



# Big Data Infrastructure

CS 489/698 Big Data Infrastructure (Winter 2016)

Week 8: Data Mining (2/4)

March 3, 2016

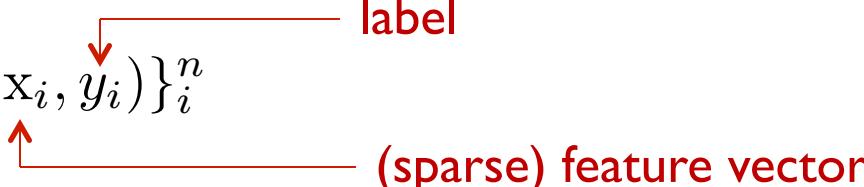
Jimmy Lin

David R. Cheriton School of Computer Science  
University of Waterloo

These slides are available at <http://lintool.github.io/bigdata-2016w/>

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


↓ label  
↑ (sparse) feature vector

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

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

- Induce  $f : X \rightarrow Y$ 
  - Such that loss is minimized
- $$\frac{1}{n} \sum_{i=0}^n \ell(f(x_i), y_i)$$


↑ loss function
- Typically, consider functions of a parametric form:

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


↑ model parameters

The background image shows a wide, open landscape with rolling green hills under a bright blue sky filled with large, white, fluffy clouds. The foreground is a grassy field with some dirt paths.

# Gradient Descent

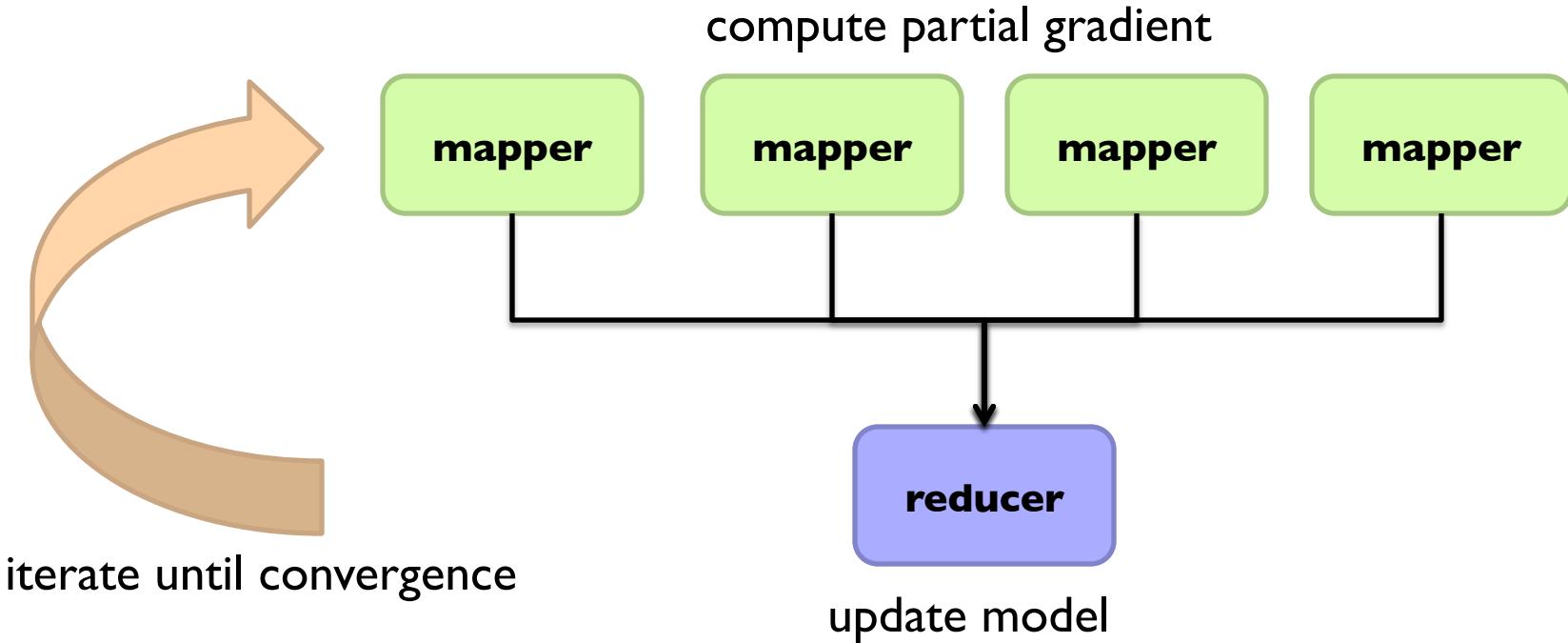
$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \frac{1}{n} \sum_{i=0}^n \nabla \ell(f(\mathbf{x}_i; \theta^{(t)}), y_i)$$

# 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

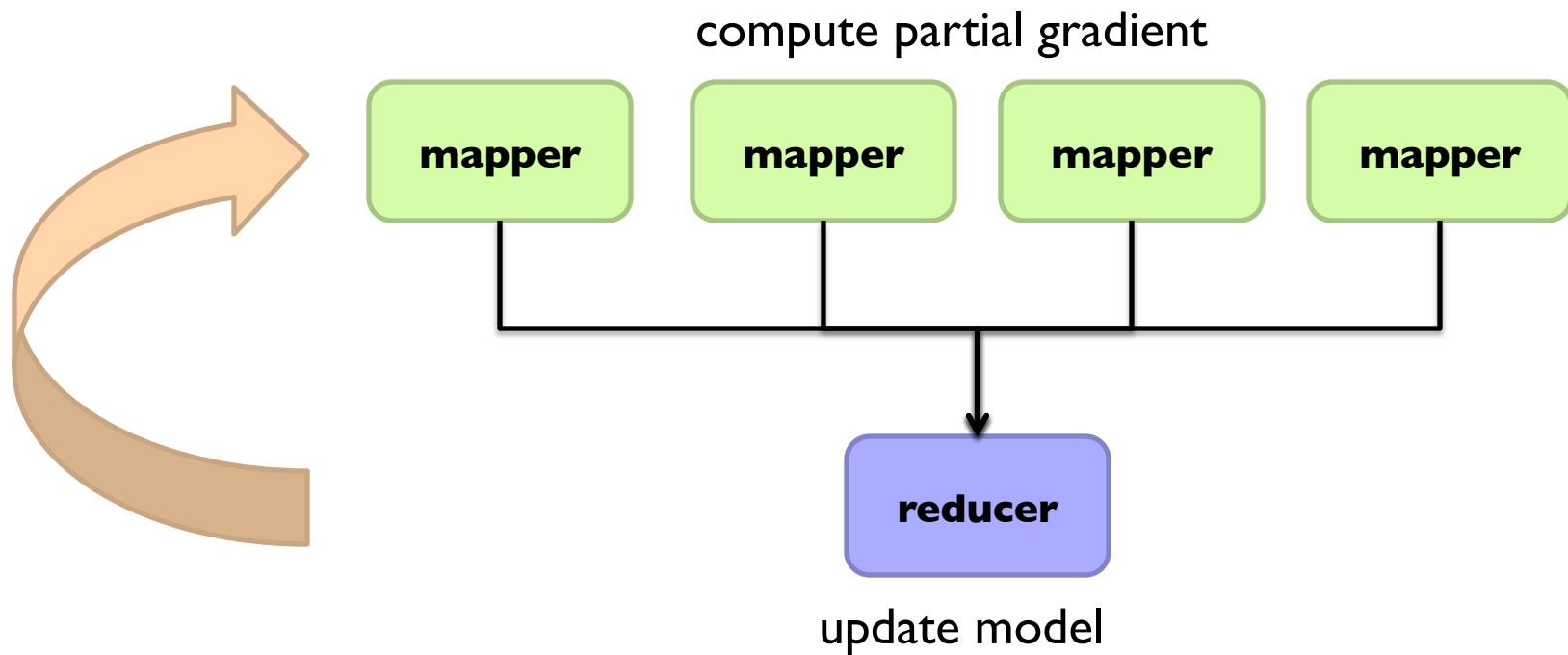


# Spark Implementation

```
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

A photograph of a vibrant water park featuring a complex network of multi-colored slides (yellow, blue, green, orange, purple) winding through a steel frame structure. In the foreground, a large yellow splash pool is visible where people are playing. The background shows a clear blue sky with scattered white clouds and some green trees.

# 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!

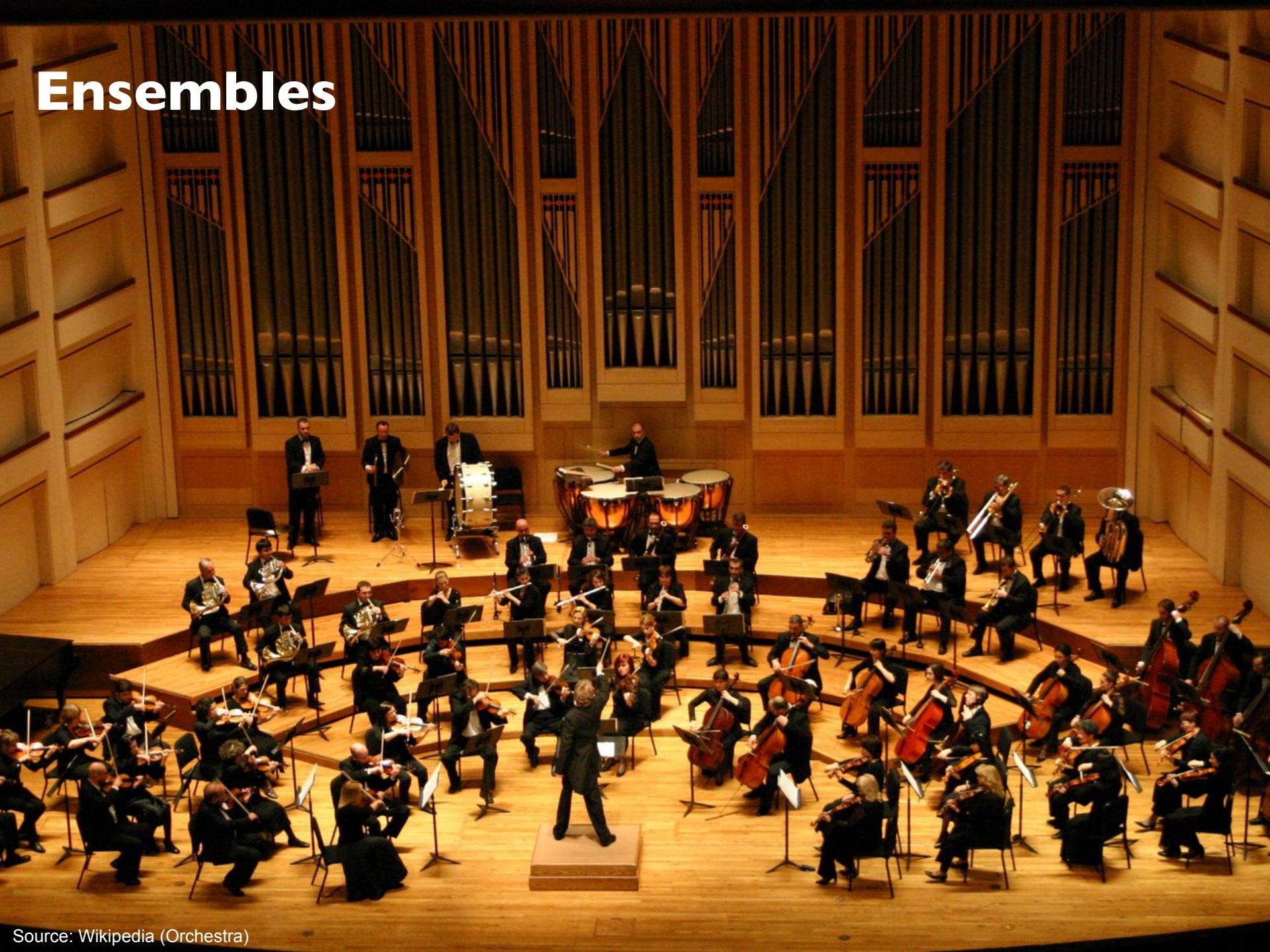
# Practical Notes

- Order of the instances important!
- Most common implementation:
  - Randomly shuffle training instances
  - Stream instances through learner
- Single vs. multi-pass approaches
- “Mini-batching” as a middle ground between batch and stochastic gradient descent

We've solved the iteration problem!

What about the single reducer problem?

# Ensembles



# Ensemble Learning

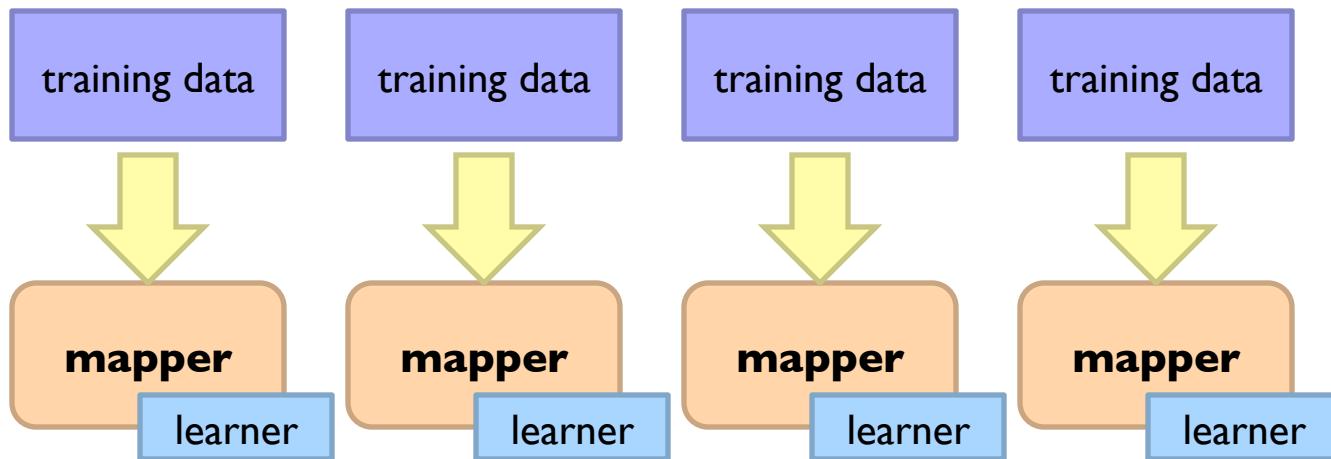
- Learn multiple models, combine results from different models to make prediction
- Why does it work?
  - If errors uncorrelated, multiple classifiers being wrong is less likely
  - Reduces the variance component of error
- A variety of different techniques:
  - Majority voting
  - Simple weighted voting:
$$y = \arg \max_{y \in Y} \sum_{k=1}^n \alpha_k p_k(y|x)$$
  - Model averaging
  - ...

# Practical Notes

- Common implementation:
  - Train classifiers on different input partitions of the data
  - Embarrassingly parallel!
- Contrast with other ensemble techniques, e.g., boosting

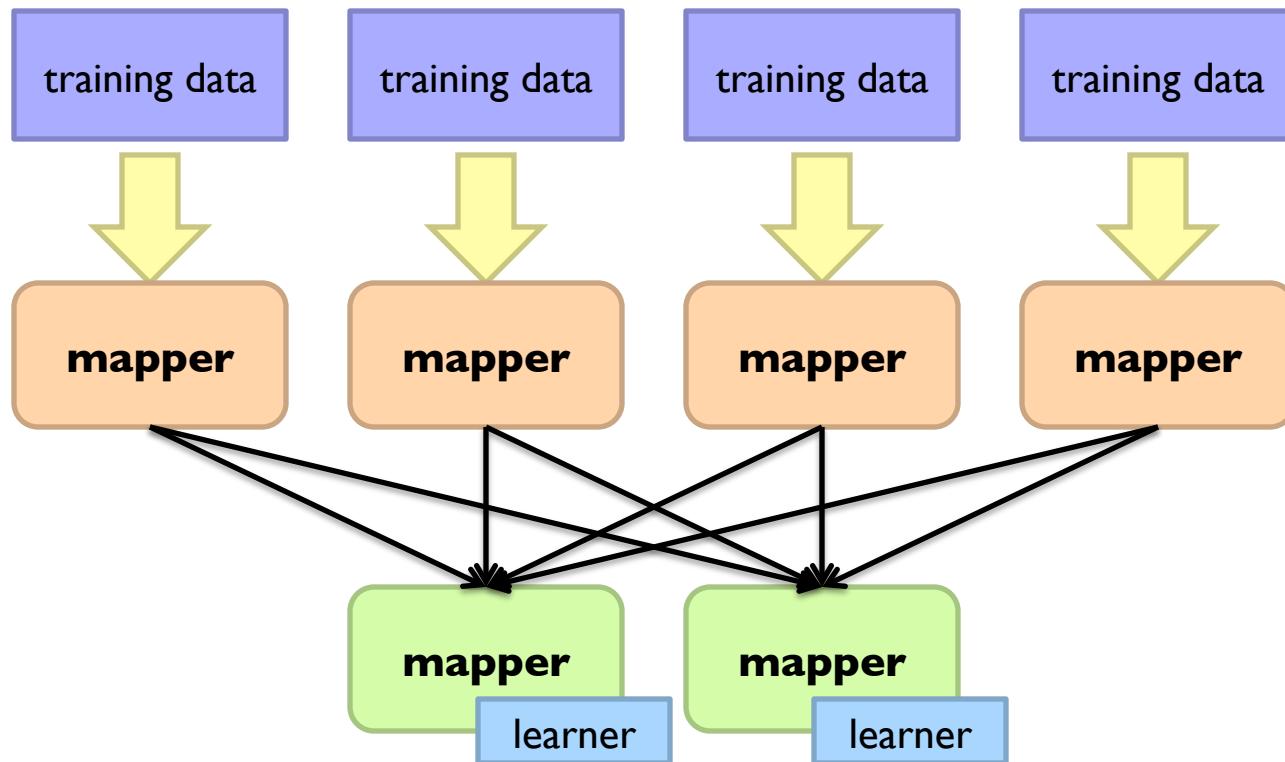
# MapReduce Implementation

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



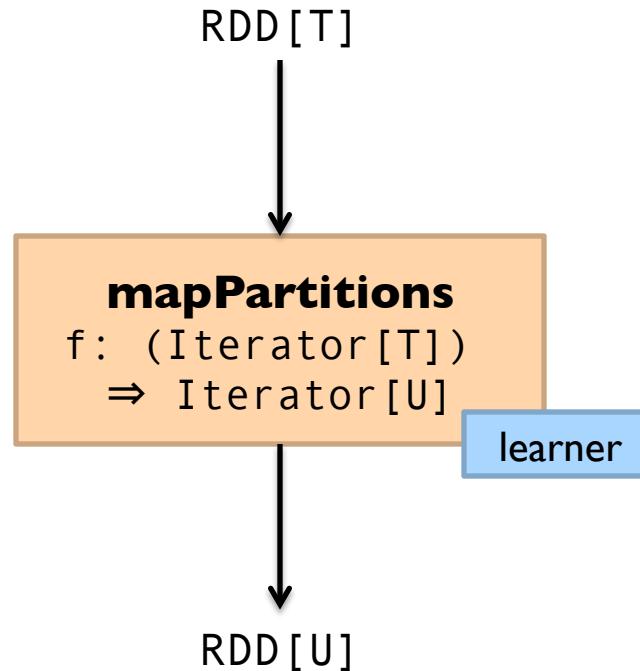
# MapReduce Implementation

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



# What about Spark?

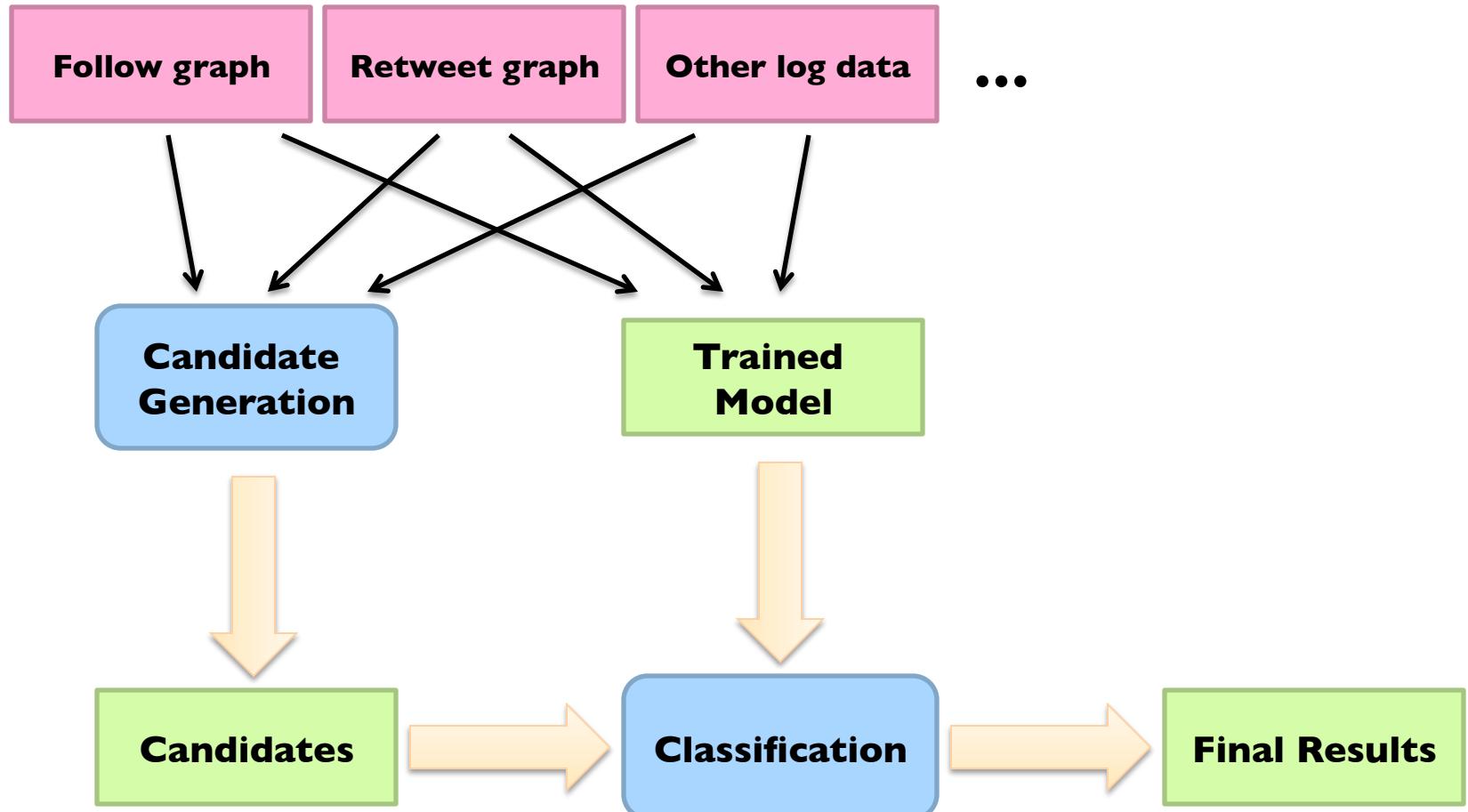
$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \nabla \ell(f(\mathbf{x}; \theta^{(t)}), y)$$



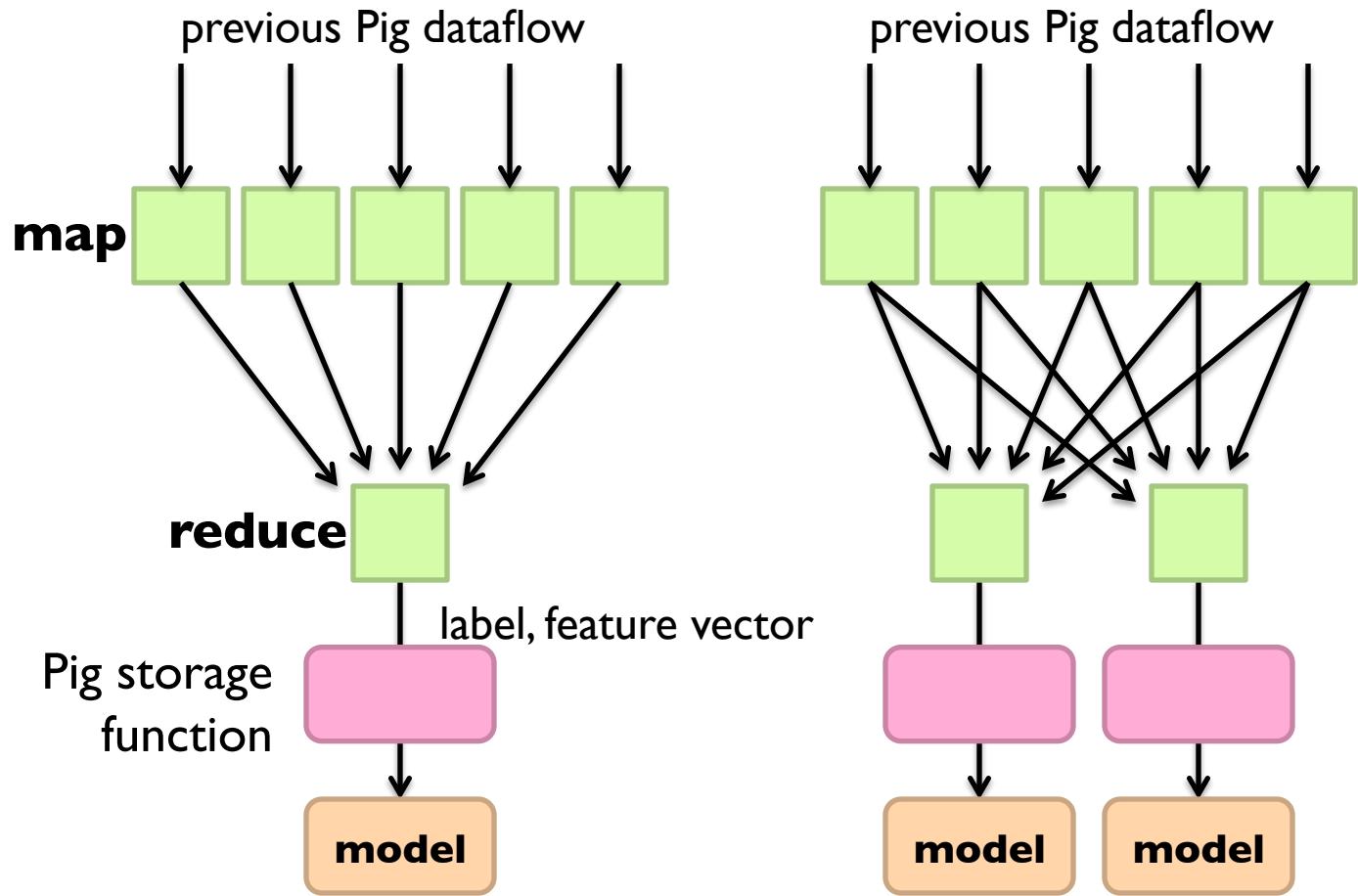
# MapReduce Implementation: Details

- Two possible implementations:
  - Write model out as “side data”
  - Emit model as intermediate output

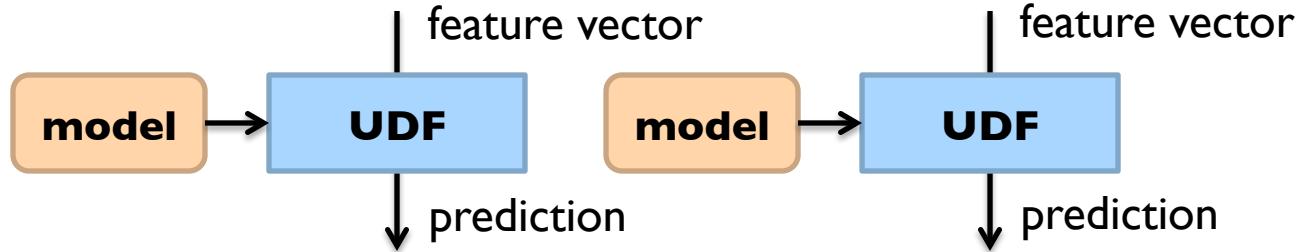
# Case Study: Link Recommendation



# 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?

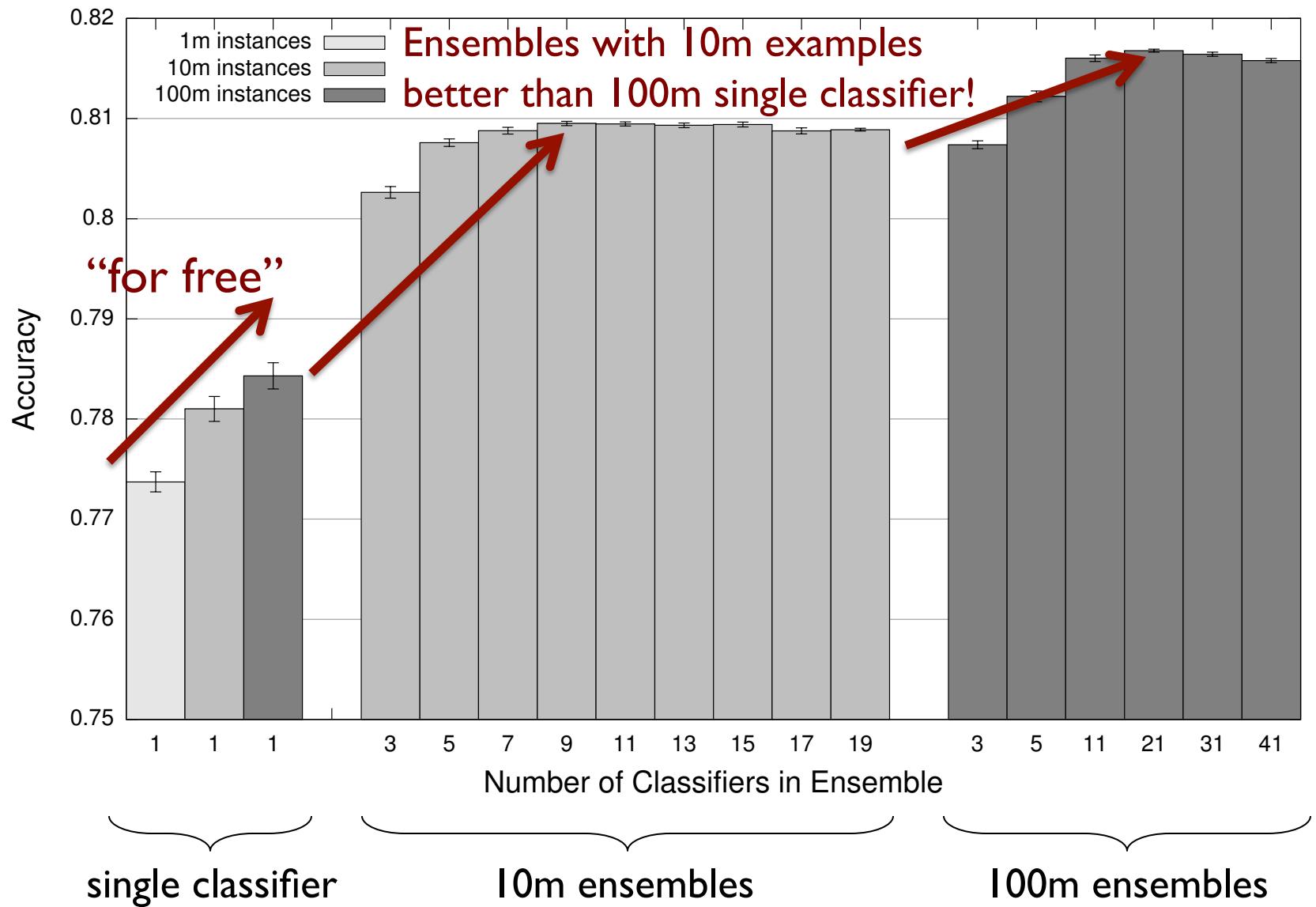
```
define Classify ClassifyWithEnsemble('model',
    'classifier.LR', 'vote');
```

# Sentiment Analysis Case Study

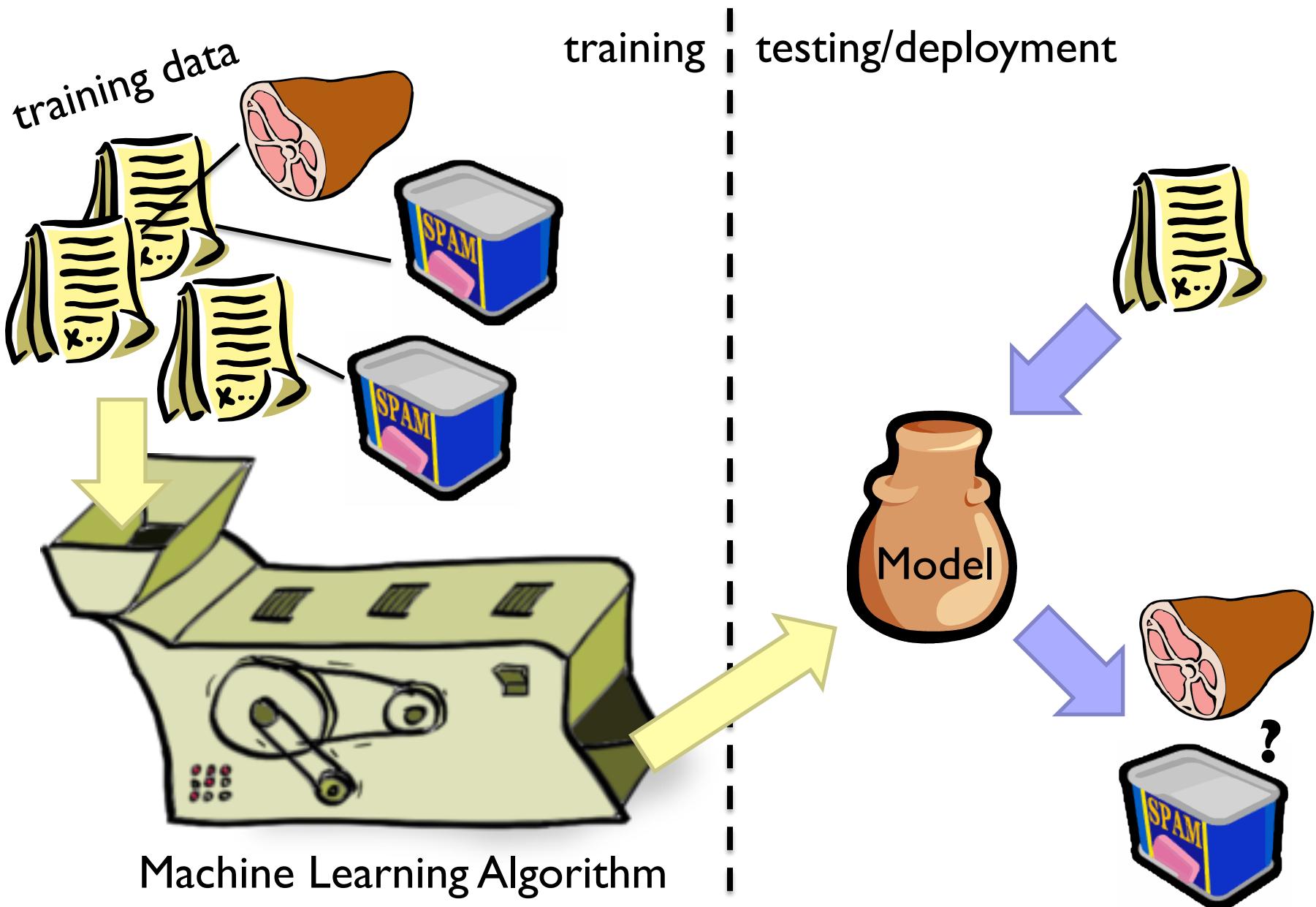
Lin and Kolcz, SIGMOD 2012

- Binary polarity classification: {positive, negative} sentiment
  - Independently interesting task
  - Illustrates end-to-end flow
  - 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:
  - Logistic regression with SGD (L2 regularization)
  - Ensembles of various sizes (simple weighted voting)

Diminishing returns...



# Supervised Machine Learning



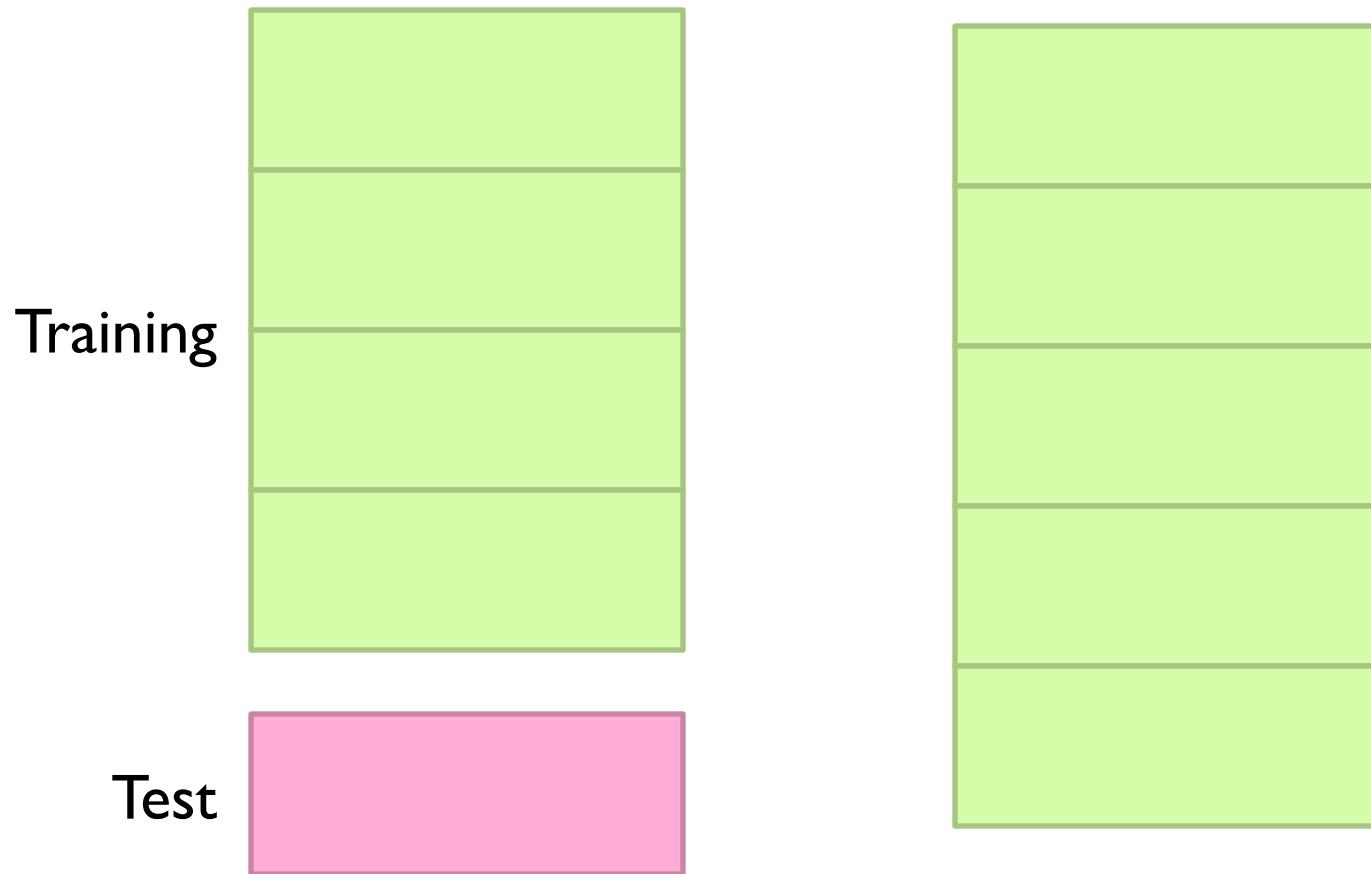
# Applied ML in Academia

- 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!

# Three Commandants of Machine Learning

Thou shalt not mix training and testing data  
Thou shalt not mix training and testing data  
Thou shalt not mix training and testing data

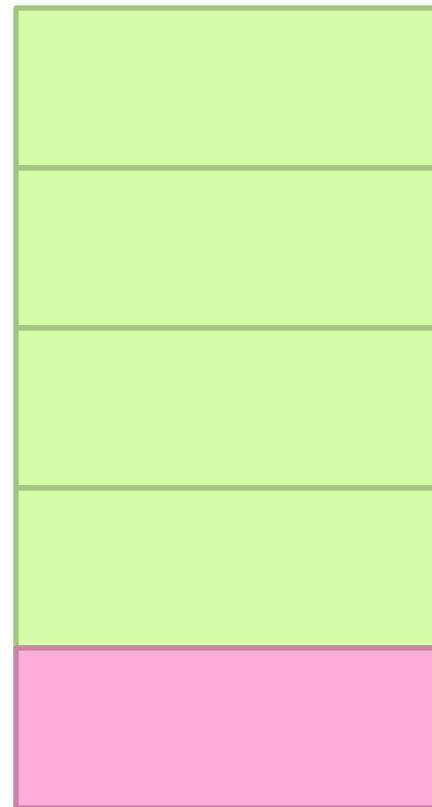
# Training/Testing Splits



What happens if you need more?

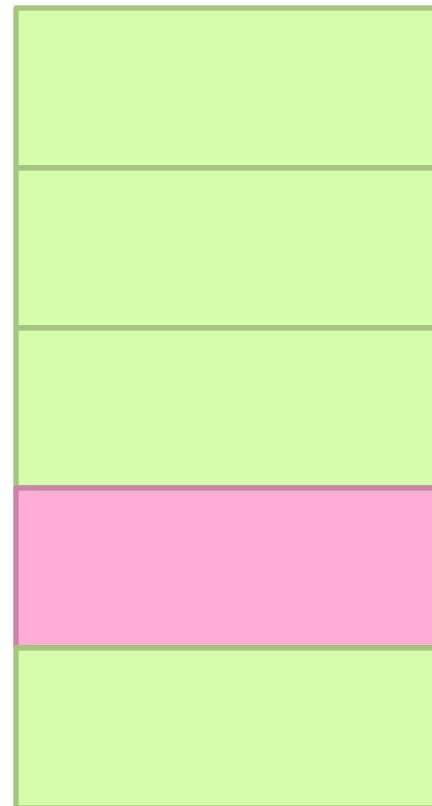
Cross-Validation

# Training/Testing Splits



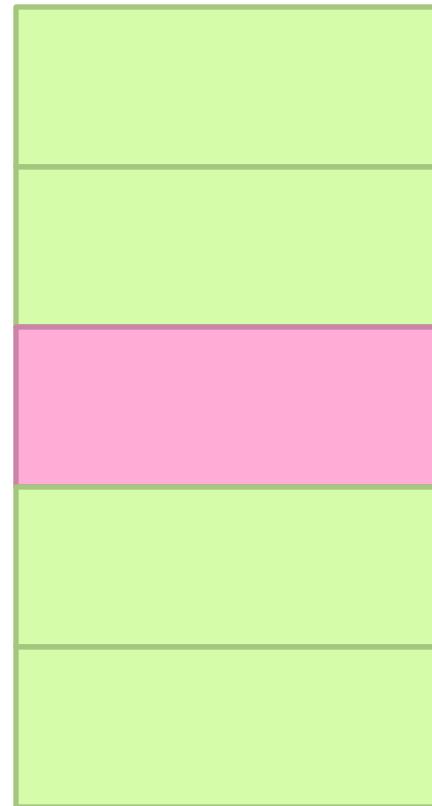
Cross-Validation

# Training/Testing Splits



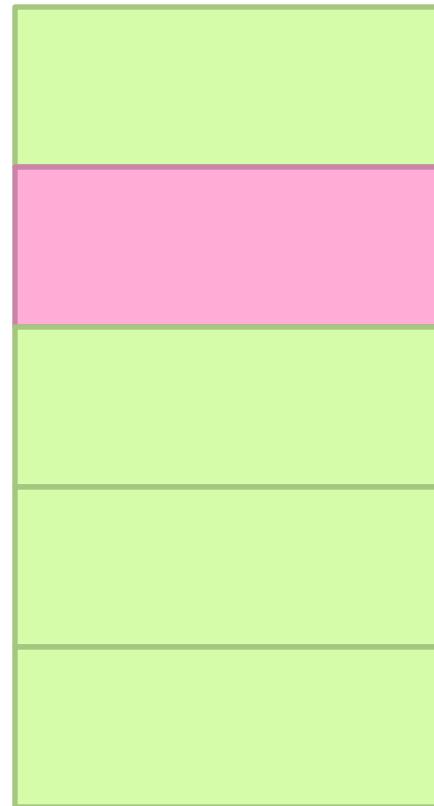
Cross-Validation

# Training/Testing Splits



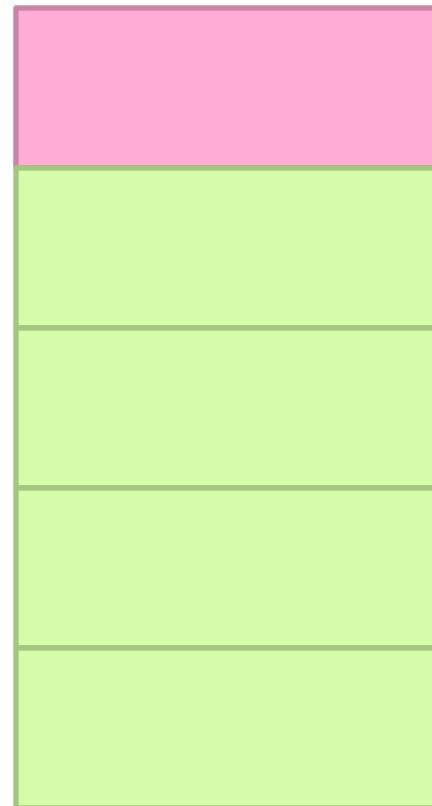
Cross-Validation

# Training/Testing Splits



Cross-Validation

# Training/Testing Splits



Cross-Validation

# 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?

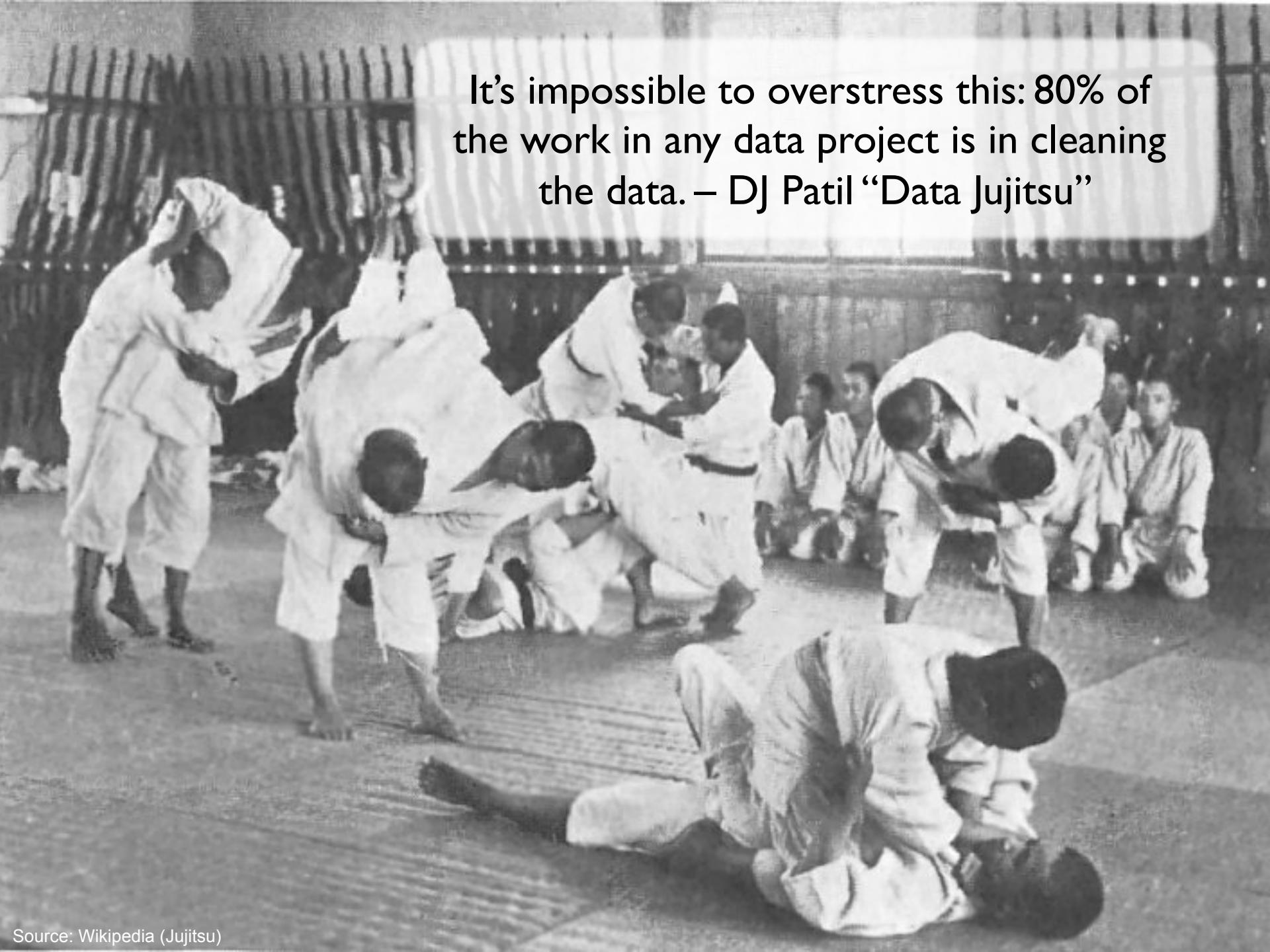
What's all the f#\$!\* crap?

Clean the data

Extract features

“Do” machine learning

Fail, iterate...



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

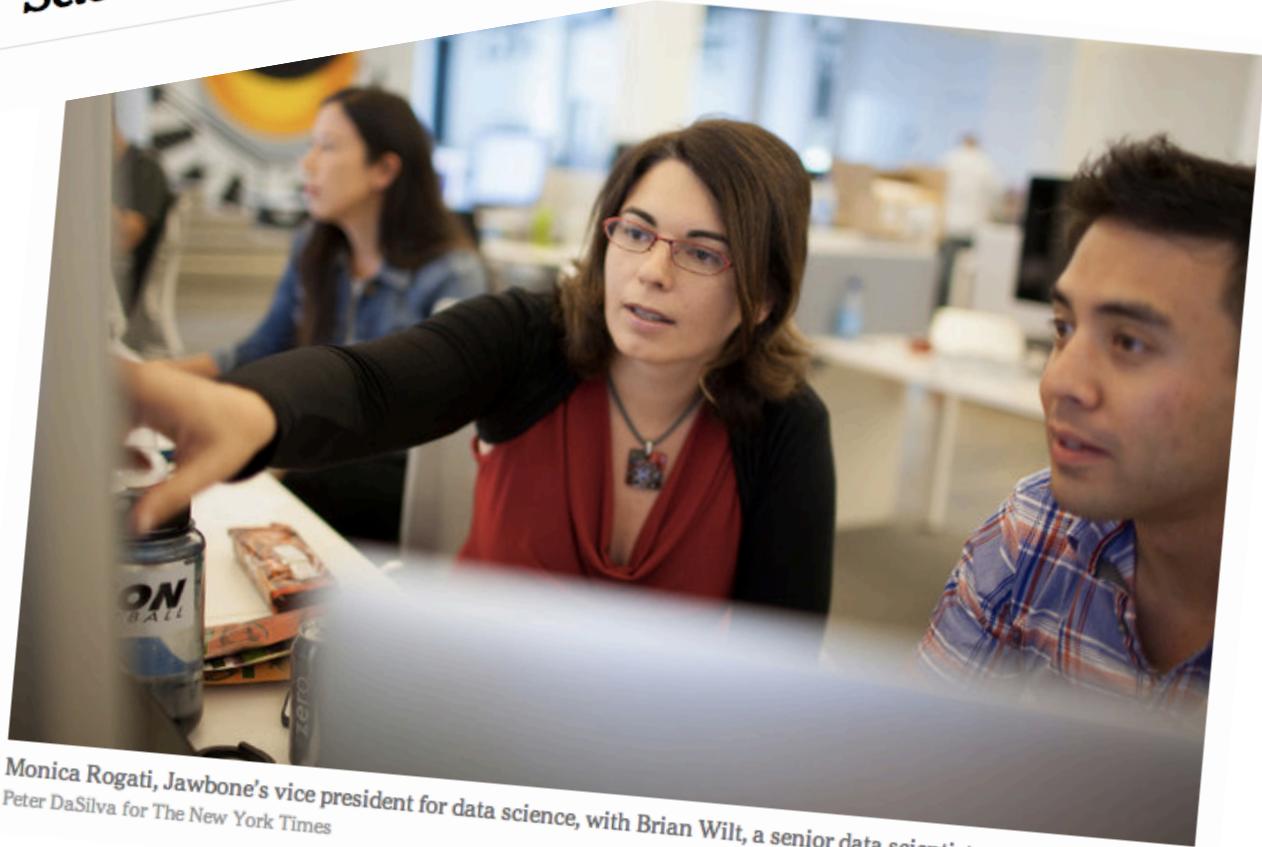
HOME

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

Following

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

Reply Retweet ★ Favorite ... More

# On finding things...

CamelCase

smallCamelCase

snake\_case

camel\_Snake

dunder\_\_snake

uid      *UserId*  
userId      *userid*

*user\_id*      *user\_Id*



Bill Graham  
@billgraham

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

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!

# 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)

A photograph showing a complex network of pipes and valves in a basement or utility room. The pipes are made of various materials, including white PVC and larger yellowish-brown pipes. A prominent red valve is visible in the center. The ceiling is made of drywall, which appears aged and slightly peeling. The lighting is warm and somewhat dim, creating deep shadows in the recesses of the pipe network.

**Takeaway lesson:**  
Plumbing matters!

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 low-lying green plants. In the background, there are more trees and shrubs, and the wooden buildings of a residence are visible behind the garden wall.

Questions?