



Data-Intensive Distributed Computing

CS 451/651 431/631 (Winter 2018)

Part 6: Data Mining (2/4)
March 1, 2018

Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo

These slides are available at <http://lintool.github.io/bigdata-2018w/>



This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States
See <http://creativecommons.org/licenses/by-nc-sa/3.0/us/> for details

The Task

Given: $D = \{(x_i, y_i)\}_i^n$

A diagram illustrating the given data. A red bracket labeled '(sparse) feature vector' encloses the term x_i . A red arrow points from the label y_i to the term y_i in the set definition.

$$x_i = [x_1, x_2, x_3, \dots, x_d]$$

$$y \in \{0, 1\}$$

Induce: $f : X \rightarrow Y$

Such that loss is minimized

$$\frac{1}{n} \sum_{i=0}^n \ell(f(x_i), y_i)$$

A diagram illustrating the loss function. A red bracket labeled 'loss function' encloses the term $\ell(f(x_i), y_i)$.

Typically, we consider functions of a parametric form:

$$\arg \min_{\theta} \frac{1}{n} \sum_{i=0}^n \ell(f(x_i; \theta), y_i)$$

A diagram illustrating the model parameters. A red bracket labeled 'model parameters' encloses the term θ .

The background image shows a wide, open landscape with rolling green hills. The sky above is a vibrant blue, filled with large, white, fluffy clouds. The foreground is a grassy field, and there are some trees on the right side.

Gradient Descent

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \frac{1}{n} \sum_{i=0}^n \nabla \ell(f(\mathbf{x}_i; \theta^{(t)}), y_i)$$

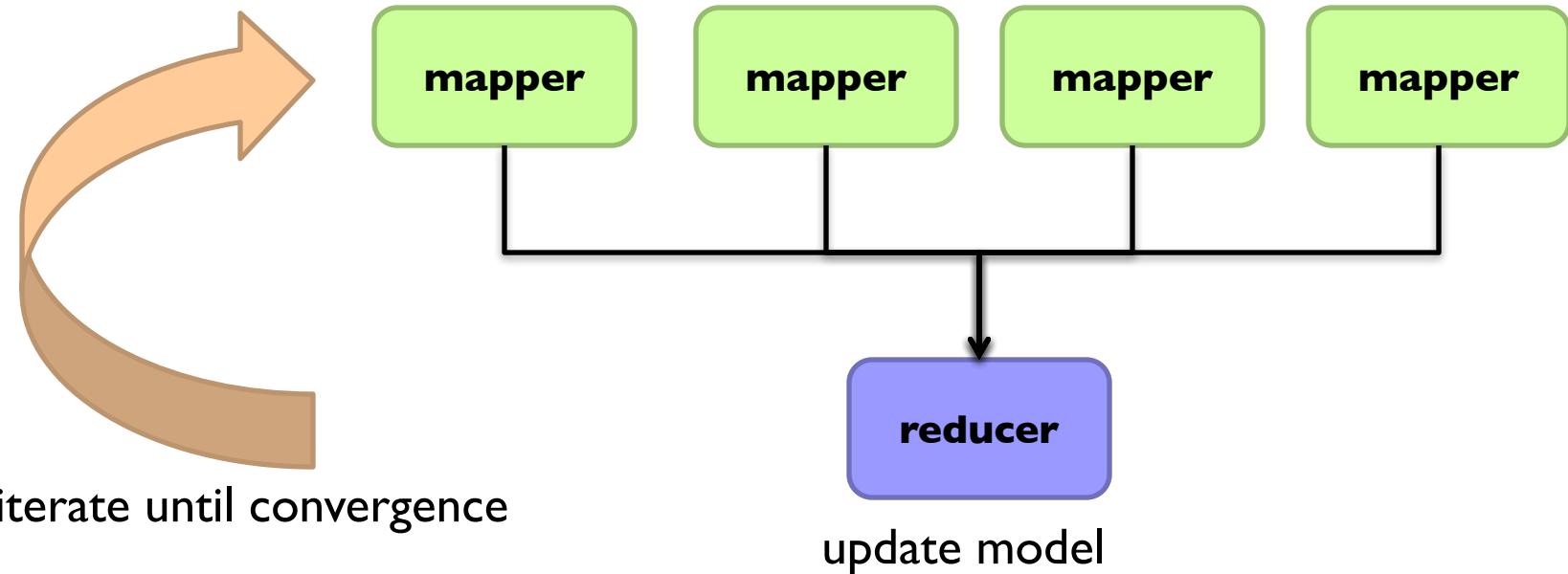
MapReduce Implementation

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \frac{1}{n} \sum_{i=0}^n \nabla \ell(f(\mathbf{x}_i; \theta^{(t)}), y_i)$$

mappers

single reducer

compute partial gradient

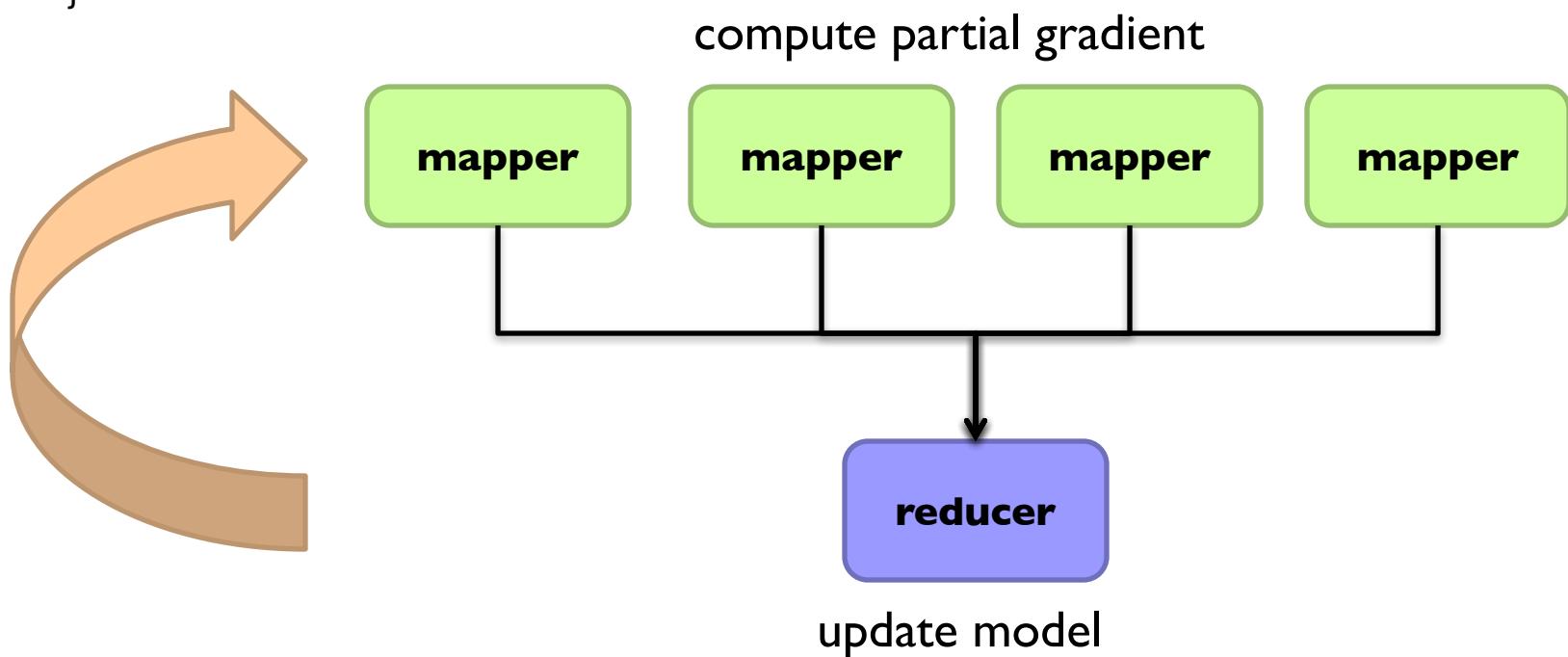


Spark Implementation

```
val points = spark.textFile(...).map(parsePoint).persist()
```

```
var w = // random initial vector
for (i <- 1 to ITERATIONS) {
    val gradient = points.map{ p =>
        p.x * (1/(1+exp(-p.y*(w dot p.x)))-1)*p.y
    }.reduce((a,b) => a+b)
    w -= gradient
}
```

What's the difference?



Gradient Descent

Stochastic Gradient Descent



Batch vs. Online

Gradient Descent

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \frac{1}{n} \sum_{i=0}^n \nabla \ell(f(\mathbf{x}_i; \theta^{(t)}), y_i)$$

“batch” learning: update model after considering all training instances

Stochastic Gradient Descent (SGD)

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \nabla \ell(f(\mathbf{x}; \theta^{(t)}), y)$$

“online” learning: update model after considering each (randomly-selected) training instance

In practice... just as good!

Opportunity to interleaving prediction and learning!

Practical Notes

Order of the instances important!

Most common implementation: randomly shuffle training instances

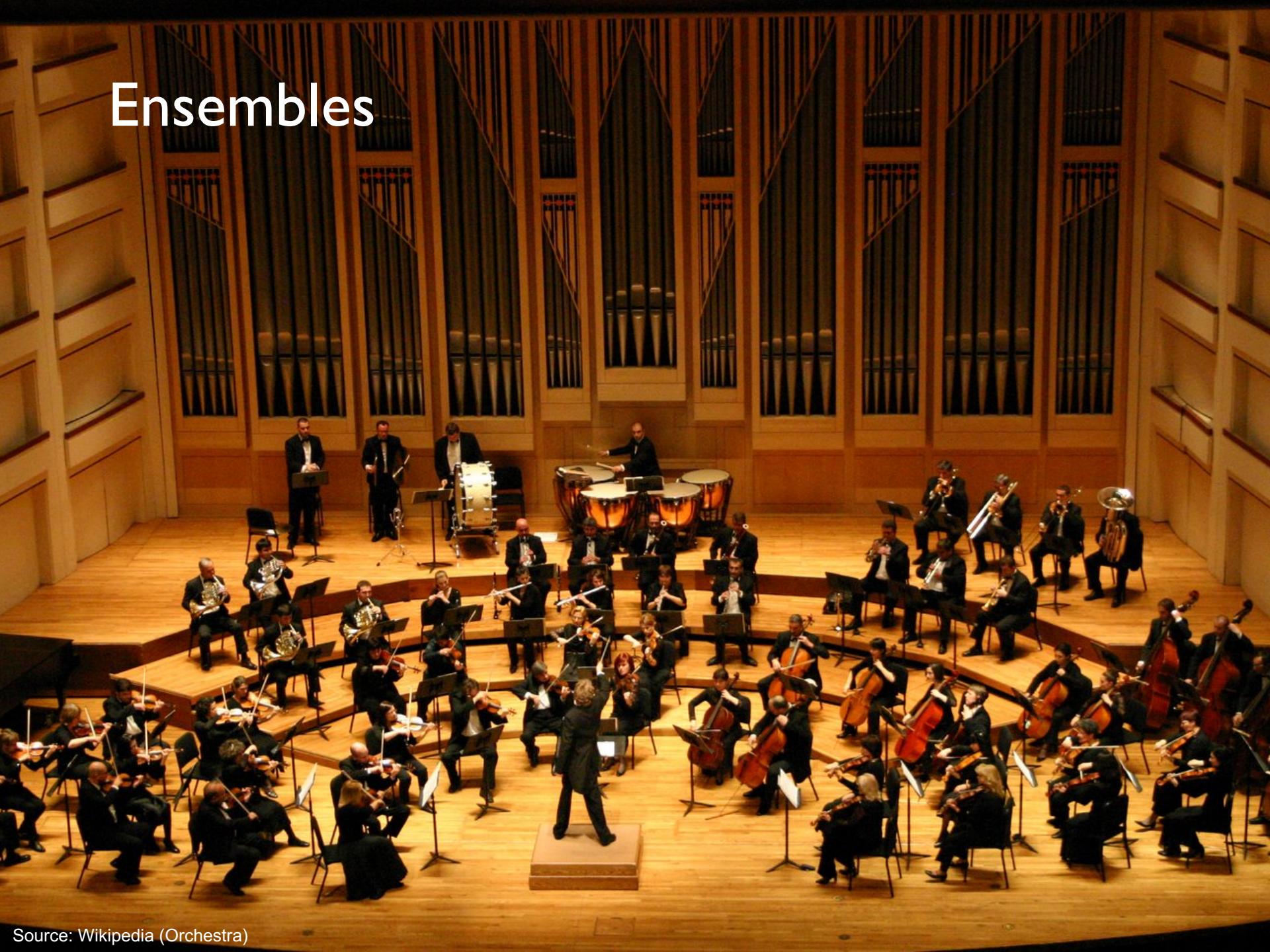
Single vs. multi-pass approaches

Mini-batching as a middle ground

We've solved the iteration problem!

What about the single reducer problem?

Ensembles



Ensemble Learning

independent
Learn multiple models, combine results from
different models to make prediction

Common implementation:

Train classifiers on different input partitions of the data
Embarrassingly parallel!

Combining predictions:

Majority voting

Simple weighted voting:

$$y = \arg \max_{y \in Y} \sum_{k=1}^n \alpha_k p_k(y|x)$$

Model averaging

...

Ensemble Learning

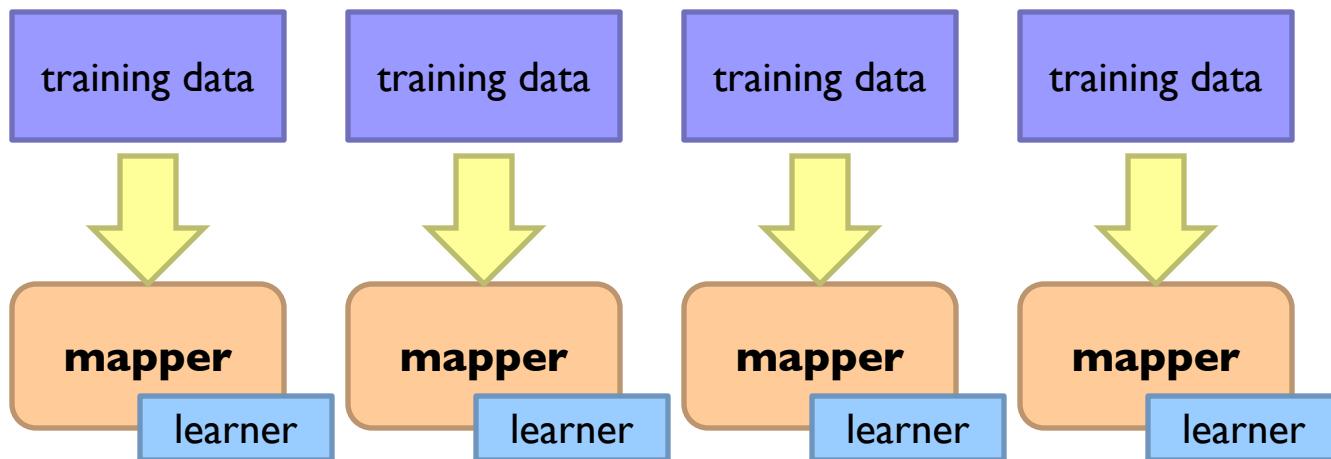
Learn multiple models, combine results from
different ^{independent} models to make prediction

Why does it work?

If errors uncorrelated, multiple classifiers being wrong is less likely
Reduces the variance component of error

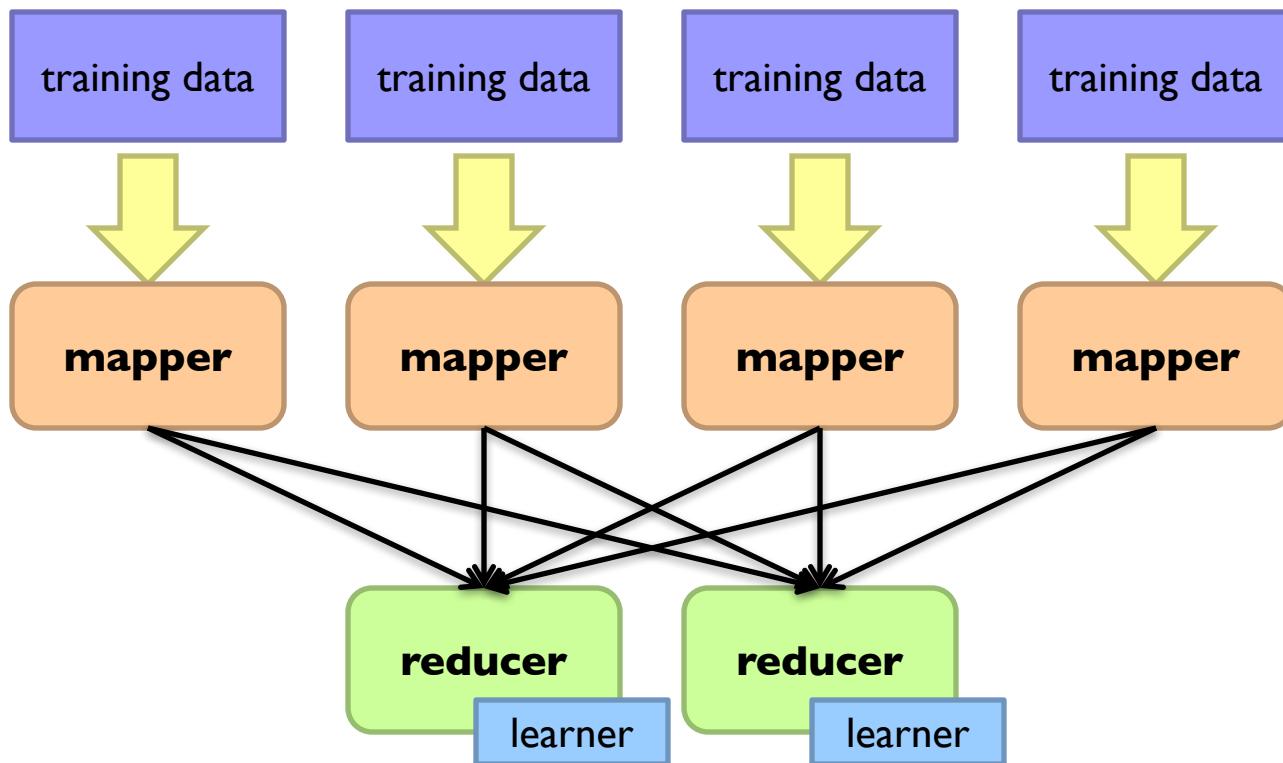
MapReduce Implementation

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \nabla \ell(f(\mathbf{x}; \theta^{(t)}), y)$$



MapReduce Implementation

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \nabla \ell(f(\mathbf{x}; \theta^{(t)}), y)$$



MapReduce Implementation

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \nabla \ell(f(\mathbf{x}; \theta^{(t)}), y)$$

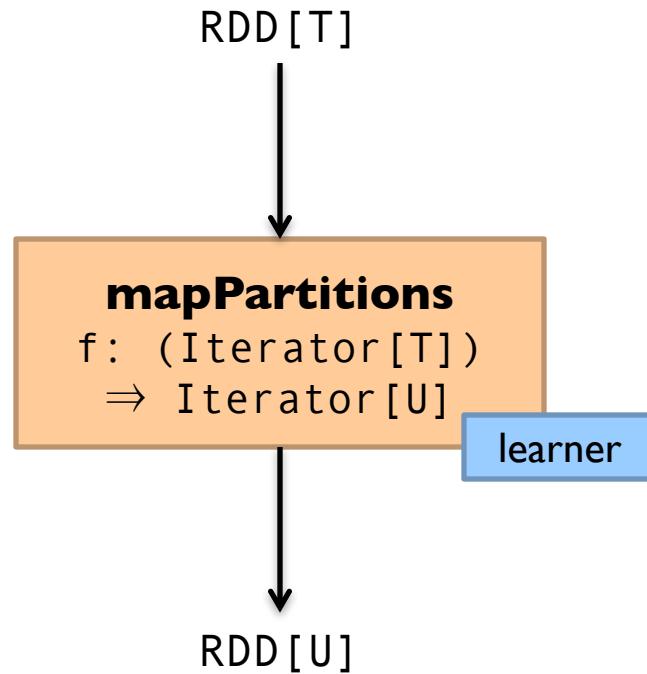
How do we output the model?

Option 1: write model out as “side data”

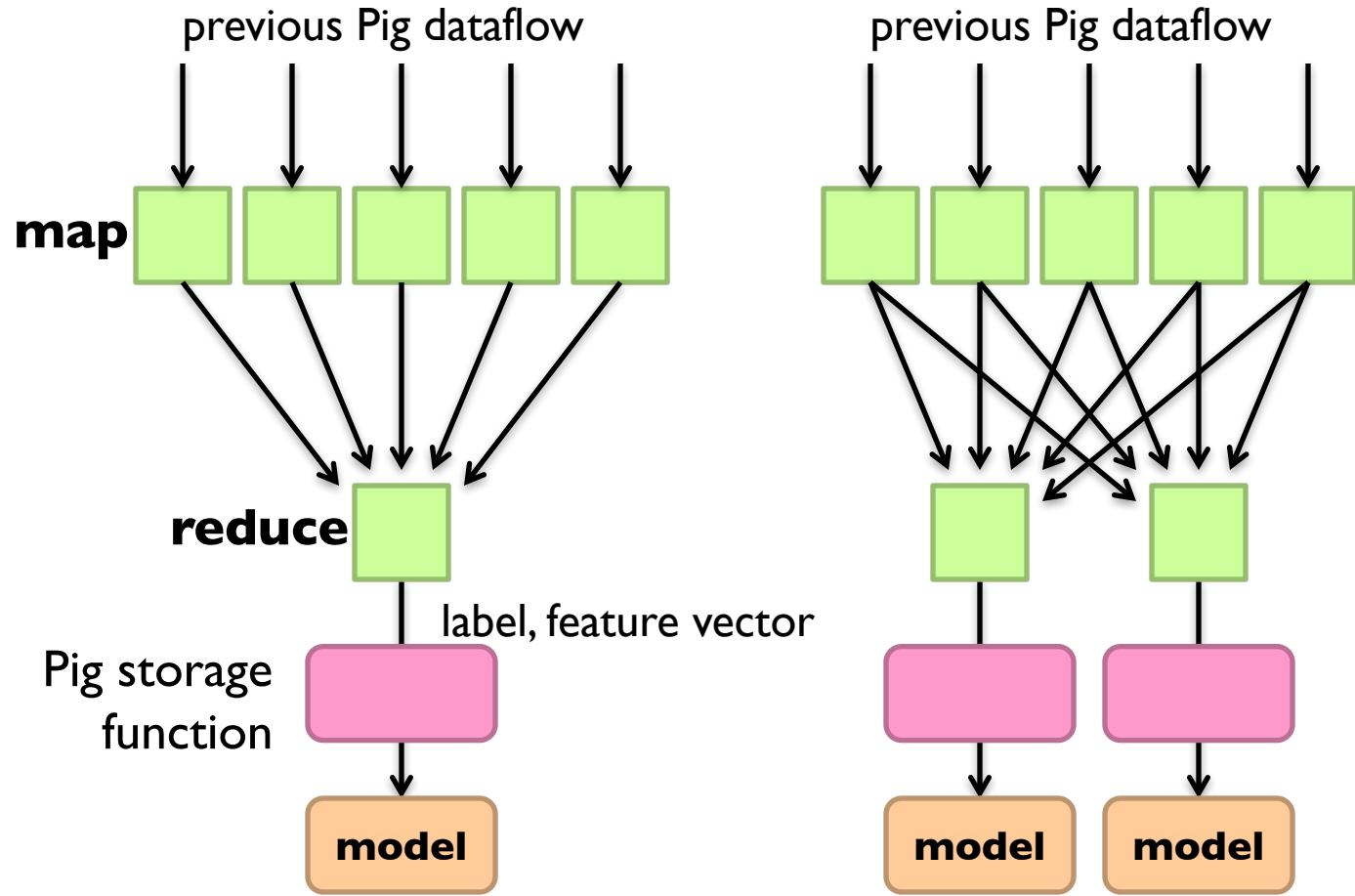
Option 2: emit model as intermediate output

What about Spark?

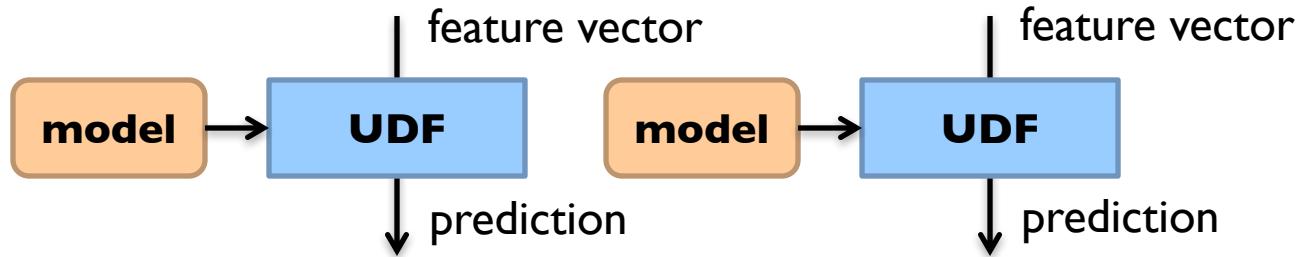
$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \nabla \ell(f(\mathbf{x}; \theta^{(t)}), y)$$



Classifier Training



Making Predictions



Just like any other parallel Pig dataflow

Classifier Training

```
training = load 'training.txt' using SVMLightStorage()  
    as (target: int, features: map[]);
```

```
store training into 'model/'  
    using FeaturesLRClassifierBuilder();
```

↑
Logistic regression + SGD (L2 regularization)
Pegasos variant (fully SGD or sub-gradient)

Want an ensemble?

```
training = foreach training generate  
    label, features, RANDOM() as random;  
training = order training by random parallel 5;
```

Making Predictions

```
define Classify ClassifyWithLR('model/');
data = load 'test.txt' using SVMLightStorage()
    as (target: double, features: map[]);
data = foreach data generate target,
    Classify(features) as prediction;
```

Want an ensemble?

```
define Classify ClassifyWithEnsemble('model',
    'classifier.LR', 'vote');
```

Sentiment Analysis Case Study

Binary polarity classification: {positive, negative} sentiment
Use the “emoticon trick” to gather data

Data

Test: 500k positive/500k negative tweets from 9/1/2011
Training: {1m, 10m, 100m} instances from before (50/50 split)

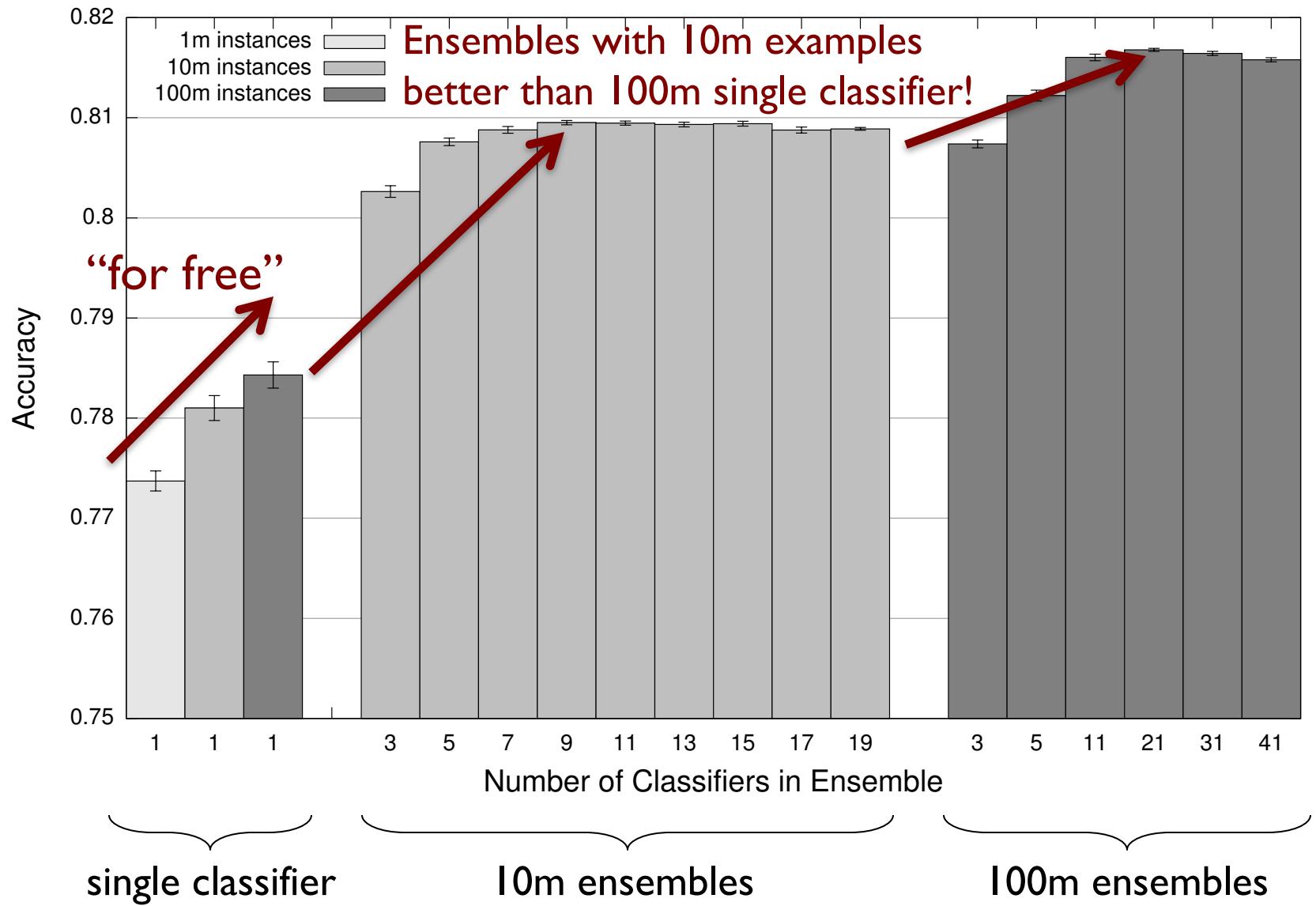
Features:

Sliding window byte-4grams

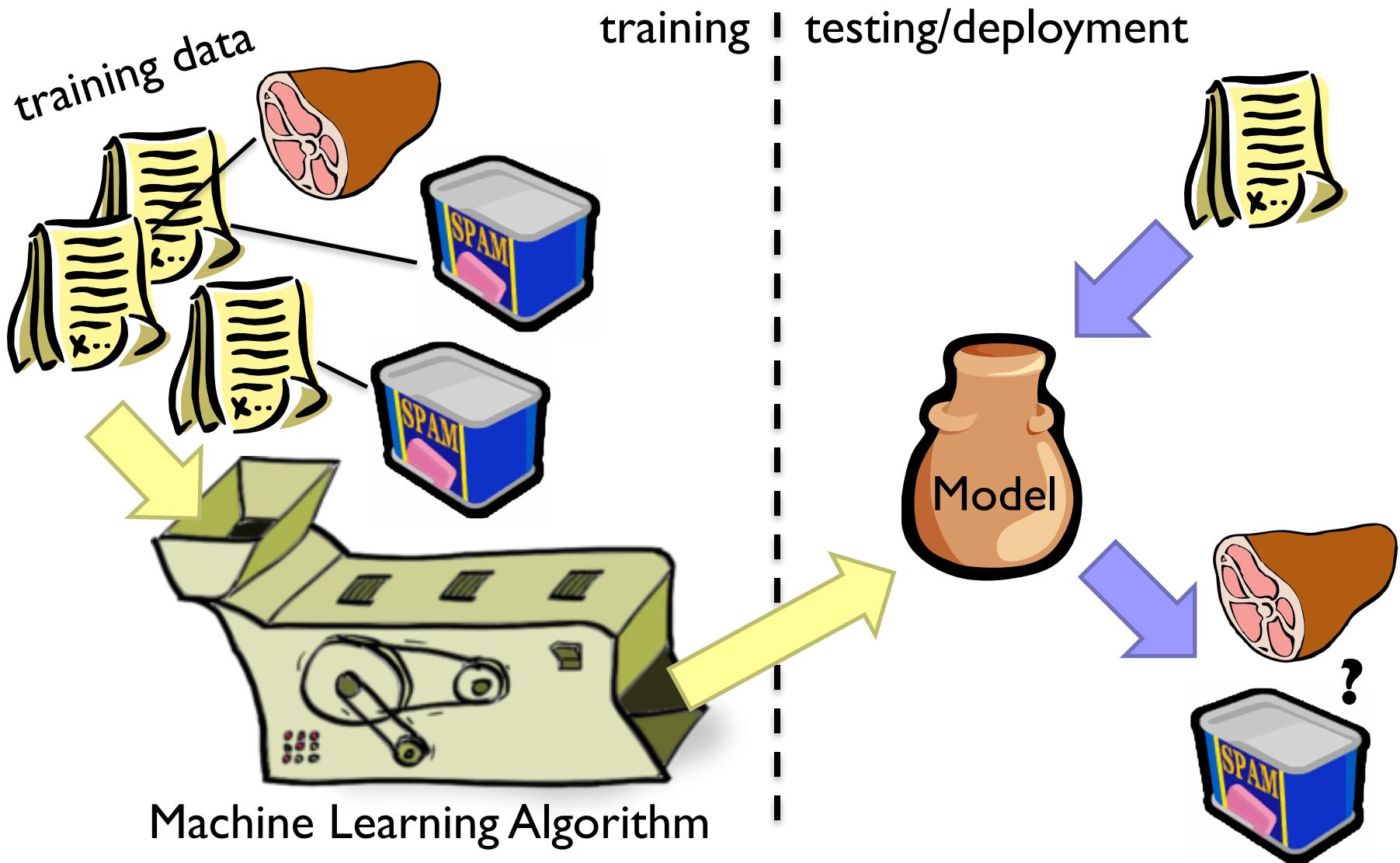
Models + Optimization:

Logistic regression with SGD (L2 regularization)
Ensembles of various sizes (simple weighted voting)

Diminishing returns...



Supervised Machine Learning



Evaluation

How do we know how well we're doing?

Why isn't this enough?

Induce: $f : X \rightarrow Y$

Such that loss is minimized

$$\arg \min_{\theta} \frac{1}{n} \sum_{i=0}^n \ell(f(x_i; \theta), y_i)$$

We need end-to-end metrics!

Obvious metric: accuracy

Why isn't this enough?

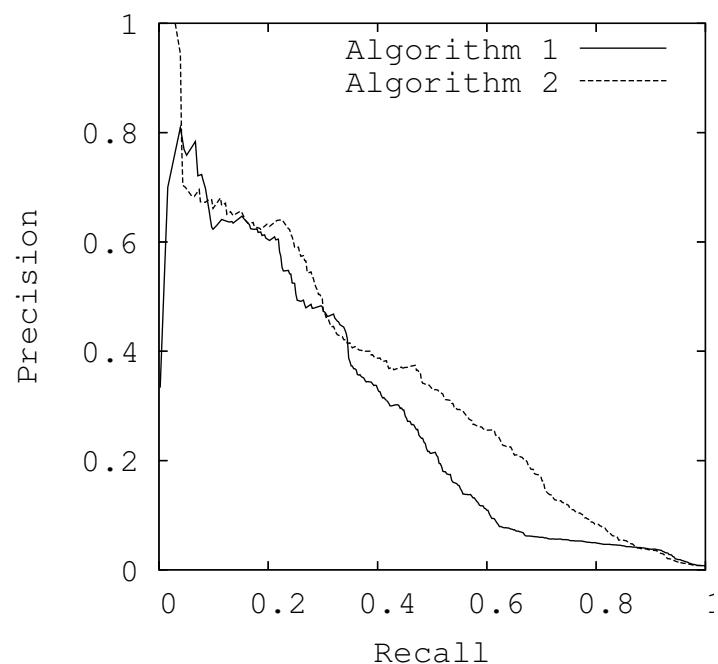
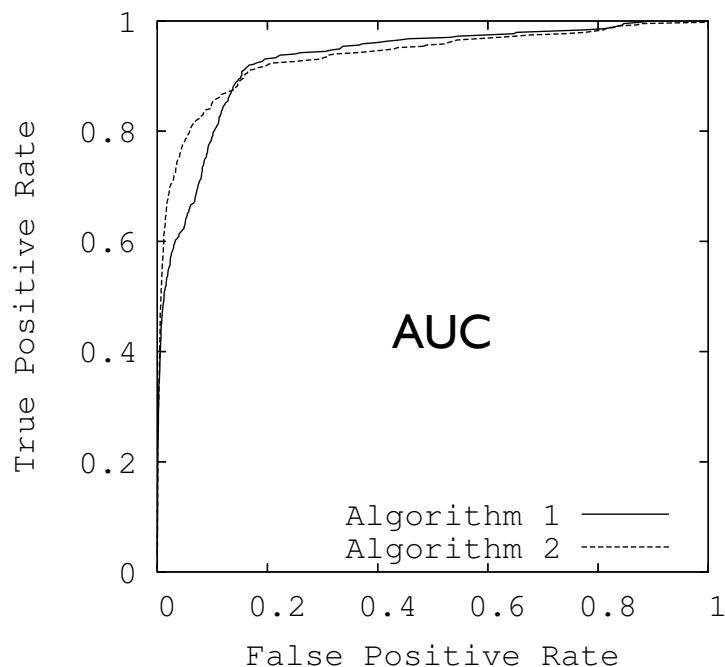
Metrics

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP) = Type I Error
	Negative	False Negative (FN) = Type II Error	True Negative (TN)
		Recall or TPR = $TP/(TP + FN)$	Fall-Out or FPR = $FP/(FP + TN)$

Precision = $TP/(TP + FP)$

Miss rate = $FN/(FN + TN)$

ROC and PR Curves



Training/Testing Splits

Training

$$\arg \min_{\theta} \frac{1}{n} \sum_{i=0}^n \ell(f(x_i; \theta), y_i)$$

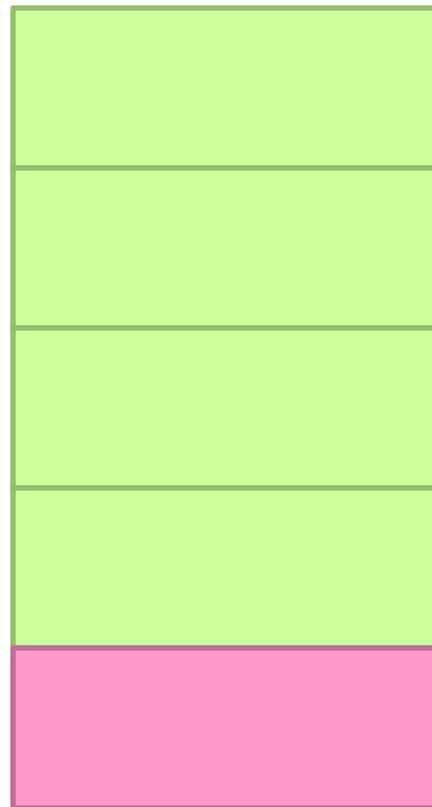
Test

Precision, Recall,
etc.

What happens if you need more?

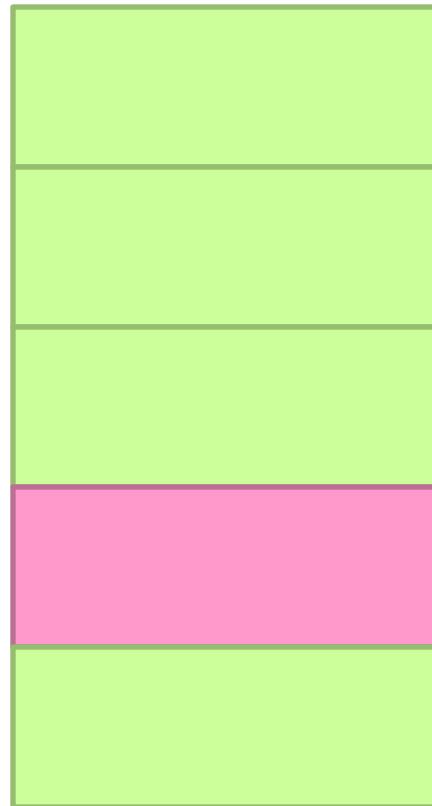
Cross-Validation

Training/Testing Splits



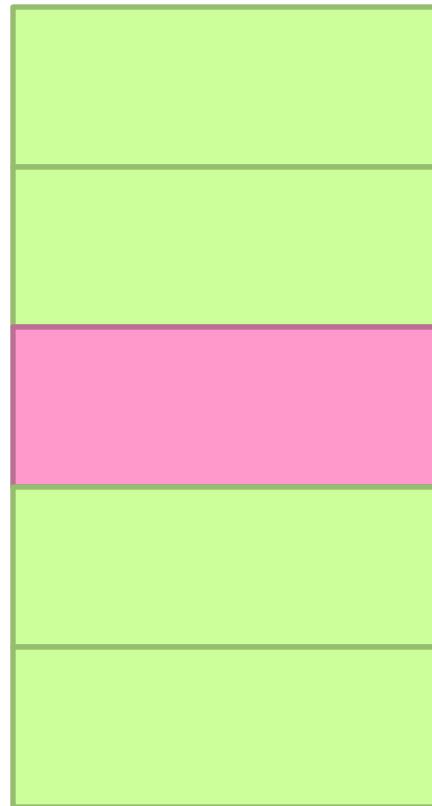
Cross-Validation

Training/Testing Splits



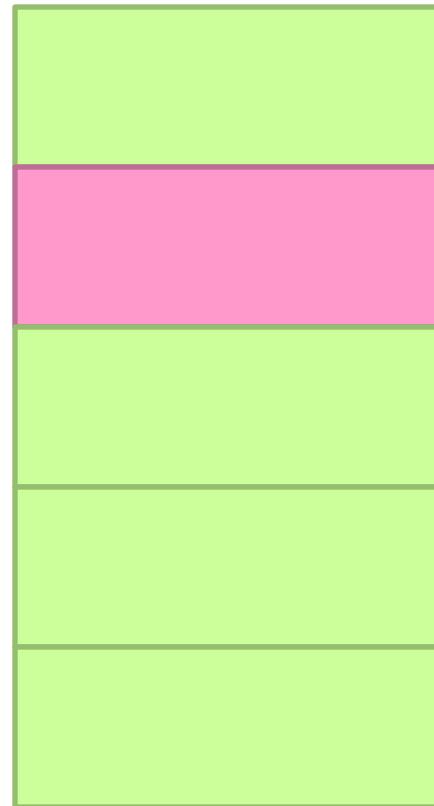
Cross-Validation

Training/Testing Splits



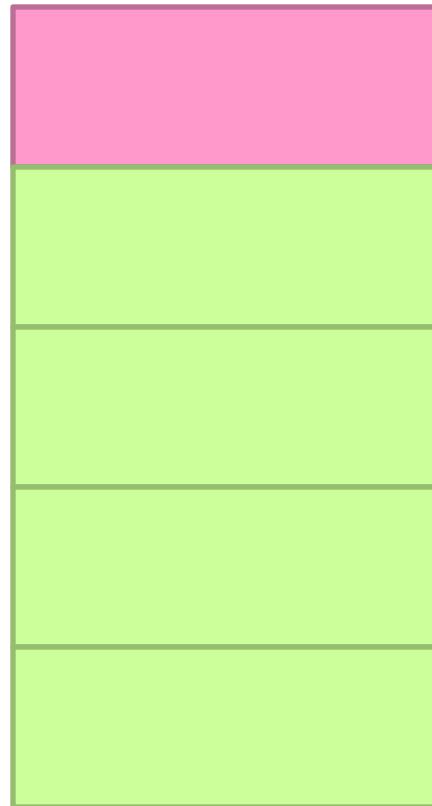
Cross-Validation

Training/Testing Splits



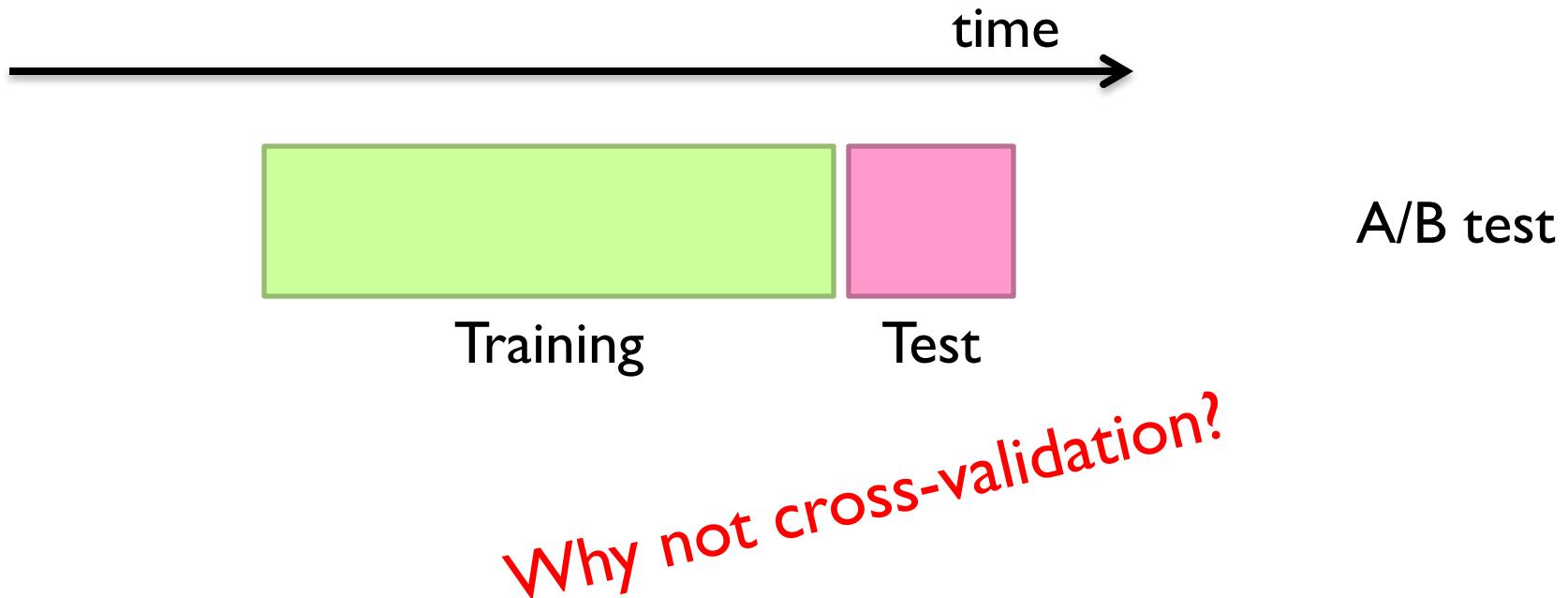
Cross-Validation

Training/Testing Splits

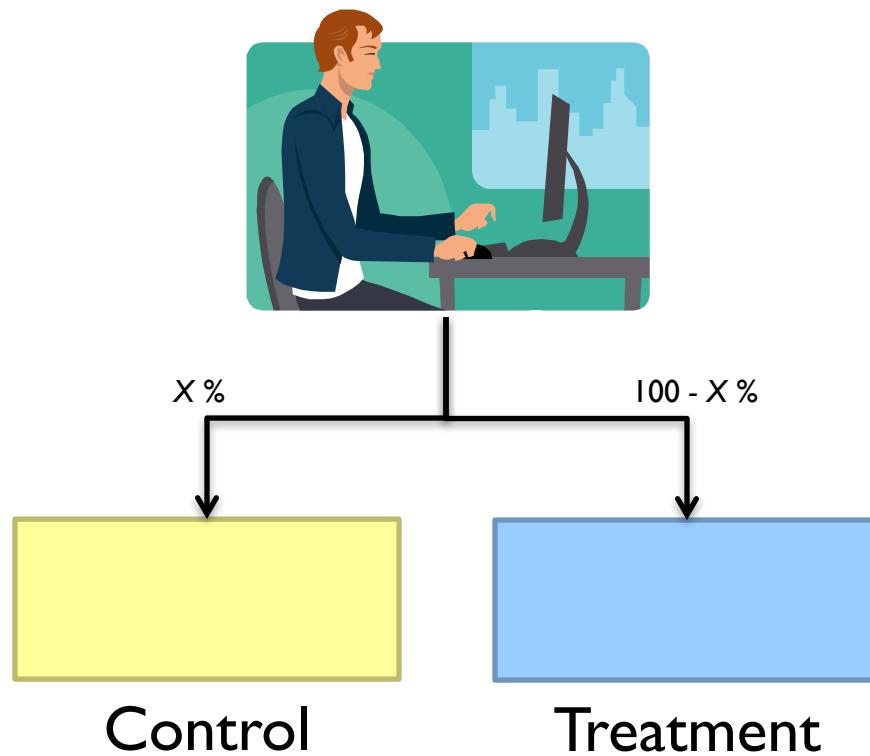


Cross-Validation

Typical Industry Setup



A/B Testing



Gather metrics, compare alternatives

A/B Testing: Complexities

Properly bucketing users

Novelty

Learning effects

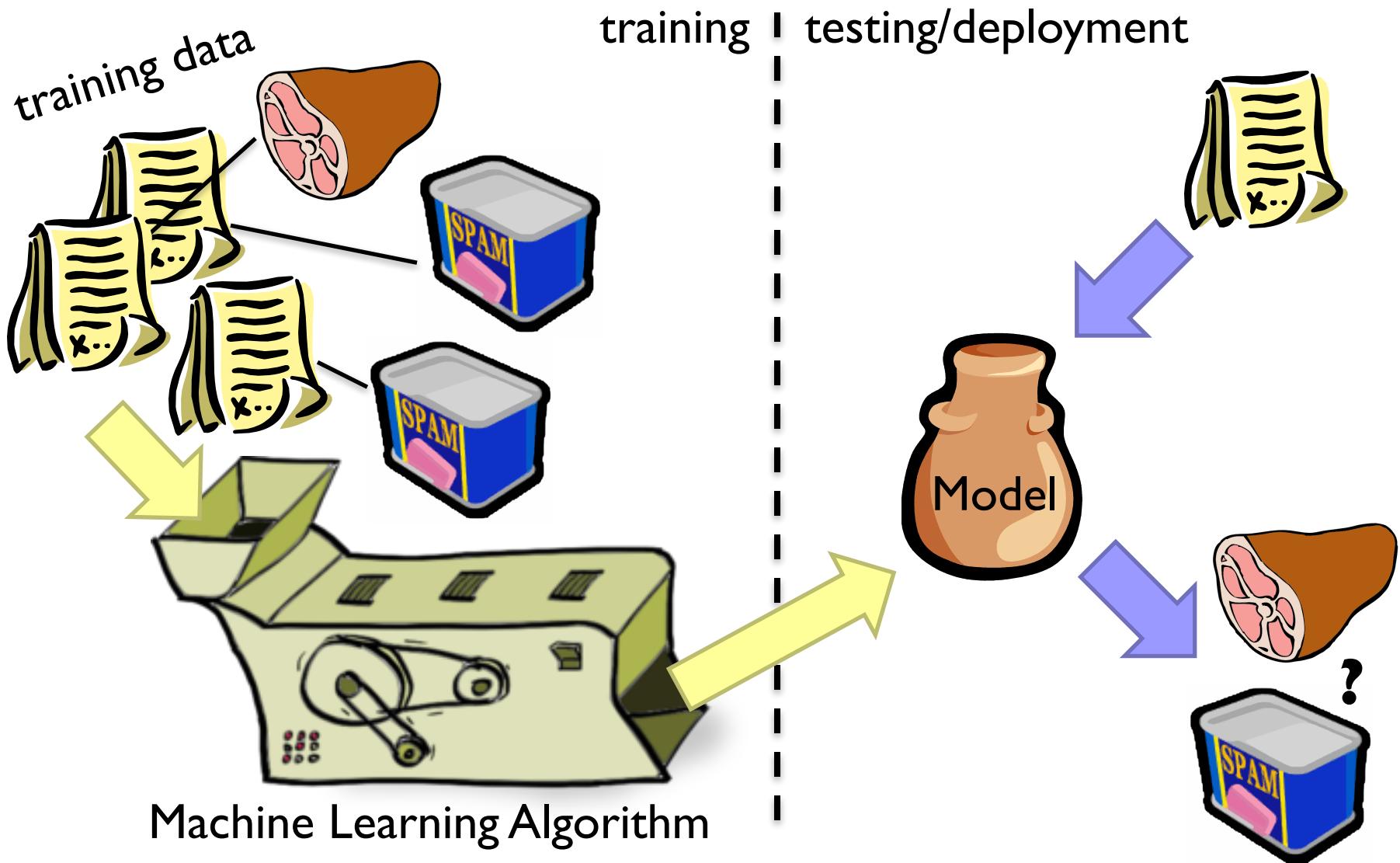
Long vs. short term effects

Multiple, interacting tests

Nosy tech journalists

...

Supervised Machine Learning



Applied ML in Academia

Download interesting dataset (comes with the problem)

Run baseline model

Train/Test

Build better model

Train/Test

Does new model beat baseline?

Yes: publish a paper!

No: try again!

THE SCIENTIFIC METHOD

Observe natural phenomena

Formulate Hypothesis

Modify Hypothesis

Test hypothesis
via rigorous Experiment

Establish Theory
based on repeated validation of results

www.phdcomics.com
JORGE CHAM © 2006

THE ACTUAL METHOD

Make up Theory
based on what
Funding Agency
Manager wants
to be true

Design minimum
experiments that
will ~~prove~~ show?
suggest Theory
is true

Modify Theory
to fit data

Publish Paper:
rename Theory a
“Hypothesis” and
pretend you used
the Scientific
Method

Defend Theory
despite all
evidence to the
contrary

DATA

Data Scientist: The Sexiest Job of the 21st Century

by **Thomas H. Davenport** and **D.J. Patil**

FROM THE OCTOBER 2012 ISSUE

Fantasy

Extract features

Develop cool ML technique

#Profit

Reality

What's the task?

Where's the data?

What's in this dataset?

What's all the f#\\$!* crap?

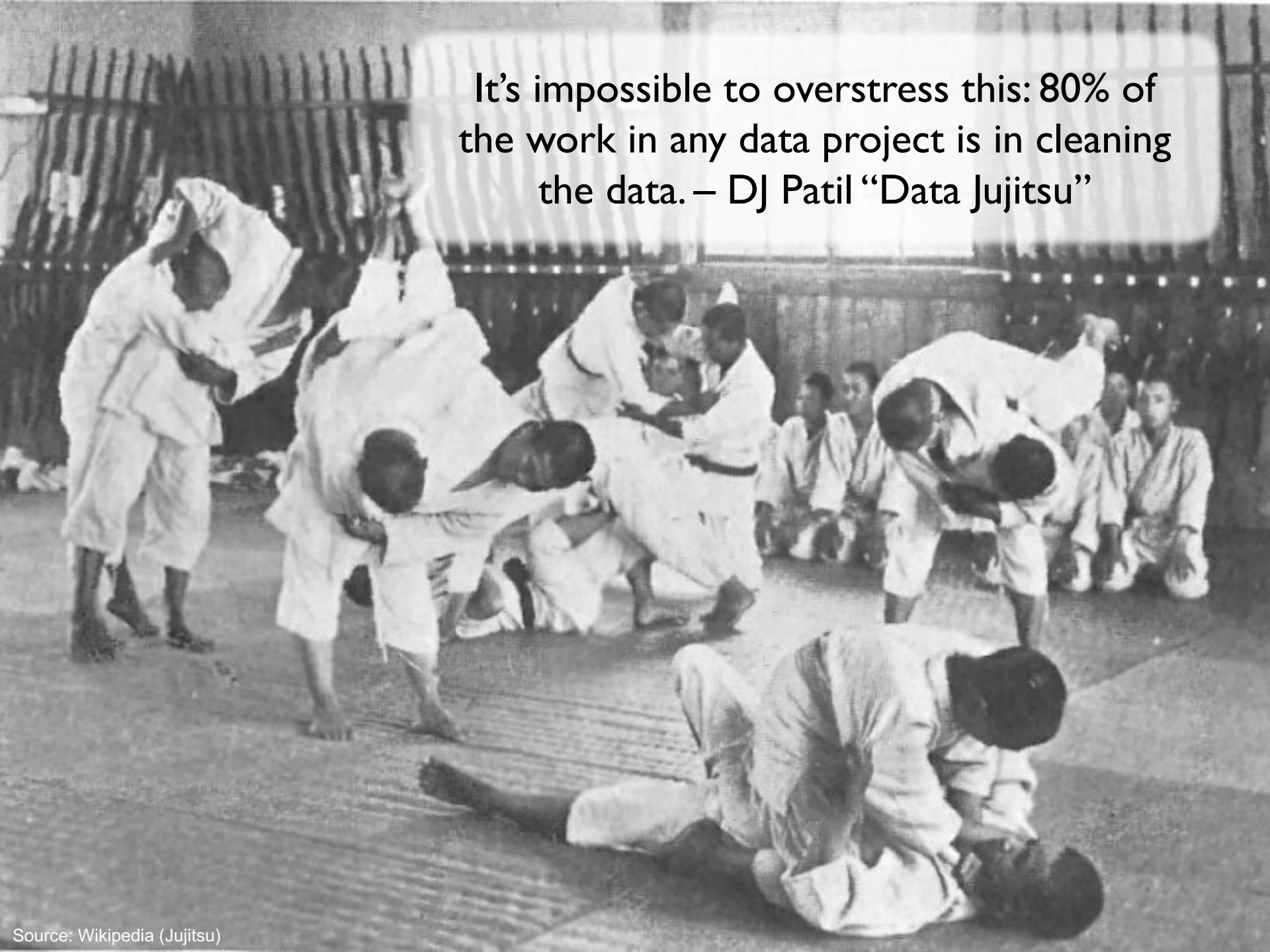
Clean the data

Extract features

“Do” machine learning

Fail, iterate...

Dirty secret: very little of data science is
about machine learning per se!



It's impossible to overstress this: 80% of the work in any data project is in cleaning the data. – DJ Patil “Data Jujitsu”

[SUBSCRIBE NOW](#)

[LOG IN](#)



The New York Times

≡ SECTIONS

HOME

SEARCH

TECHNOLOGY

For ‘Big Data’ Scientists, Hurdle to Insights Is ‘Janitor Work’

By STEVE LOHR AUG. 17, 2014



Monica Rogati, Jawbone's vice president for data science, with Brian Wilt, a senior data scientist.
Peter DaSilva for The New York Times

On finding things...



P. Oscar Boykin
@posco

OH: "... so to recap, tweets are statuses, favorites are favourings, retweets are shares."

Reply Retweet ★ Favorite ... More

On naming things...

CamelCase

uid *UserId*
userId

smallCamelCase

user_id *user_Id*

snake_case



Bill Graham
@billgraham

camel_Snake

Yesterday I had a run in with the
camel_Snake in our code. Today, I came
across the feared dunder_snake. Yow! /via
@THISWILLWORK

Reply Retweet Favorite More

dunder_snake

1 FAVORITE 
10:46 PM - Sep 12, 2012 from SoMa, San Francisco

On feature extraction...

```
^(\w+\s+\d+\s+\d+:\d+:\d+)\s+
([^\@]+?)@(\S+)\s+(\S+):\s+(\S+)\s+(\S+)
\s+((?:\S+, \s+)*(?:\S+))\s+(\S+)\s+(\S+)
\s+\[(^\[\]+)\]\s+"(\w+)\s+([^\"]\\"\")*
(?:\\\\. [^\"]\\"\")*)*\s+(\S+)\s+"\\s+(\S+)\s+
(\S+)\s+"([^\"]\\"\")*(?:\\\\. [^\"]\\"\")*)*
"\s+"([^\"]\\"\")*(?:\\\\. [^\"]\\"\")*)*\s*
(\d*- [\d-] *)?\s*(\d+)?\s*(\d*\.\[ \d\.\] *)?
(\s+[-\w]+)?.*$
```

An actual Java regular expression used to parse log message at Twitter circa 2010

Friction is cumulative!



Data Plumbing...

Gone Wrong!

[scene: consumer internet company in the Bay Area...]

Frontend Engineer

It's over here...

Well, it wouldn't fit, so we had to shoehorn...

Hang on, I don't remember...

Uh, bad news. Looks like we forgot to log it...

Data Scientist

Okay, let's get going... where's the click data?

Well, that's kinda non-intuitive, but okay...

Oh, BTW, where's the timestamp of the click?

[grumble, grumble, grumble]

Frontend Engineer

Develops new feature, adds
logging code to capture clicks

Data Scientist

Analyze user behavior, extract
insights to improve feature

Fantasy

Extract features

Develop cool ML technique

#Profit

Reality

What's the task?

Where's the data?

What's in this dataset?

What's all the f#\\$!* crap?

Clean the data

Extract features

“Do” machine learning

Fail, iterate...

Finally works!

A wide-angle photograph of a rural landscape. In the foreground, there are rolling green hills with some brown, possibly harvested, fields. The middle ground shows more hills and a valley. In the background, there are distant mountains. The sky is blue with scattered white and grey clouds.

Congratulations, you're halfway there...

Congratulations, you're halfway there...

Does it actually work?
A/B testing

Is it fast enough?

Good, you're two thirds there...

Productionize



Productionize

What are your jobs' dependencies?

How/when are your jobs scheduled?

Are there enough resources?

How do you know if it's working?

Who do you call if it stops working?

Infrastructure is critical here!

(plumbing)



Takeaway lessons:
Most of data science isn't glamorous!

A photograph of a traditional Japanese rock garden. In the foreground, a gravel path is raked into fine, parallel lines. Several large, dark, irregular stones are scattered across the garden. A small, shallow pond is visible in the middle ground, surrounded by more stones and some low-lying green plants. In the background, there are more stones, some small trees, and a building with a tiled roof. The overall atmosphere is peaceful and minimalist.

Questions?