

Intuition Extraction System

"You don't know it until you can teach it."

- Find basic patterns
- Extract general rules
- Simplicity valued over correctness

Expert Prediction System

"You don't know it until you can predict it."

- Find exact patterns
- Extract precise relationships
- Correctness valued over simplicity

Models of Learning

Most machine learning tasks are **labeling** tasks:

- **features** $\rightarrow$ **label**
- car weight, engine size $\rightarrow$ MPG
- purchase location, amount $\rightarrow$ fraudulent?
- image of face $\rightarrow$ name
- historic stock values $\rightarrow$ future value

Models of Learning

Most machine learning tasks are **labeling** tasks:

- **features** $\rightarrow$ **label**
- car weight, engine size $\rightarrow$ MPG
- purchase location, amount $\rightarrow$ fraudulent?
- image of face $\rightarrow$ name
- historic stock values $\rightarrow$ future value

Numeric labels? **Regression**
Categorical labels? **Classification**

Models of Learning

Two main types of learning algorithms...

Supervised Learning

- Learn through labeled examples
- Starting knowledge: **training set**
- Example: database of cars and mileage
- Algorithm will map cars to mileage

Models of Learning

Two main types of learning algorithms...

Supervised Learning

- Learn through labeled examples
- Starting knowledge: **training set**
- Example: database of cars and mileage
- Algorithm will map cars to mileage

Unsupervised Learning

- Learn likely labels from “raw” data
- Starting knowledge: **a priori information**
- Example: a list of senators and how they vote
- Algorithm will cluster senators

Models of Learning

	Supervised	**Unsupervised**
"Expert" (black box)	Ensemble SVM	Clustering Gaussian Mixture
"Intuition" (white box)	Linear fit Trees	Signal separation

Today

Linear fit, trees, ensemble

Software packages

- No “one package to rule them all”
- Weka: point-and-click ML sandbox
- `scikit-learn`: a Python ML package

Task: Predicting MPG

Features

- Cylinders $= c$
- Displacement $= d$
- Horsepower $= h$
- Weight $= w$
- Acceleration $= a$
- Year $= y$

Label

- Mileage $= m$

Goal

Given features, predict the label.
Given information about a car, predict its mileage.

Dataset

398 labeled cars

Supervised or unsupervised?

Task: Predicting MPG

Features

- Cylinders $= c$
- Displacement $= d$
- Horsepower $= h$
- Weight $= w$
- Acceleration $= a$
- Year $= y$

Label

- Mileage $= m$

Goal

Given features, predict the label.
Given information about a car, predict its mileage.

Dataset

398 labeled cars

Supervised or unsupervised?

Supervised

Linear regression

The oft-forgotten ancestor of machine learning

Linear regression

The oft-forgotten ancestor of machine learning

$$m = \alpha_c * c + \alpha_d * d + \alpha_h * h + \alpha_w * w + \alpha_a * a + \alpha_y * y + \beta$$

Linear regression

The oft-forgotten ancestor of machine learning

$$m = \alpha_c * c + \alpha_d * d + \alpha_h * h + \alpha_w * w + \alpha_a * a + \alpha_y * y + \beta$$

$$m = \vec{\alpha} \cdot \vec{f} + \beta$$

Linear regression

Demo!

Linear regression

Feature	Coefficient
Year	0.758260124408
Cylinders	-0.253094820718
Acceleration	0.0948814284056
Weight	-0.00699036036498
Displacement	0.00690031871391
Horsepower	0.00302869962434

Questions

1. Simple random sample?
2. Features independent?

Coefficient analysis can tell you quite a bit about the general patterns of your data.

Linear regression

Model Evaluation

But how good is our model for prediction? It should be good at predicting the training set, but how will it **generalize**?

- Don't just use R^2
- Don't just use root mean squared error

Linear regression

Model Evaluation

But how good is our model for prediction? It should be good at predicting the training set, but how will it **generalize**?

- Don't just use R^2
- Don't just use root mean squared error
- ... don't use anything from running the model on the entire training set

Instead, use cross validation!

Cross validation

Suppose you had 9 training samples, a through i.

Fold 1

Test			Train					
a	b	c	d	e	f	g	h	i

Fold 2

Train			Test			Train		
a	b	c	d	e	f	g	h	i

Fold 3

Train						Test		
a	b	c	d	e	f	g	h	i

Cross validation

Suppose you had 9 training samples, a through i.

Fold 1

Test			Train					
a	b	c	d	e	f	g	h	i

Fold 2

Train			Test			Train		
a	b	c	d	e	f	g	h	i

Fold 3

Train						Test		
a	b	c	d	e	f	g	h	i

Demo!

Cross validation

Suppose you had 9 training samples, a through i.

Fold 1

Test			Train					
a	b	c	d	e	f	g	h	i

Fold 2

Train			Test			Train		
a	b	c	d	e	f	g	h	i

Fold 3

Train						Test		
a	b	c	d	e	f	g	h	i

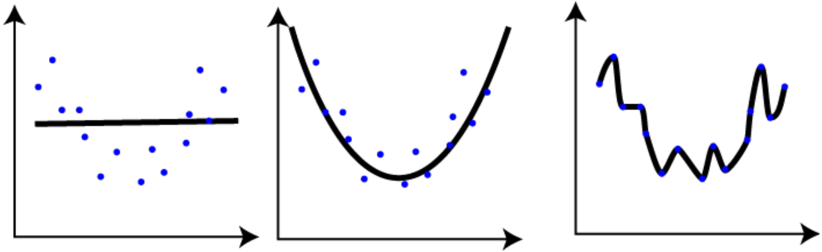
Demo! Our absolute mean error: -3.37825068688.

Cross validation

- You can always get 0 error in the training set
- ... but you generally don't want to

Cross validation

- You can always get 0 error in the training set
- ... but you generally don't want to



Source: `shapeofdata.wordpress.com`

Cross validation

Helps...

- Prevent overfitting
- Compare different models

Assumes...

- No skew in sampling
- No "twinning" (uniqueness)

Restaurant Performance

- Do customers stay or go?

Restaurant Performance

Dataset: Cold Rocks Java Cafe

Meal	Weather	Wait	Temperature	Action
Lunch	Sunny	None	48F	Stay
Lunch	Rain	Short	40F	Left
Lunch	Cloudy	Long	78F	Left
Lunch	Sunny	None	88F	Stay
Dinner	Rain	None	65F	Stay
Dinner	Sunny	Long	97F	Stay
Dinner	Rain	Long	92F	Left
Dinner	Rain	Short	88F	Stay

Restaurant Performance

Dataset: Cold Rocks Java Cafe

Features				**Labels**
Categorical			*Scalar*	
Meal	Weather	Wait	Temperature	Action
Lunch	Sunny	None	48F	Stay
Lunch	Rain	Short	40F	Left
Lunch	Cloudy	Long	78F	Left
Lunch	Sunny	None	88F	Stay
Dinner	Rain	None	65F	Stay
Dinner	Sunny	Long	97F	Stay
Dinner	Rain	Long	92F	Left
Dinner	Rain	Short	88F	Stay

Restaurant Performance

Goal

To understand what variables affect customer retention.

Expert or Intuition?

Restaurant Performance

Goal
To understand what variables affect customer retention.

Expert or Intuition?
Intuition

Tools
We'll use Weka. Demo!

Restaurant Performance

- Decision trees identify important factors
- ... can help explain decision processes
- ... are (somewhat) robust to noise

Parameters

Like most ML methods, decision tree learners generally have a lot of parameters...

- Minimum leaf size (increase to generalize)
- Maximum depth (decrease to generalize)

Restaurant Performance

Decision Tree Issues

- Without CV, can create very complex trees
- "Prior sensitivity" (class distribution)
- Verifying that a tree is optimal is NP-complete[1]

[1] Computer science speak for "takes way too long".

Hardware Failure Prediction

- The Oak Ridge/U of Tennessee Kraken Supercomputer
- 2008 - 2014
- 112,896 cores for 1.17 petaflops (2% of Trinity)
- 147 TB of memory (6% of Trinity)

JMTTI

- Big machines have failures.
- Average time between failures: JMTTI (6 - 12 hours)
- We want to increase that.

Hardware Failure Prediction

- The Oak Ridge/U of Tennessee Kraken Supercomputer
- 2008 - 2014
- 112,896 cores for 1.17 petaflops (2% of Trinity)
- 147 TB of memory (6% of Trinity)

JMTTI

- Big machines have failures.
- Average time between failures: JMTTI (6 - 12 hours)
- We want to increase that.
- Can we predict which nodes will fail?

Hardware Failure Prediction

Dataset

- 1000 training samples (small!)
- 200 obfuscated features (all numeric) represent pre-event measurments
- Labels, `fail` or `nofail`
- 568 non-failures, 432 failures

What to do?

Hardware Failure Prediction

Dataset

- 1000 training samples (small!)
- 200 obfuscated features (all numeric) represent pre-event measurments
- Labels, `fail` or `nofail`
- 568 non-failures, 432 failures

What to do?

- **The Data Scientist** carefully considers the pros and cons of different models, applies and validates a few of them based on experience and the model's mathematical properties.

Hardware Failure Prediction

Dataset

- 1000 training samples (small!)
- 200 obfuscated features (all numeric) represent pre-event measurments
- Labels, `fail` or `nofail`
- 568 non-failures, 432 failures

What to do?

- **The Data Scientist** carefully considers the pros and cons of different models, applies and validates a few of them based on experience and the model's mathematical properties.
- ... we're going to throw models at the wall and see what sticks.

Ensemble Methods

Ensemble methods **combine simple methods** into expert systems. They are...

- Slow to train
- Opaque
- Often what gives the best result

Ensemble Methods

Ensemble methods **combine simple methods** into expert systems. They are...

- Slow to train
- Opaque
- Often what gives the best result

In `scikit-learn`, using ensemble methods is easy. Just change:

```
linear_reg = sklearn.linear_model.LinearRegression()
```

to (for example)

```
linear_reg = sklearn.ensemble.GradientBoostingRegressor()
```

Ensemble Methods

Ensemble methods to try...

- Bagging: combine any learner. Great way to improve an already-OK model
- Random forest: fast(er) approach, a classic
- Gradient boosting: more “state of the art”, very slow, poor scaling

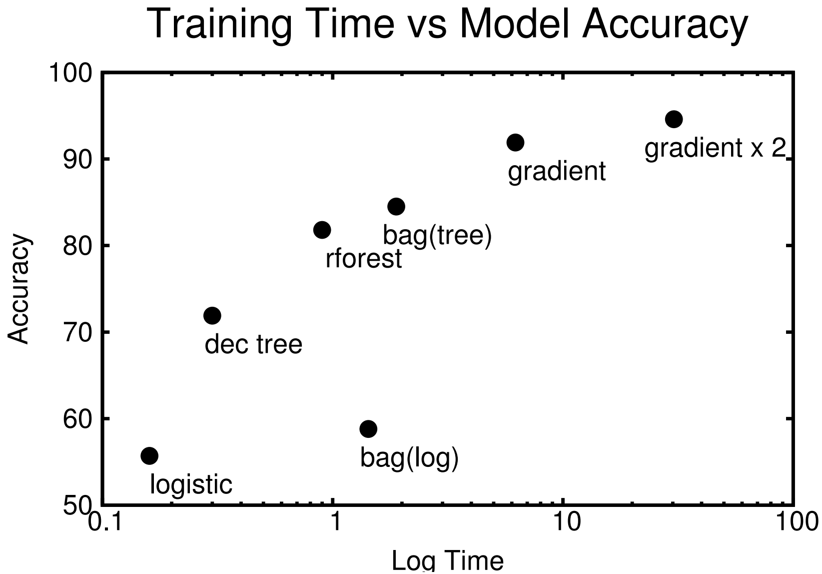
Hardware Failure Prediction

Model	**Description**
Logistic	Fit a logistic to the data
Tree	Use a decision tree
bag(Log)	Use **b**ootstrap **ag**gregating (bagging) on logs
bag(Tree)	Use bagging on trees
Random forest	Use a random forest of trees
Gradient Boosting	Use gradient boosting to fit trees

Hardware Failure Prediction

Model	**Description**
Logistic	Fit a logistic to the data
Tree	Use a decision tree
bag(Log)	Use **b**ootstrap **ag**gregating (bagging) on logs
bag(Tree)	Use bagging on trees
Random forest	Use a random forest of trees
Gradient Boosting	Use gradient boosting to fit trees

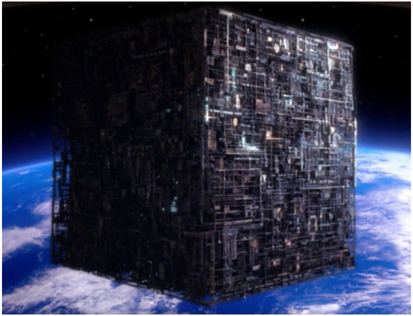
Demo!



Note the **log scale** on the x-axis

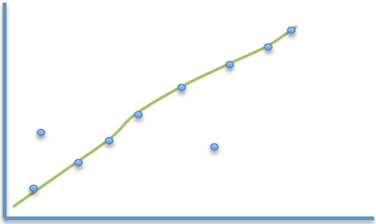
General Advice

- We are data science. Your discipline's methodological and technological distinctiveness will be added to our own. Resistance is futile.

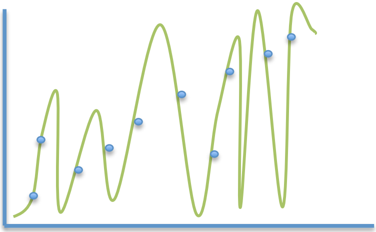


General Advice

- Most common mistake: **overfitting**



Good fit, captures pattern



Lower error, bad fit!

General Advice

When possible, **avoid complexity**.

- Start with linear regression
- Then try a decision tree
- Then kNN, SVM
- Random forest
- Bagging
- Gradient boosting

Resources

- GitHub Repo: https://github.com/RyanMarcus/MLForHumans
- scikit-learn user guide: http://scikit-learn.org/stable/user_guide.html
- Weka: http://www.cs.waikato.ac.nz/ml/weka/
- "The Book": www.amazon.com/Elements-Statistical-Learning-Prediction-Statistics/dp/0387848576/
- A more general, higher level, extremely well-regarded AI book: http://www.amazon.com/Artificial-Intelligence-Modern-Approach-Edition/dp/0136042597/

Ryan Marcus, Brandeis University, LANL HPC-5

- @RyanMarcus
- RyanMarcus
- RyanCMarcus
- `http://ryanmarc.us`
- rmarcus@lanl.gov