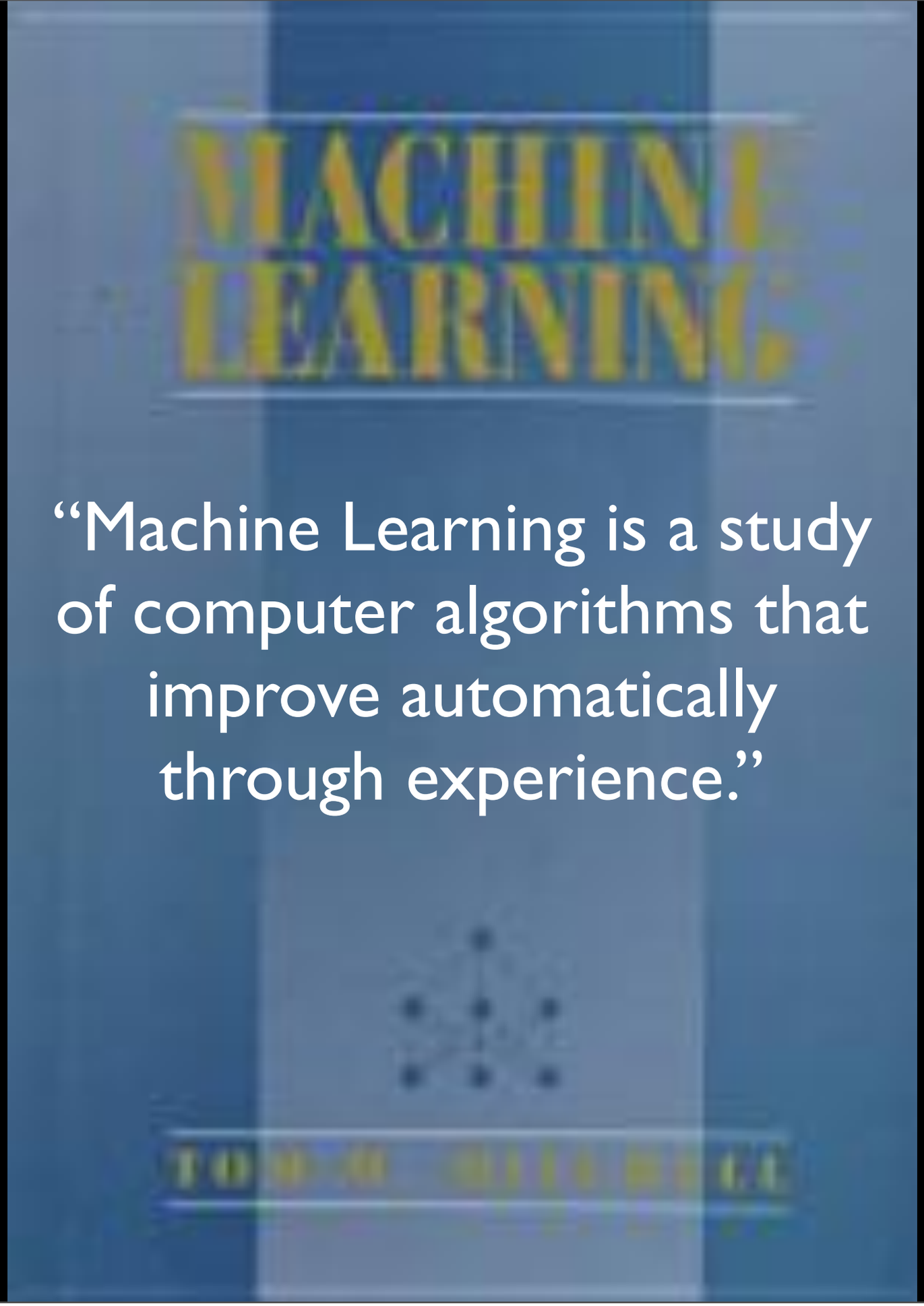
Google Prediction API and Distributed Machine Learning

Max Lin
Software Engineer | Google

HPC and Cloud Computing Workshop | LBNL & CITRIS
UC Berkeley | June 22nd, 2011

Outline

- What is Machine Learning?
- Why Machine Learning on the Cloud?
- What is Google Prediction API?
- How to scale up machine learning algorithms?



"Machine Learning is a study of computer algorithms that improve automatically through experience."

The New York Times

Science

SMARTER THAN YOU THINK

Google Cars Drive Themselves, in Traffic

Ramin Rahimian for The New York Times

Google

when was nsa founded

Search

About 3,110,000 results (0.23 seconds)

Advanced search

- Everything
- Images
- Videos
- News
- Shopping
- More

Mountain View, CA
Change location

All results
Timeline

More search tools

Best guess for **NSA Founded** is **1952** - Feedback
Mentioned on at least 5 websites including infoplease.com, historycommons.org and answers.com -
Show sources

National Security Agency — Infoplease.com
National Security Agency (**NSA**), an independent agency within the U.S. Dept. of Defense. **Founded** by presidential order in **1952**, its primary function is to **...**
www.infoplease.com/ce6/history/A0834970.html - Cached - Similar

Context of '**1952: NSA Founded**'
It contains events related to the event **1952**: **NSA Founded**. You can narrow or broaden the context of this timeline by adjusting the zoom level. **...**
www.historycommons.org/context.jsp?item=civilliberties_88 - Cached - Similar

National Security Agency: Information from Answers.com
National Security Agency (**NSA**), an independent agency within the U.S. Dept. of Defense. **Founded** by presidential order in **1952**, its primary function is to **...**
Organization - Effect on non-governmental ... - NSANet - NSA programs
www.answers.com/topic/national-security-agency - Cached - Similar

Google

when was nsa founded

About 3,110,000 results (0.23 seconds)

- Everything
- Images
- Videos
- News
- Shopping
- More

Mountain View, CA
Change location

▶ Best guess for **NSA Founded** is **1952** - Feedback
Mentioned on at least 5 websites including infoplease.com, hist
Show sources

National Security Agency — Infoplease.com
National Security Agency (**NSA**), an independent agency withir
Founded by presidential order in **1952**, its primary function is to
www.infoplease.com/ce6/history/A0834970.html - Cached - Sin

Context of '**1952**: **NSA Founded**'
It contains events related to the event **1952**: **NSA Founded**. Yo
context of this timeline by adjusting the zoom level. ...
www.historycommons.org/context.jsp?item=civilliberties_88 - C

Google Translate

translate.google.com/#

Web Images Videos Maps News Shopping Gmail more ▼

Google translate

From: English ▼ To: Chinese… ▼ Translate

to be or not to be -- that is the question

Listen

English to Chinese (Traditional) translation

生存還是毀滅是 -這是個問題

Listen Read phonetically

New! Click the words above to view alternate translations.
Dismiss

Google Translate for my: Searches Videos Email Phone Chat Business

About Google Translate Turn off instant translation Privacy Help

The quick brown fox
jumped over the lazy dog.

The quick brown fox
jumped over the lazy dog.

English

The quick brown fox
jumped over the lazy dog.

To err is human, but to
really foul things up you
need a computer.

English

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	?

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	?
La fe mueve montañas.	

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	?
La fe mueve montañas.	?

Training

The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	?
La fe mueve montañas.	?

Training Input X	
The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	?
La fe mueve montañas.	?

Training Input X	Output Y
The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish
To be or not to be -- that is the question	?
La fe mueve montañas.	?

Training Input X	Output Y
The quick brown fox jumped over the lazy dog.	English
To err is human, but to really foul things up you need a computer.	English
No hay mal que por bien no venga.	Spanish
La tercera es la vencida.	Spanish

Model f(x)

To be or not to be -- that is the question	?
La fe mueve montañas.	?

Training	Input X	Output Y
	The quick brown fox jumped over the lazy dog.	English
	To err is human, but to really foul things up you need a computer.	English
	No hay mal que por bien no venga.	Spanish
	La tercera es la vencida.	Spanish

Model f(x)

Testing		
	To be or not to be -- that is the question	?
	La fe mueve montañas.	?

Training	Input X	Output Y
	The quick brown fox jumped over the lazy dog.	English
	To err is human, but to really foul things up you need a computer.	English
	No hay mal que por bien no venga.	Spanish
	La tercera es la vencida.	Spanish

Model f(x)

Testing		
	To be or not to be -- that is the question.	?
	La fe mueve montañas.	?

f(x')

Training	Input X	Output Y
	The quick brown fox jumped over the lazy dog.	English
	To err is human, but to really foul things up you need a computer.	English
	No hay mal que por bien no venga.	Spanish
	La tercera es la vencida.	Spanish

Model f(x)

Testing		
	To be or not to be -- that is the question.	?
	La fe mueve montañas.	?

f(x’) = y’

Linear Classifier

Linear Classifier

The quick brown fox jumped over the lazy dog.

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ... ‘the’

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ... ‘the’ ...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ... ‘the’ ... ‘montañas’

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ... ‘the’ ... ‘montañas’ ...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ... ‘the’ ... ‘montañas’ ...

0,

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’ ... ‘aardvark’ ... ‘dog’ ... ‘the’ ... ‘montañas’ ...

0, ...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...			0,					

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...			0,		...			

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...	0,			...	1,			

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...	0,	...	1,	...				

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...	0,	...	1,	...	1,			

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...	0,	...	1,	...	1,	...		

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...	0,	...	1,	...	1,	...	0,	

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
0,	...	0,	...	1,	...	1,	...	0,	...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
[0,	...	0,	...	1,	...	1,	...	0,	...

Linear Classifier

The quick brown fox jumped over the lazy dog.

‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
[0,	...	0,	...	1,	...	1,	...	0,	...]

Linear Classifier

The quick brown fox jumped over the lazy dog.

	‘a’	...	‘aardvark’	...	‘dog’	...	‘the’	...	‘montañas’	...
x [	0,	...	0,	...	1,	...	1,	...	0,	...]

Linear Classifier

The quick brown fox jumped over the lazy dog.

	'a'	...	'aardvark'	...	'dog'	...	'the'	...	'montañas'	...
x	[0,	...	0,	...	1,	...	1,	...	0,	...]
	[0.1,	...	132,	...	150,	...	200,	...	-153,	...]

Linear Classifier

The quick brown fox jumped over the lazy dog.

	'a'	...	'aardvark'	...	'dog'	...	'the'	...	'montañas'	...
x	[0,	...	0,	...	1,	...	1,	...	0,	...]
w	[0.1,	...	132,	...	150,	...	200,	...	-153,	...]

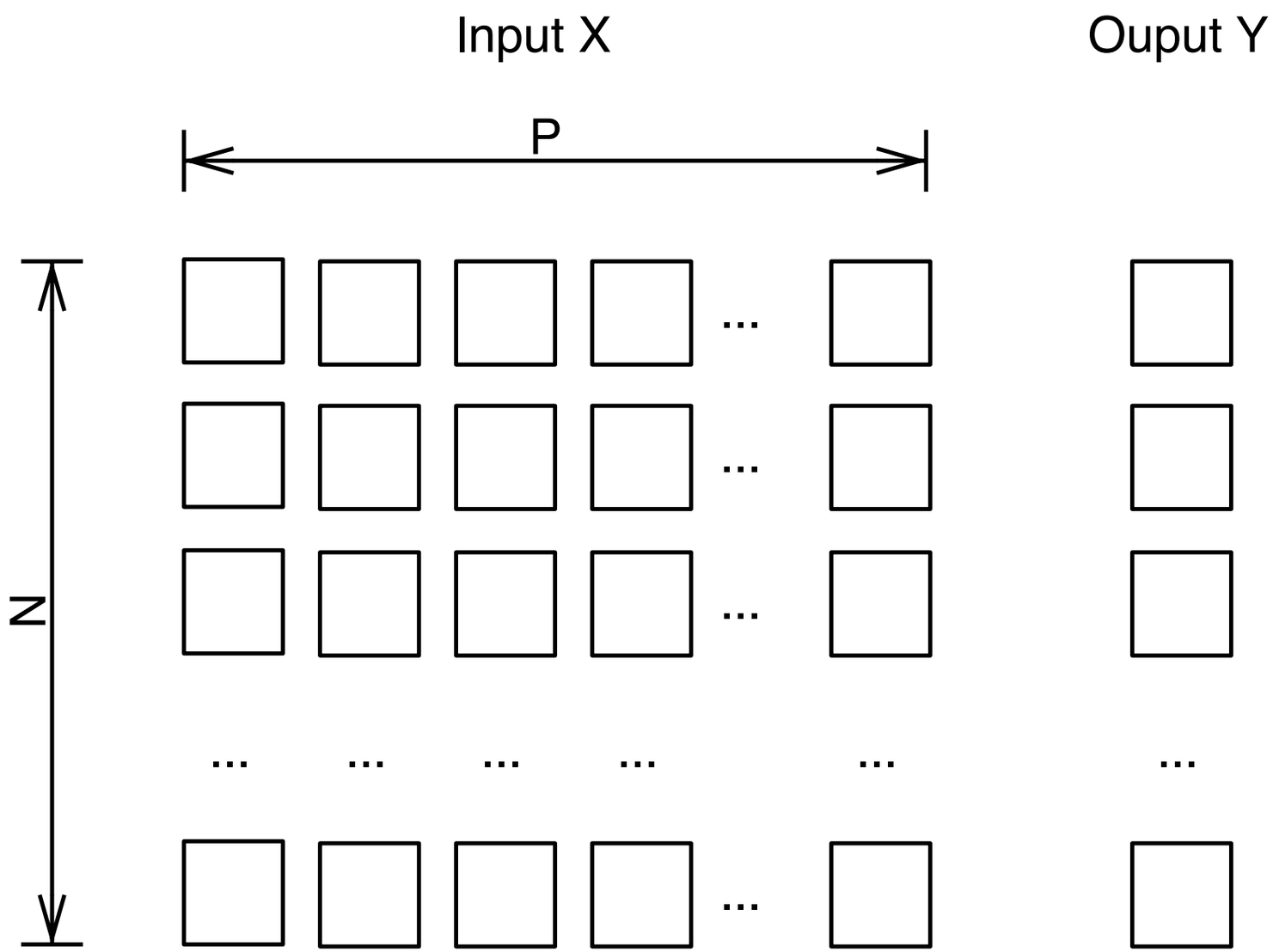
Linear Classifier

The quick brown fox jumped over the lazy dog.

	'a'	...	'aardvark'	...	'dog'	...	'the'	...	'montañas'	...
x	[0,	...	0,	...	1,	...	1,	...	0,	...]
w	[0.1,	...	132,	...	150,	...	200,	...	-153,	...]

$$f(x) = \mathbf{w} \cdot \mathbf{x} = \sum_{p=1}^{P} w_p * x_p$$

Training Data



Typical machine learning data at Google

N: 100 billions / 1 billion

P: 1 billion / 10 million

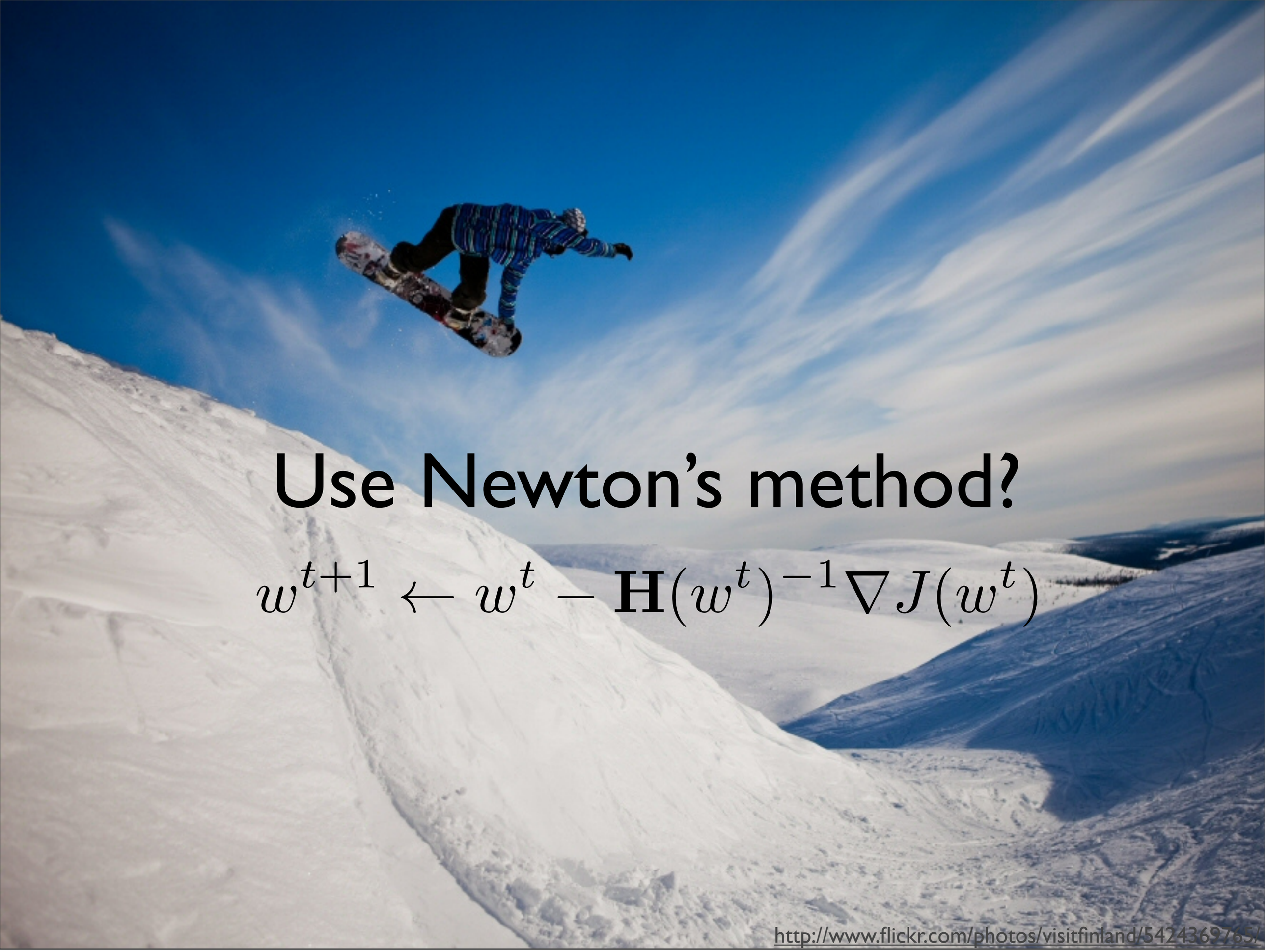
(mean / median)

http://www.flickr.com/photos/mr_t_in_dc/5469563053

Classifier Training

- Training: Given {(x, y)} and f, minimize the following objective function

$$\arg\min_{\mathbf{w}} \sum_{n=1}^{N} L(y_i, f(x_i; \mathbf{w})) + R(\mathbf{w})$$



Use Newton's method?

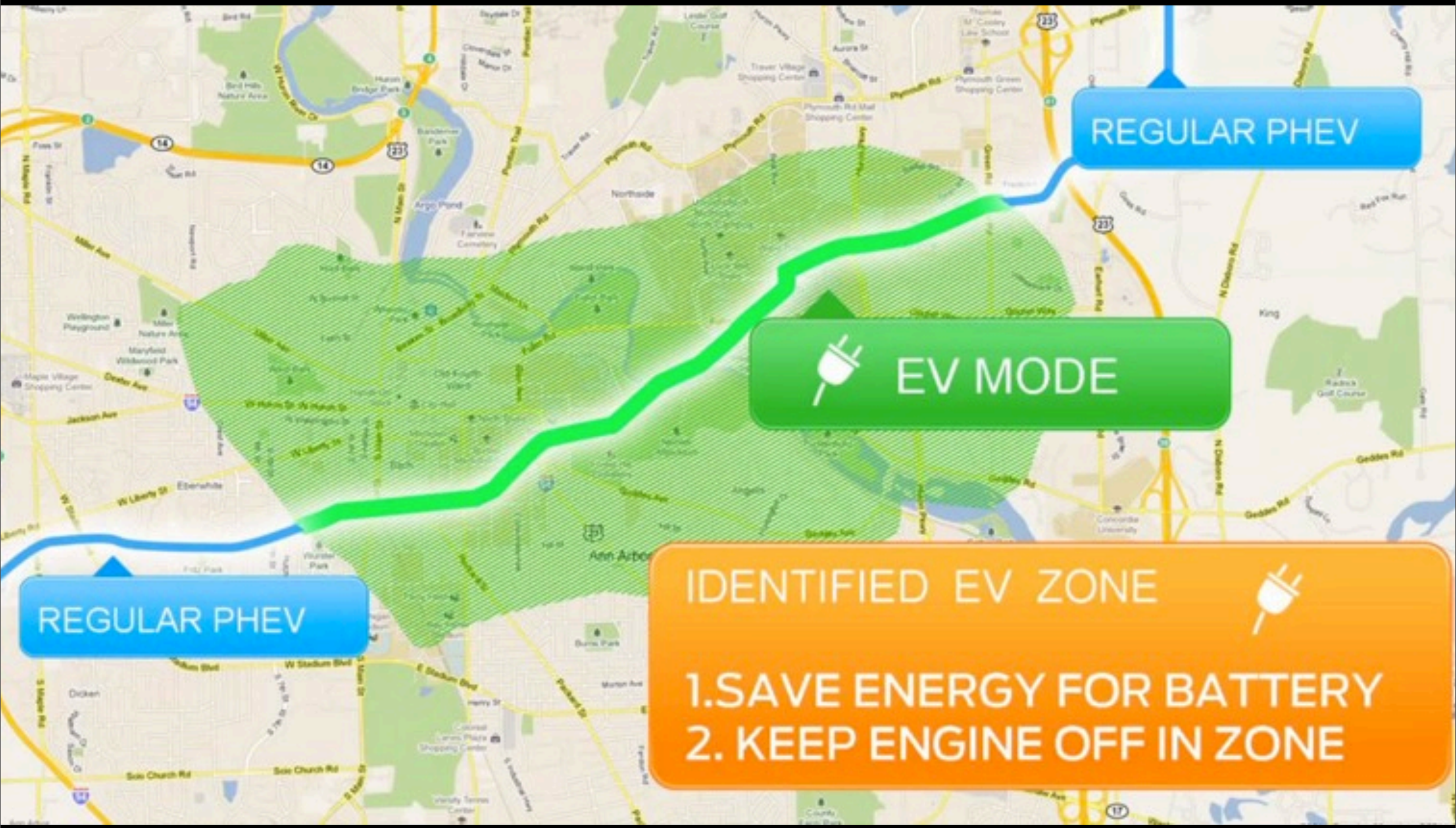
$$w^{t+1} \leftarrow w^t - \mathbf{H}(w^t)^{-1} \nabla J(w^t)$$

http://www.flickr.com/photos/visitfinland/5424369765/

Why Machine Learning in the Cloud?

http://www.flickr.com/photos/turtlemom_nancy/2046347762/sizes/l/in/photostream/


PRECONDITIONS:
AT WORK, TUESDAY, 5:00 PM
VEHICLE:
GOOD AFTERNOON RYAN, ARE YOU HEADED HOME?
RYAN:
YES
VEHICLE:
I SEE THERE IS A EV DRIVING ZONE ALONG YOUR ROUTE. WOULD YOU LIKE TO RESERVE ENERGY FOR IT?
RYAN:
YES, THAT WOULD BE GREAT
Ford


REGULAR PHEV
EV MODE
REGULAR PHEV
IDENTIFIED EV ZONE
1.SAVE ENERGY FOR BATTERY
2. KEEP ENGINE OFF IN ZONE

CLOUD-BASED POWERTRAIN OPTIMIZATION

DRIVING **HISTORY**

PREDICT **DESTINATION**

OPTIMIZE **POWERTRAIN**

Google Prediction API

Prediction API

Prediction API

- Machine learning as a web service

Prediction API

- Machine learning as a web service
- Use Google's machine learning algorithms

Prediction API

- Machine learning as a web service
- Use Google's machine learning algorithms
- Use Google's computing infrastructure

Prediction API

- Machine learning as a web service
- Use Google's machine learning algorithms
- Use Google's computing infrastructure
- Beta in 2010, generally available in 2011

1. Upload — Upload your training data to Google Storage

2. Train — Build a model from your data

3. Predict — Make new predictions

Step 1: Upload

Step 1: Upload

- Training data: inputs to outputs

Step 1: Upload

- Training data: inputs to outputs
- Data format: Comma-Separated Value

```
english, “to err is human, but to really ...”
spanish, “no hay mal que por bien no benga. “
```

Step 1: Upload

- Training data: inputs to outputs
- Data format: Comma-Separated Value

```
english, “to err is human, but to really ...”
spanish, “no hay mal que por bien no benga. “
```

- Upload to Google Storage

```
$ gsutil cp my_data gs://my/data
```

Step 2: Train

Step 2: Train

- To train a model
  `POST prediction/v1.2/training`

  ```
  {id: 'my/data'}
  ```

Step 2: Train

- To train a model
  `POST prediction/v1.2/training`

  `{id: 'my/data'}`

- To see if a training job is finished
  `GET prediction/v1.2/training/my%2Fdata`

Step 3: Predict

Step 3: Predict

- Make a prediction
  POST prediction/v1.2/training/my%2Fdata

  {input:
    {csvInstance: [“To be or not to be ...”]}}

- {outputLabel: “english”,
  outputMulti: [{“label”: “english”,
                “score”: 0.92},
               {“label”: “spanish”,
                “score”: 0.08}]}

Step 4: Adapt

- Stream new data to new model

```
PUT prediction/v1.2/training/my%2Fdata

{classLabel: “french”,
  csvInstance: [“J’aime X! C’est le meiller”]}
```

 Customer Sentiment	 Transaction Risk	 Species Identification	 Message Routing	 Diagnostics
 Churn Prediction	 Legal Docket Classification	 Suspicious Activity	 Work Roster Assignment	 Inappropriate Content
 Recommend Products	 Political Bias	 Uplift Marketing	 Email Filtering	 Career Counselling

- Sign up the Google Prediction API
  Google APIs Console
  http://code.google.com/apis/console

- More info about the Prediction API
  http://code.google.com/apis/predict

Scaling Up

Scaling Up

- Why big data?

Scaling Up

- Why big data?
- Parallelize machine learning algorithms

Scaling Up

- Why big data?
- Parallelize machine learning algorithms
  - Embarrassingly parallel

Scaling Up

- Why big data?
- Parallelize machine learning algorithms
  - Embarrassingly parallel
  - Parallelize sub-routines

Scaling Up

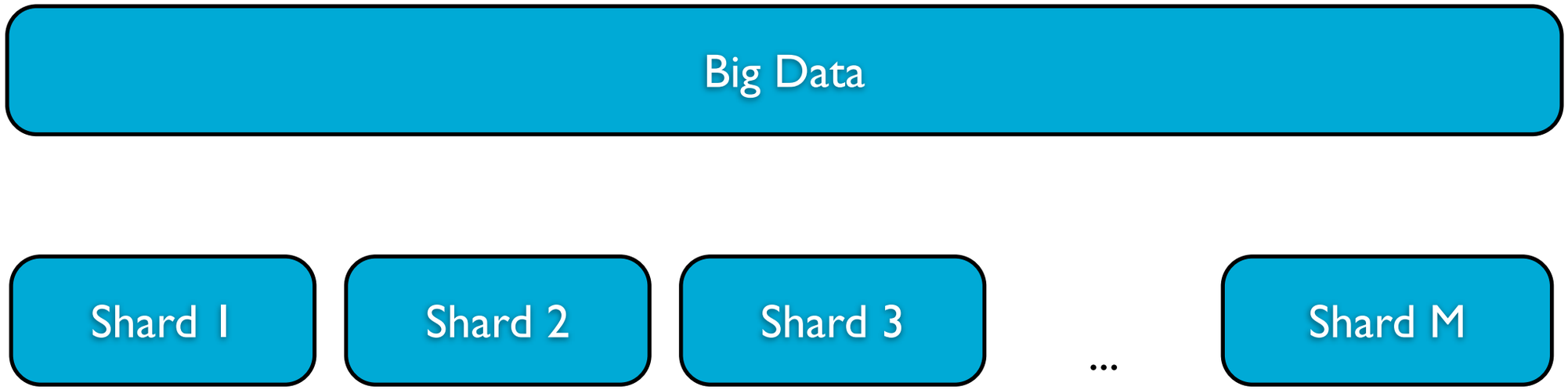
- Why big data?
- Parallelize machine learning algorithms
  - Embarrassingly parallel
  - Parallelize sub-routines
  - Distributed learning

Subsampling

Subsampling

Big Data

Subsampling

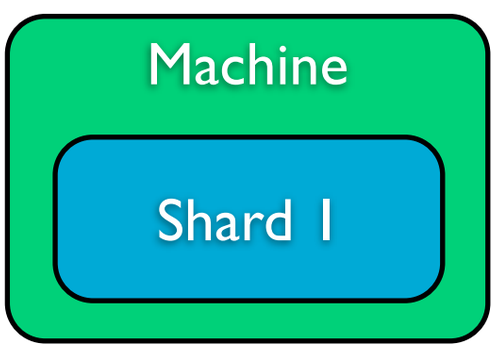


Subsampling

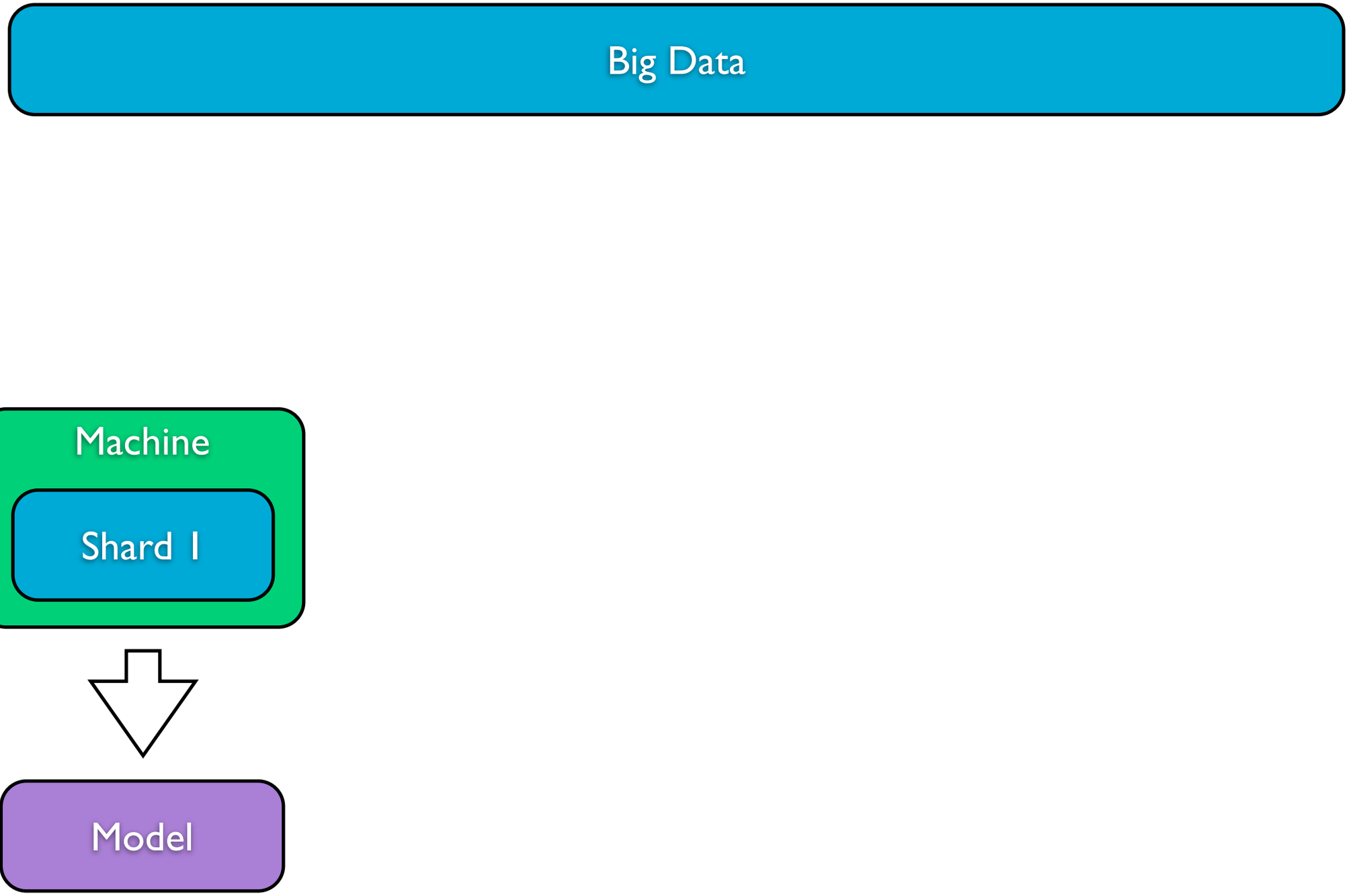
Subsampling

Subsampling

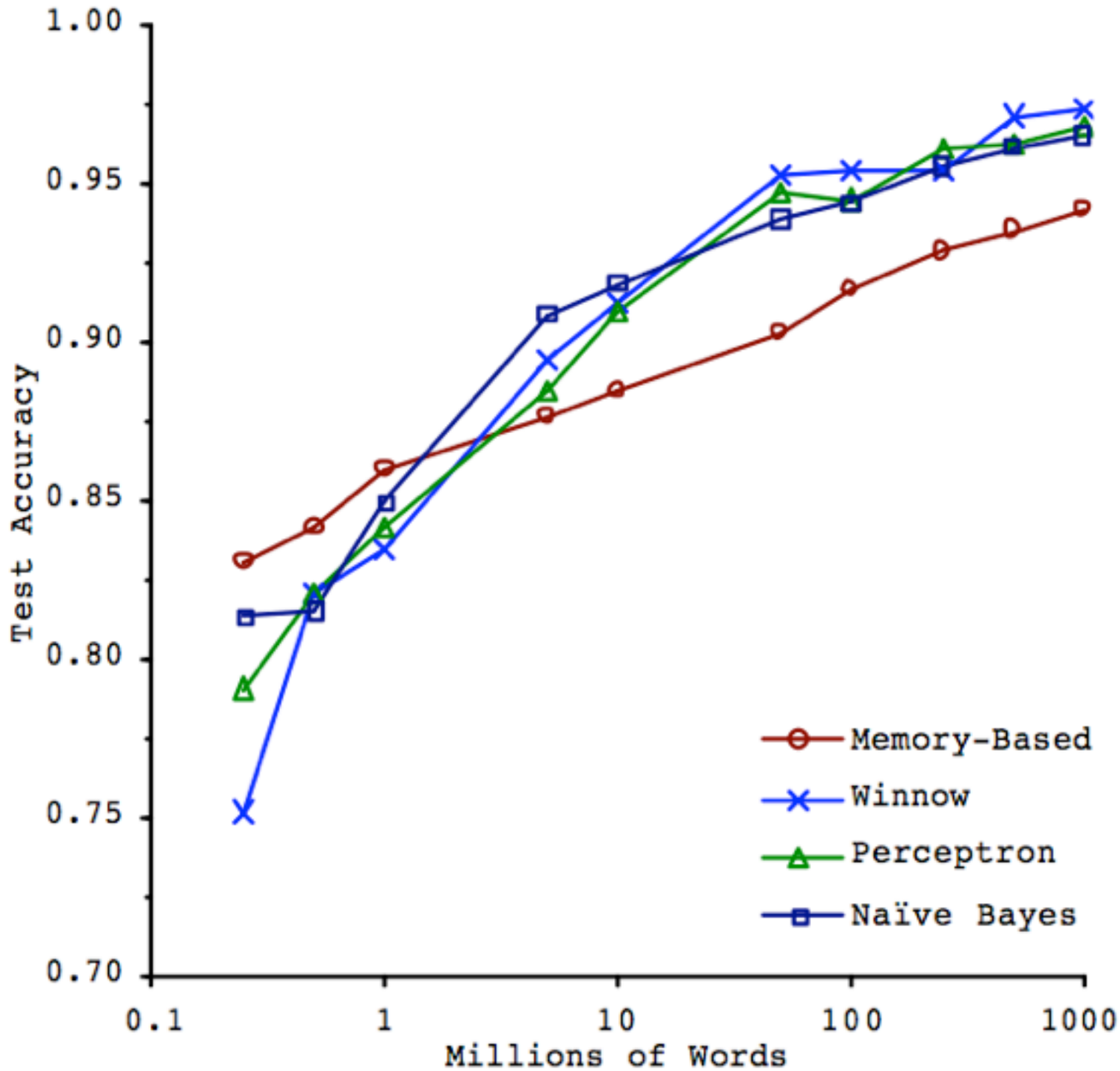
Reduce N



Subsampling



Why not Small Data?



[Banko and Brill, 2001]

Scaling Up

- Why big data?
- Parallelize machine learning algorithms
  - Embarrassingly parallel
  - Parallelize sub-routines
  - Distributed learning

Parallelize Optimization

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

Parallelize Optimization

- Maximum Entropy Classifiers

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

Parallelize Optimization

- Maximum Entropy Classifiers

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

Parallelize Optimization

- Maximum Entropy Classifiers

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

Parallelize Optimization

- Maximum Entropy Classifiers

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

- Good: J(w) is concave

Parallelize Optimization

- Maximum Entropy Classifiers

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

- Good: J(w) is concave
- Bad: no closed-form solution like NB

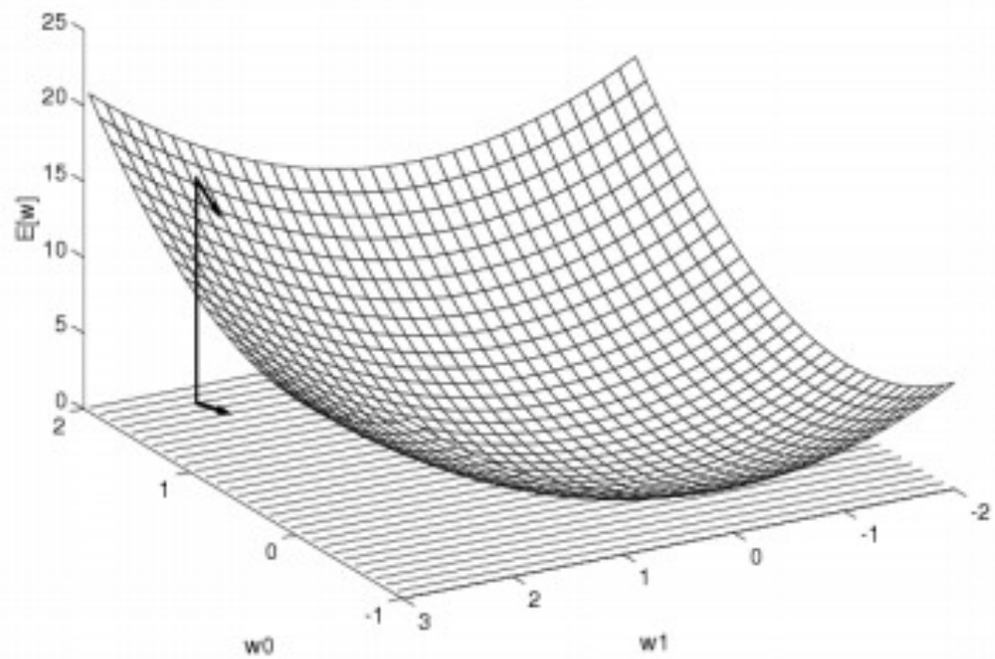
Parallelize Optimization

- Maximum Entropy Classifiers

$$\arg\min_{\mathbf{w}} \prod_{i=1}^{N} \frac{\exp(\sum_{p=1}^{P} w_p * x_p^i)^{y_i}}{1 + \exp(\sum_{p=1}^{P} w_p * x_p^i)}$$

- Good: J(w) is concave
- Bad: no closed-form solution like NB
- Ugly: Large N

Gradient Descent



Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

http://www.cs.cmu.edu/~epxing/Class/10701/Lecture/lecture7.pdf

Gradient Descent

Gradient Descent

- w is initialized as zero
- for t in 1 to T
  - Calculate gradients
  -

Gradient Descent

- w is initialized as zero
- for t in 1 to T
  - Calculate gradients $\nabla J(\mathbf{w})$
  -

Gradient Descent

- w is initialized as zero
- for t in 1 to T
  - Calculate gradients $\nabla J(\mathbf{w})$
  - $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^{t} - \eta \nabla J(\mathbf{w})$

Gradient Descent

- w is initialized as zero
- for t in 1 to T
  - Calculate gradients $\nabla J(\mathbf{w})$
  - $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^{t} - \eta \nabla J(\mathbf{w})$

$$\nabla J(w) = \sum_{i=1}^{N} P(\mathbf{w}, x_i, y_i)$$

Distribute Gradient

- w is initialized as zero
- for t in 1 to T
  - Calculate gradients in **parallel**

- Training CPU: O(TPN) to O(TPN / M)

Distribute Gradient

- w is initialized as zero
- for t in 1 to T
  - Calculate gradients in **parallel**

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^{t} - \eta \nabla J(\mathbf{w})$$

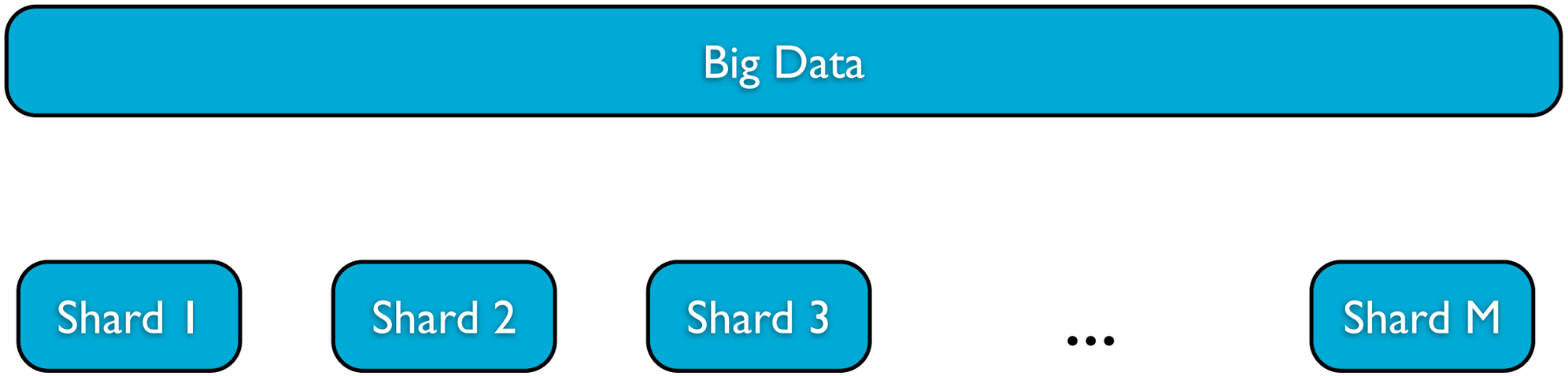
- Training CPU: O(TPN) to O(TPN / M)

Distribute Gradient

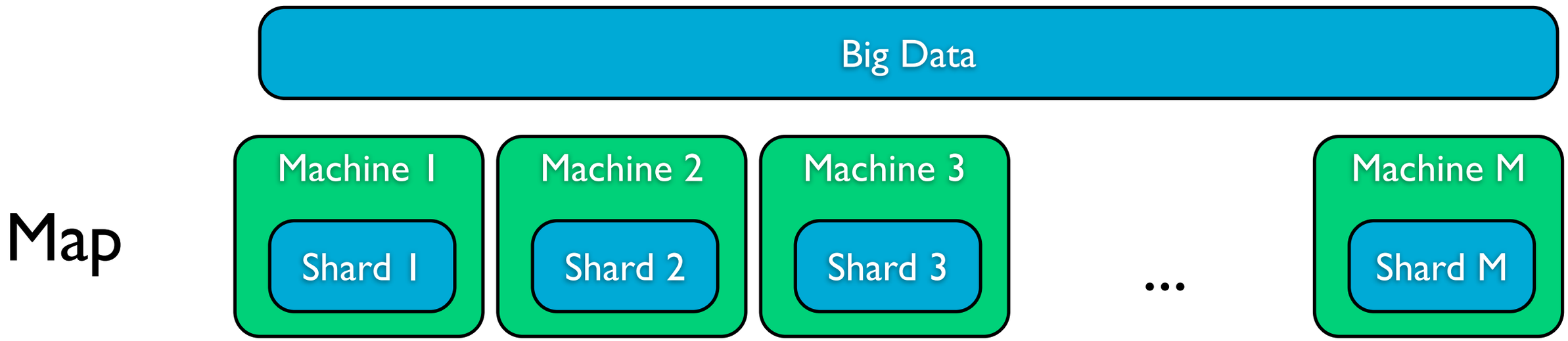
Distribute Gradient

Big Data

Distribute Gradient

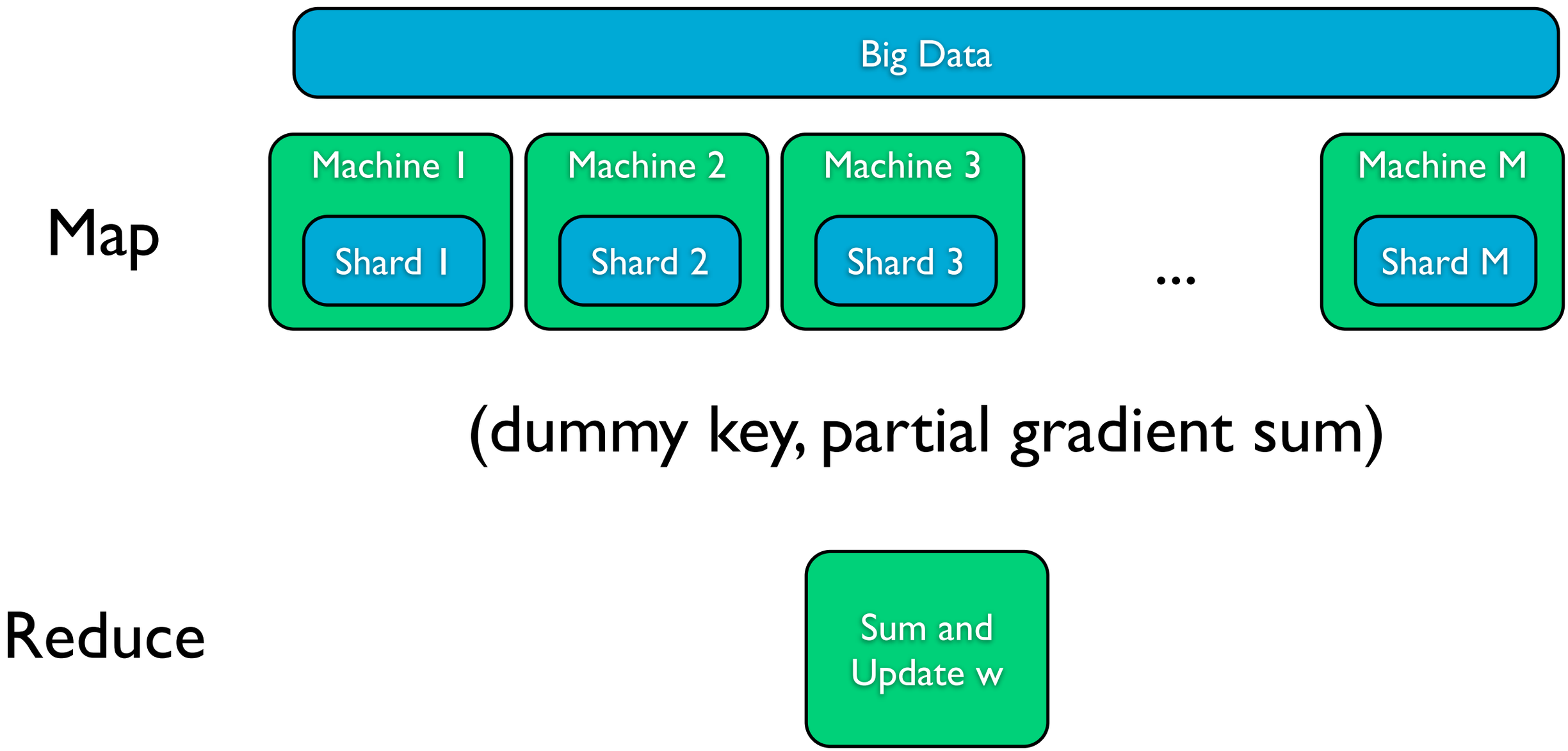


Distribute Gradient

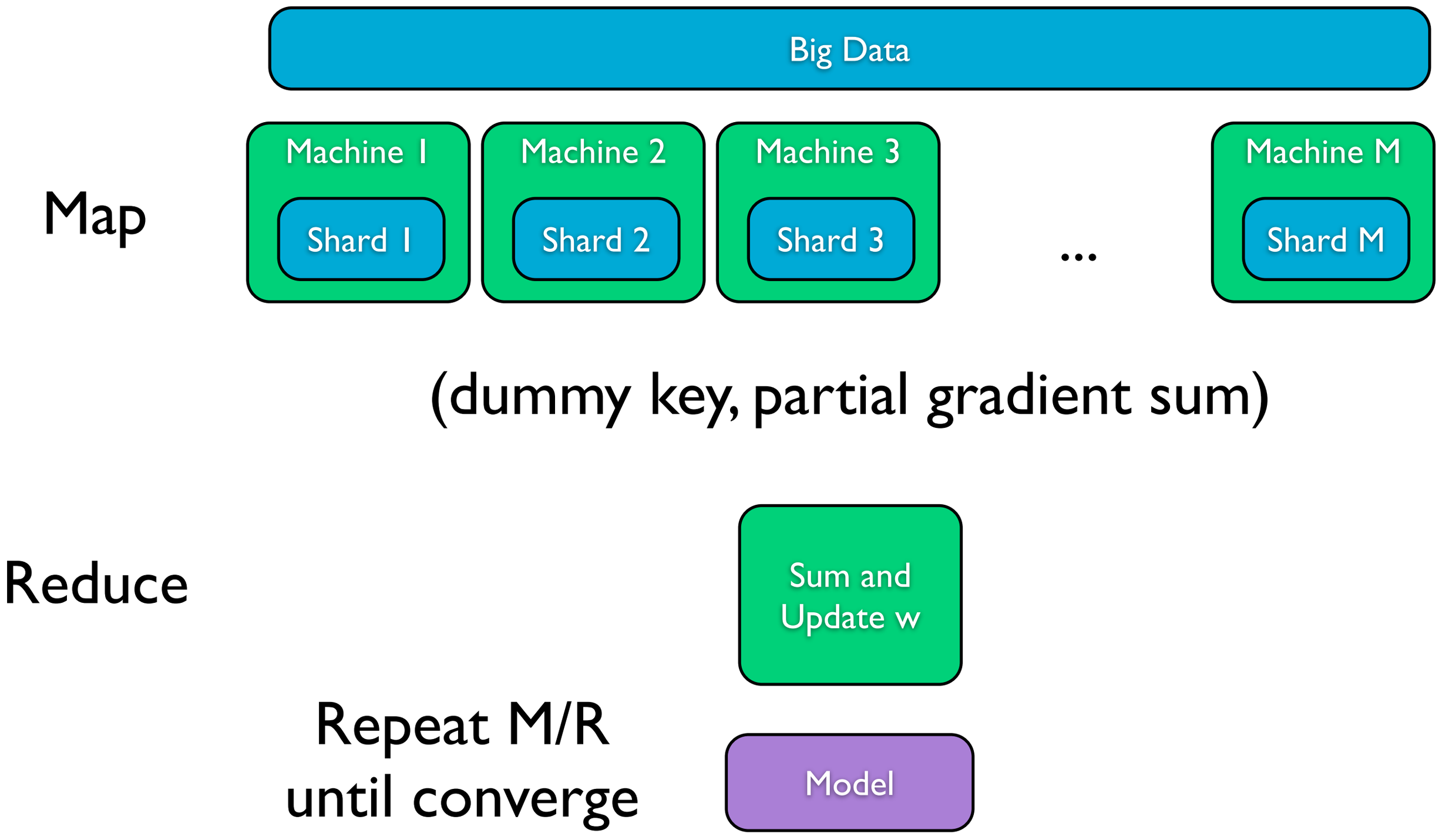


(dummy key, partial gradient sum)

Distribute Gradient



Distribute Gradient



Scaling Up

- Why big data?
- Parallelize machine learning algorithms
  - Embarrassingly parallel
  - Parallelize sub-routines
  - Distributed learning

Parallelize Subroutines

- Support Vector Machines

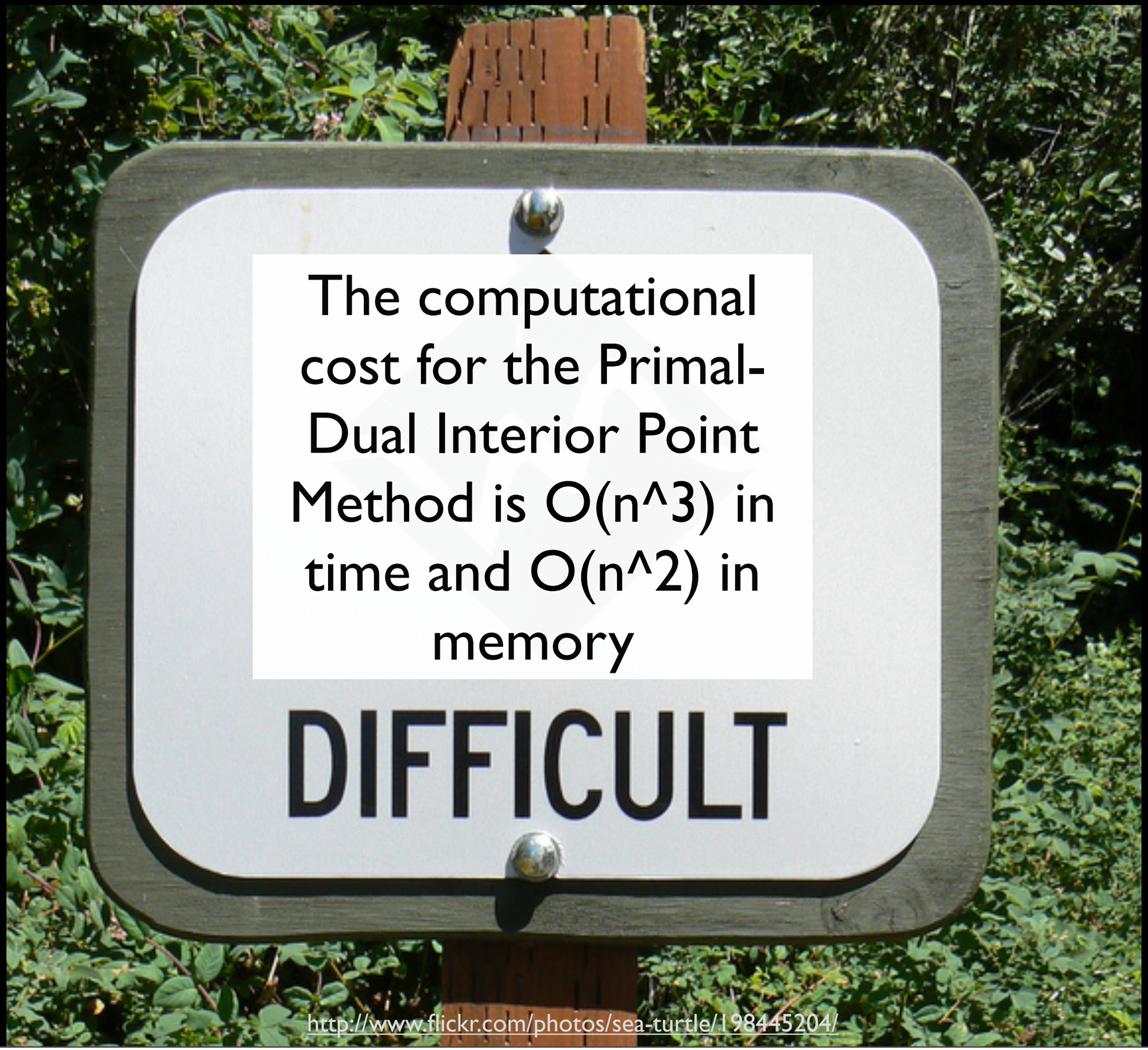
$$\arg\min_{\mathbf{w},b,\zeta} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{n}\zeta_i$$

$$\text{s.t.} \quad 1 - y_i(\mathbf{w}\cdot\phi(x_i) + b) \leq \zeta_i, \zeta_i \geq 0$$

- Solve the dual problem

$$\arg\min_{\alpha} \frac{1}{2}\alpha^T\mathbf{Q}\alpha - \alpha^T\mathbf{1}$$

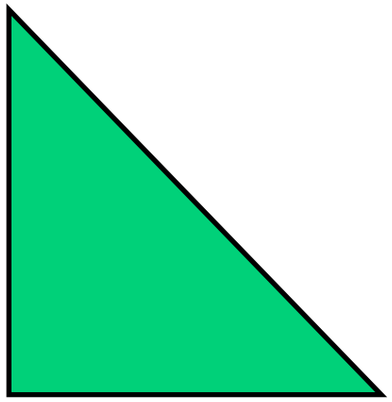
$$\text{s.t.} \quad \mathbf{0} \leq \alpha \leq \mathbf{C}, \mathbf{y}^T\alpha = 0$$



The computational cost for the Primal-Dual Interior Point Method is O(n^3) in time and O(n^2) in memory

http://www.flickr.com/photos/sea-turtle/198445204/

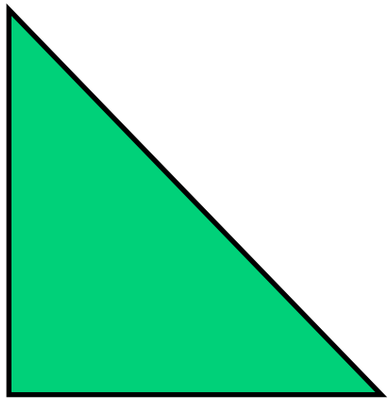
Parallel SVM [Chang et al, 2007]



$$\sqrt{N}$$

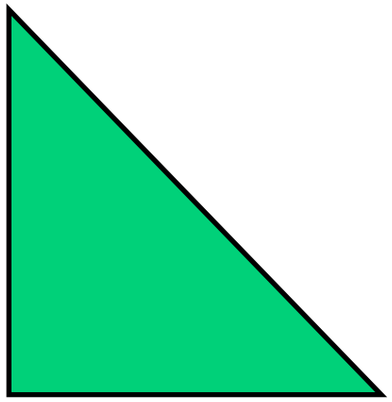
Parallel SVM [Chang et al, 2007]

- Parallel, row-wise incomplete Cholesky Factorization for Q



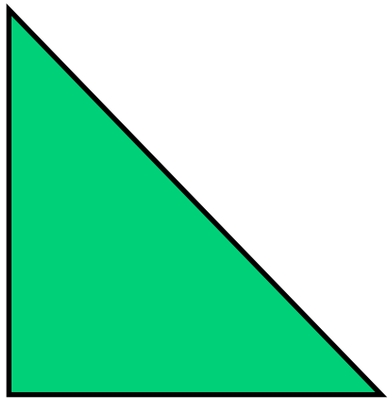
$$\sqrt{N}$$

Parallel SVM [Chang et al, 2007]

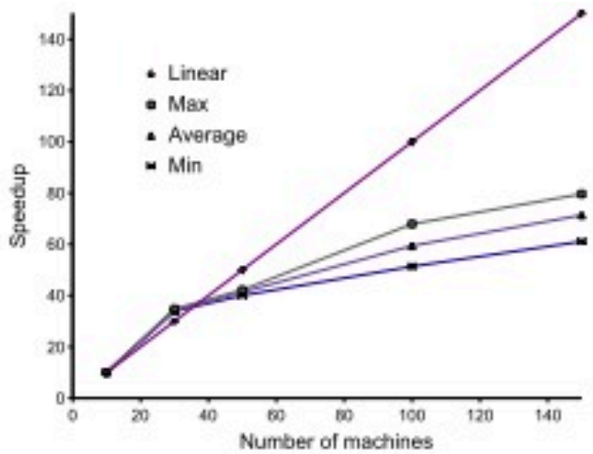


- Parallel, row-wise incomplete Cholesky Factorization for Q
- Parallel interior point method
  - Time O(n^3) becomes O(n^2 / M)
  - Memory O(n^2) becomes O(n $\sqrt{N}$ / M)

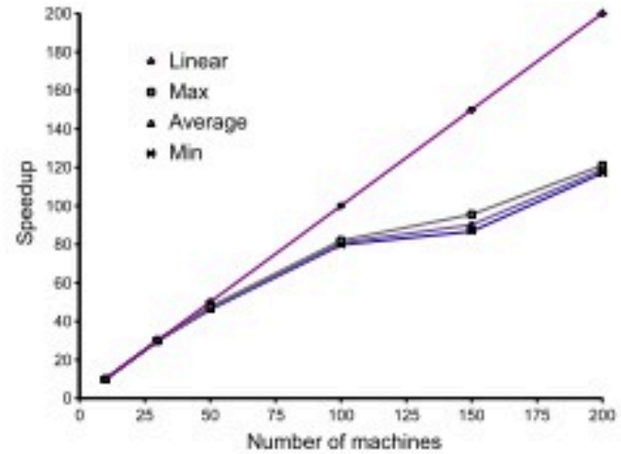
Parallel SVM [Chang et al, 2007]



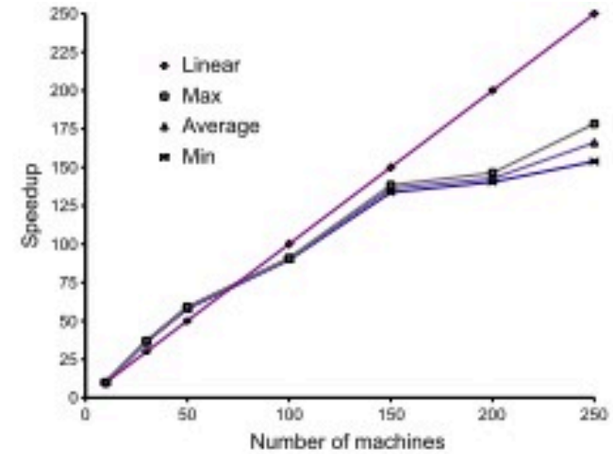
- Parallel, row-wise incomplete Cholesky Factorization for Q
- Parallel interior point method
  - Time O(n^3) becomes O(n^2 / M)
  - Memory O(n^2) becomes O(n $\sqrt{N}$ / M)
- Parallel Support Vector Machines (psvm) http://code.google.com/p/psvm/
  - Implement in MPI



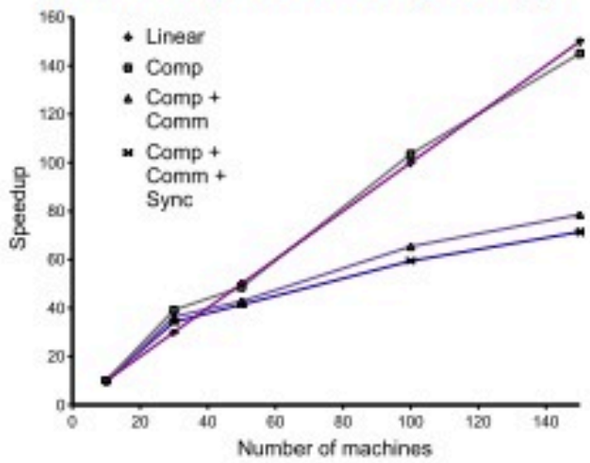
(a) Image (200k) speedup



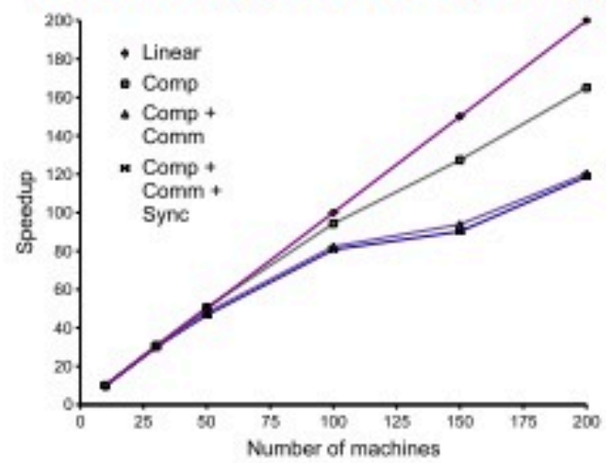
(b) Covertype (500k) speedup



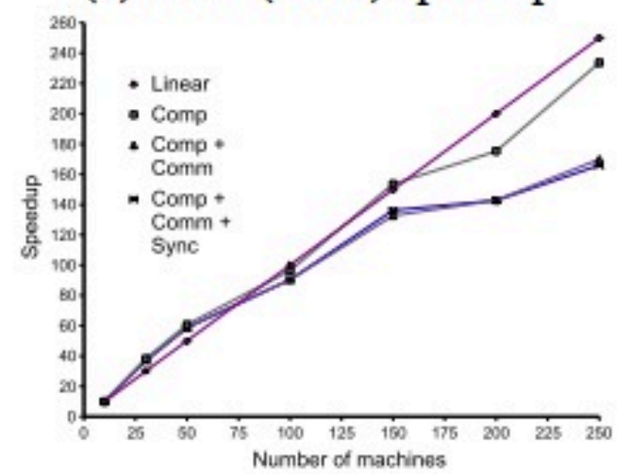
(c) RCV (800k) speedup



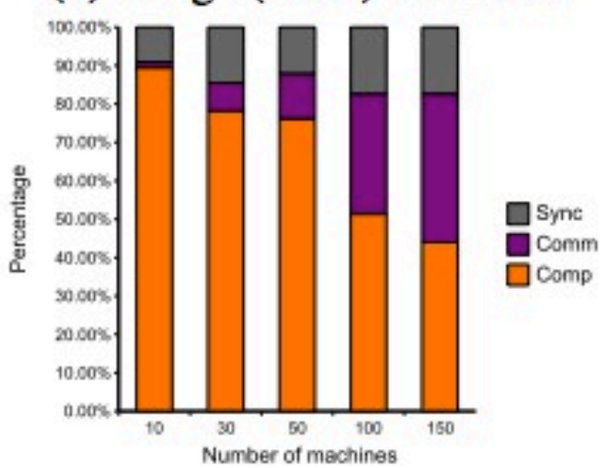
(d) Image (200k) overhead



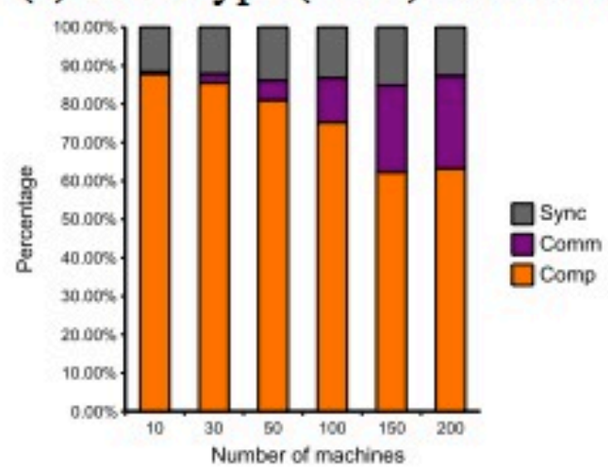
(e) Covertype (500k) overhead



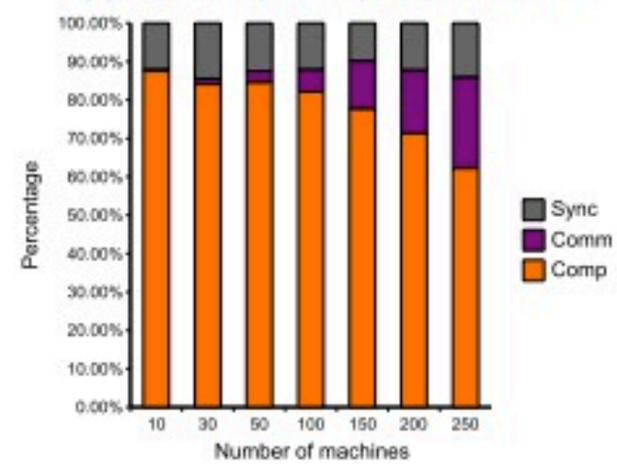
(f) RCV (800k) overhead



(g) Image (200k) fraction



(h) Covertype (500k) fraction



(i) RCV (800k) fraction

Scaling Up

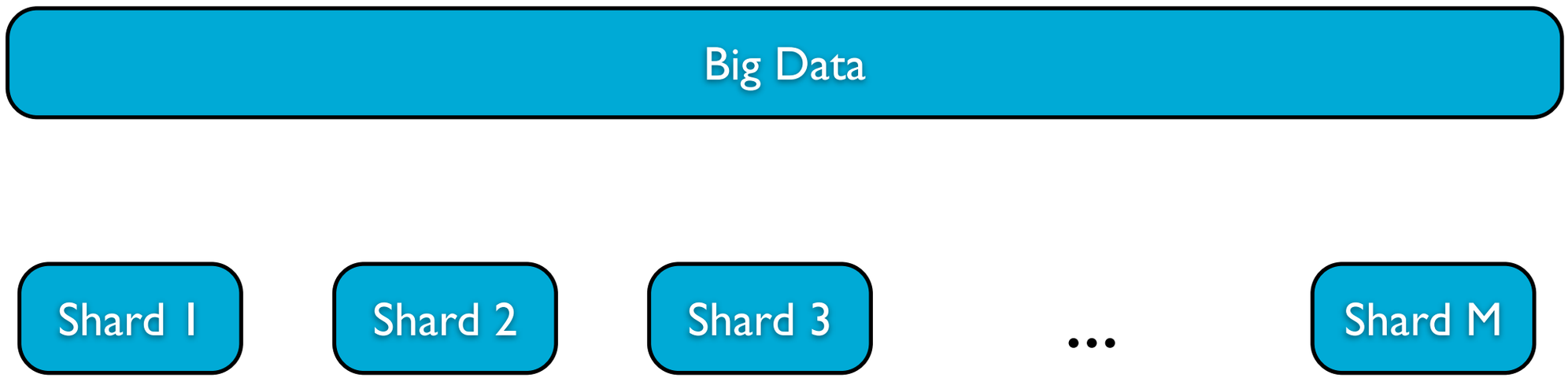
- Why big data?
- Parallelize machine learning algorithms
  - Embarrassingly parallel
  - Parallelize sub-routines
  - Distributed learning

Majority Vote

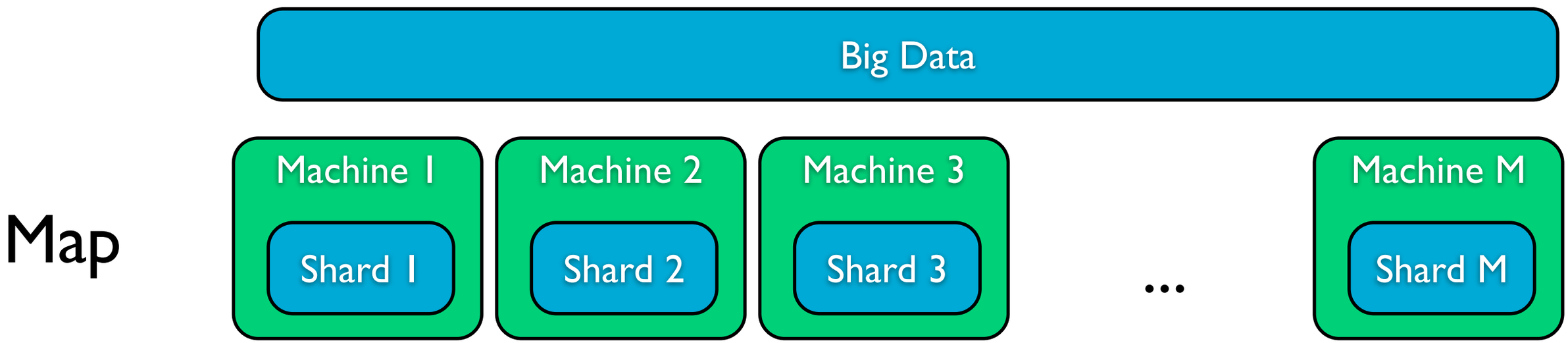
Majority Vote

Big Data

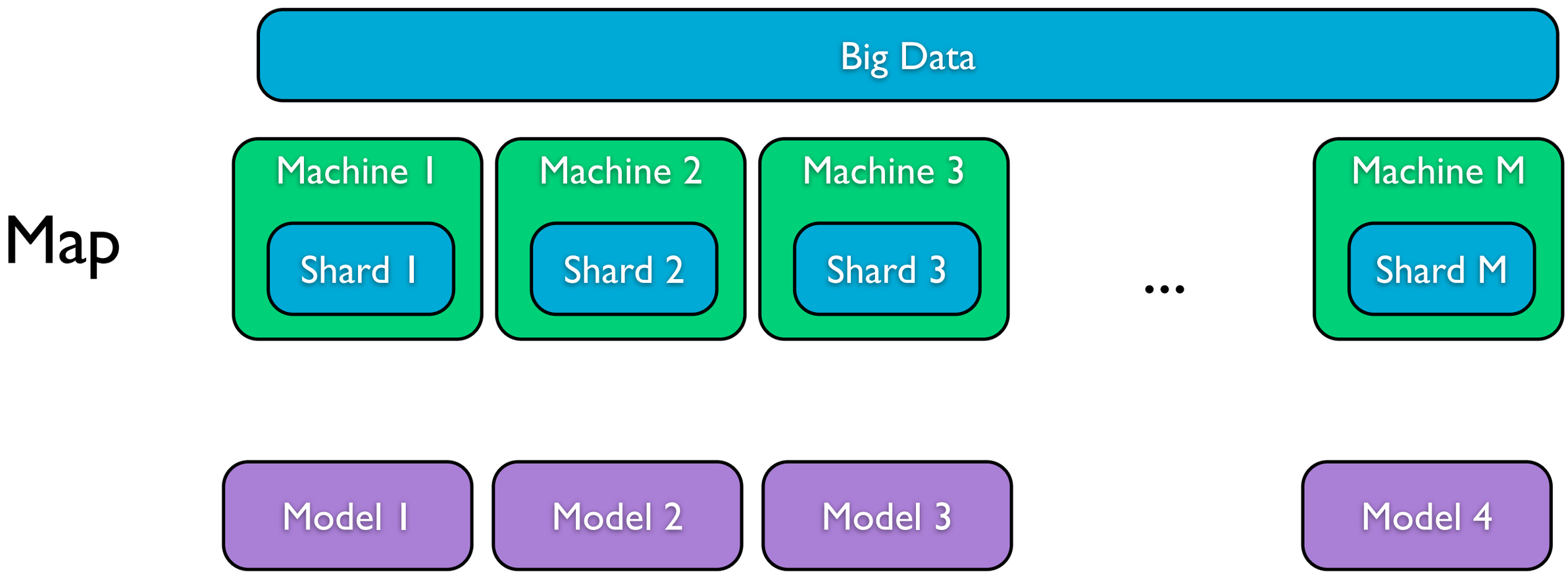
Majority Vote



Majority Vote



Majority Vote



Majority Vote

- Train individual classifiers independently
- Predict by taking majority votes
- Training CPU: O(TPN) to O(TPN / M)

Parameter Mixture

[Mann et al, 2009]

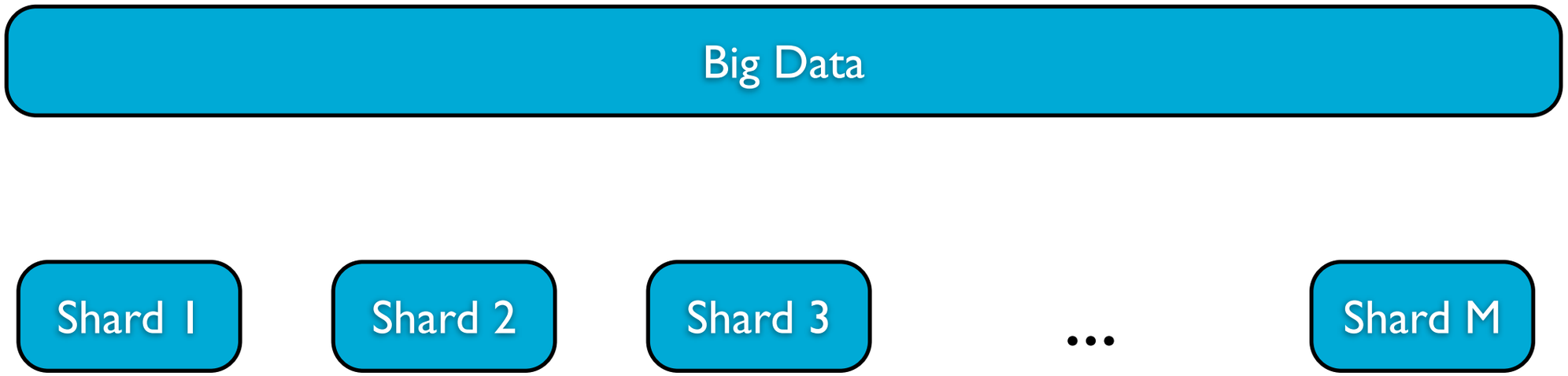
Parameter Mixture

[Mann et al, 2009]

Big Data

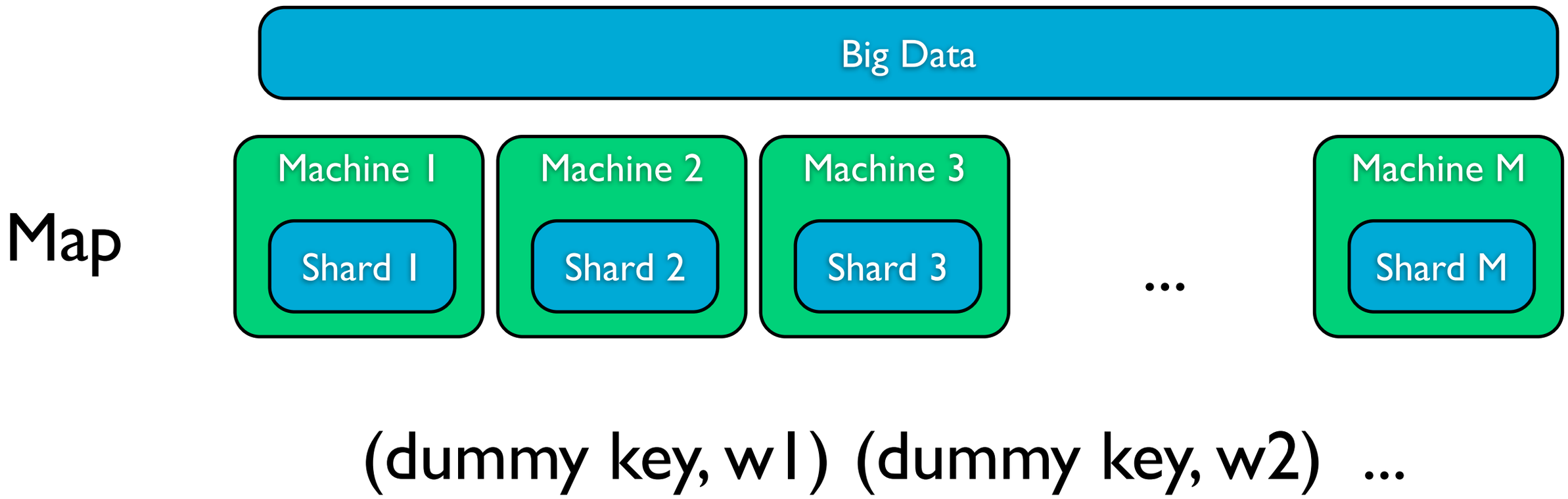
Parameter Mixture

[Mann et al, 2009]



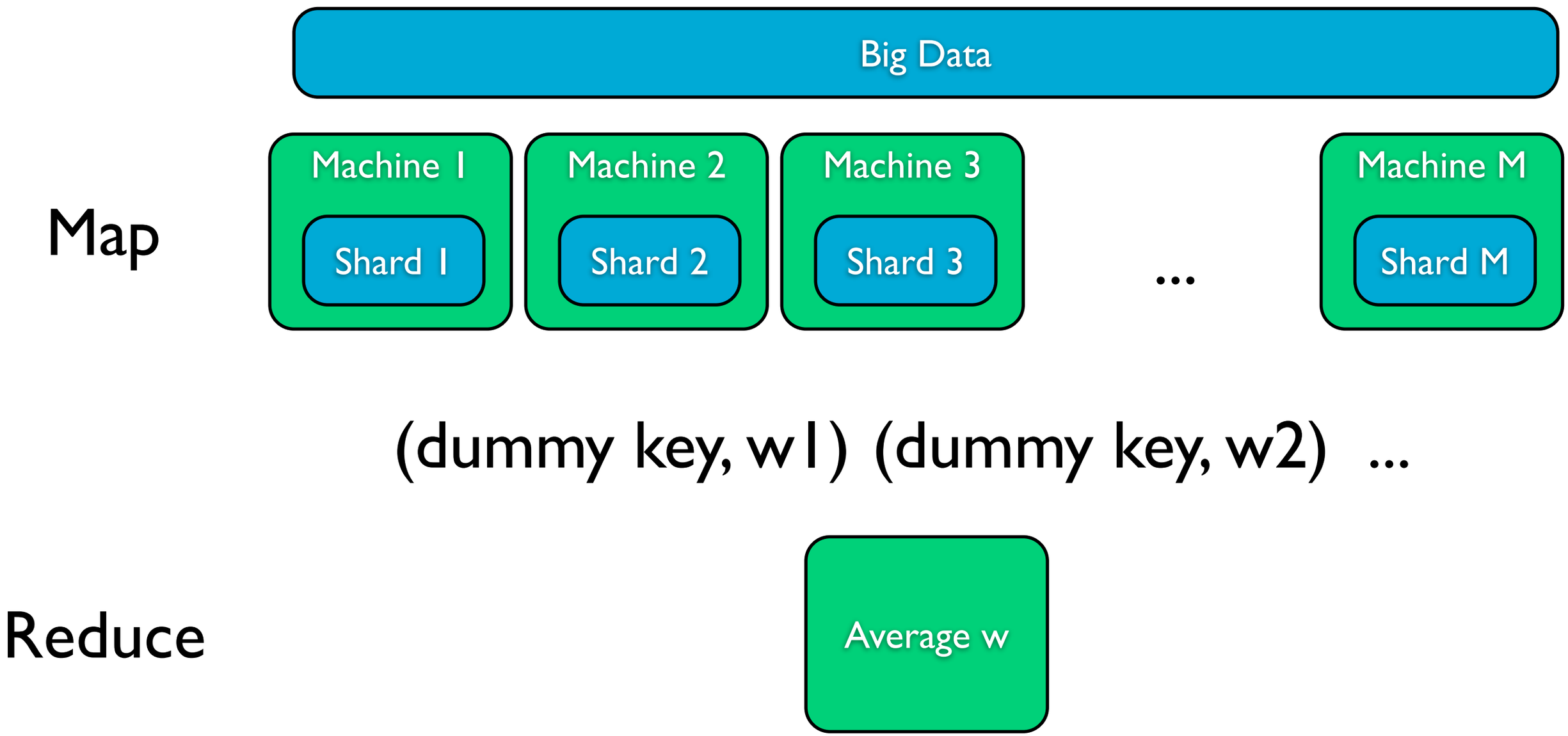
Parameter Mixture

[Mann et al, 2009]



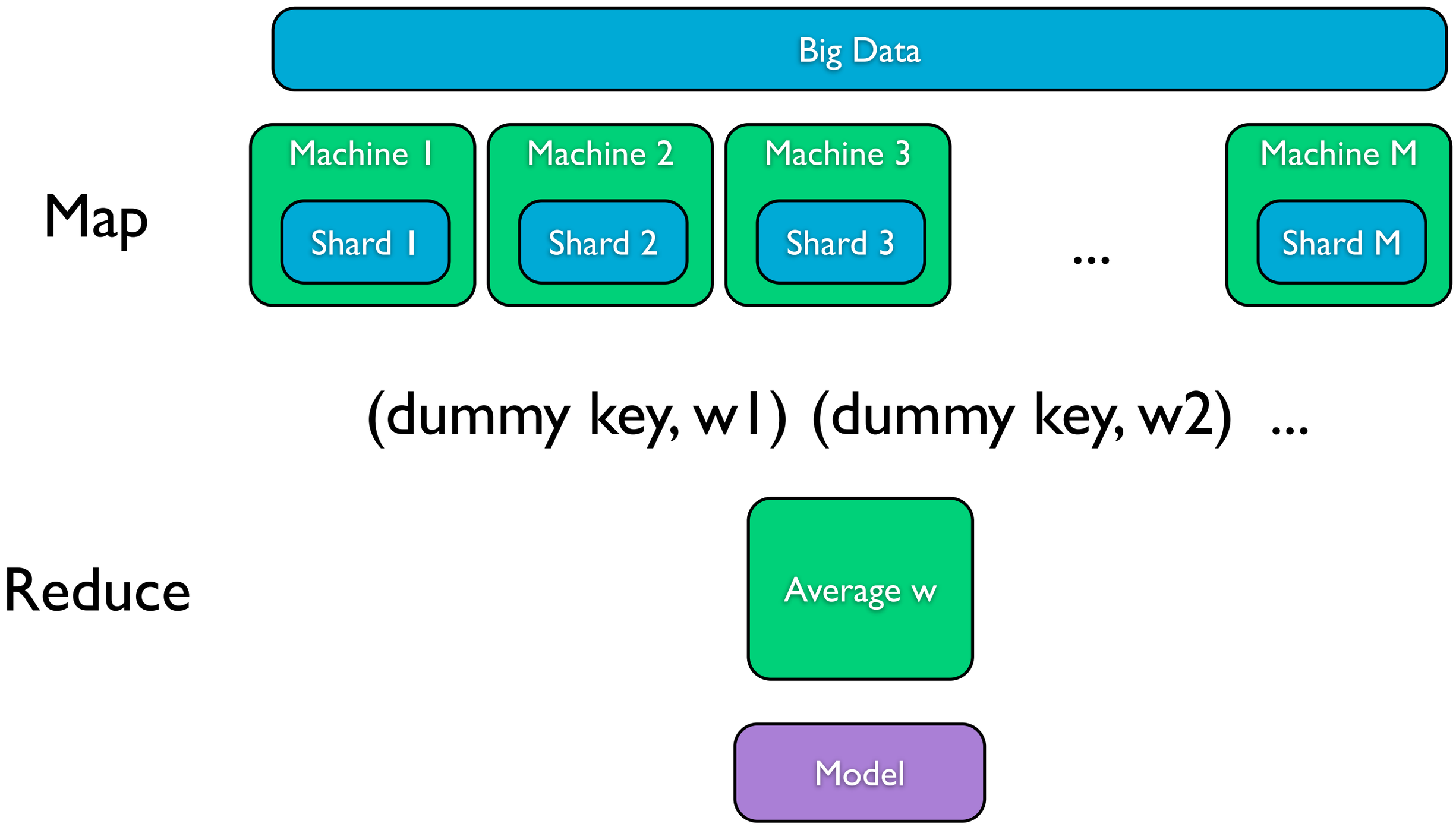
Parameter Mixture

[Mann et al, 2009]



Parameter Mixture

[Mann et al, 2009]



tp://www.flickr.com/photos/annamatic3000/127945652/

Much Less network usage than distributed gradient descent

O(MN) vs. O(MNT)

	Training Method	Accuracy	Wall Clock	Cumulative CPU	Network Usage
English POS	Distributed Gradient	97.60%	17.5 m	11.0 h	652 GB
(m=100k,p=10)	Majority Vote	96.80%	12.5 m	18.5 h	0.686 GB
	Mixture Weight	96.80%	5 m	11.5 h	0.015 GB
Sentiment	Distributed Gradient	81.18%	104 m	123 h	367 GB
(m=900k,p=10)	Majority Vote	81.25%	131 m	168 h	3 GB
	Mixture Weight	81.30%	110 m	163 h	9 GB
RCV1-v2	Distributed Gradient	27.03%	48 m	407 h	479 GB
(m=2.6M,p=10)	Majority Vote	26.89%	54 m	474 h	3 GB
	Mixture Weight	27.15%	56 m	473 h	0.108 GB
Speech	Distributed Gradient	34.95%	160 m	511 h	200 GB
(m=100k,p=499)	Mixture Weight	34.99%	130 m	534 h	158 GB
Deja	Distributed Gradient	64.74%	327 m	733 h	5,283 GB
(m=1.5M,p=200)	Mixture Weight	65.46%	316 m	707 h	48 GB
Deja 250K	Distributed Gradient	67.03%	340 m	698 h	17,428 GB
(m=1.5M,p=200)	Mixture Weight	66.86%	300 m	710 h	65 GB
Gigaword	Distributed Gradient	51.16%	240 m	18,598 h	13,000 GB
(m=1M,p=1k)	Mixture Weight	50.12%	215 m	17,998 h	21 GB

	Training Method	Accuracy	Wall Clock	Cumulative CPU	Network Usage
English POS	Distributed Gradient	97.60%	17.5 m	11.0 h	652 GB
(m=100k,p=10)	Majority Vote	96.80%	12.5 m	18.5 h	0.686 GB
	Mixture Weight	96.80%	5 m	11.5 h	0.015 GB
Sentiment	Distributed Gradient	81.18%	104 m	123 h	367 GB
(m=900k,p=10)	Majority Vote	81.25%	131 m	168 h	3 GB
	Mixture Weight	81.30%	110 m	163 h	9 GB
RCV1-v2	Distributed Gradient	27.03%	48 m	407 h	479 GB
(m=2.6M,p=10)	Majority Vote	26.89%	54 m	474 h	3 GB
	Mixture Weight	27.15%	56 m	473 h	0.108 GB
Speech	Distributed Gradient	34.95%	160 m	511 h	200 GB
(m=100k,p=499)	Mixture Weight	34.99%	130 m	534 h	158 GB
Deja	Distributed Gradient	64.74%	327 m	733 h	5,283 GB
(m=1.5M,p=200)	Mixture Weight	65.46%	316 m	707 h	48 GB
Deja 250K	Distributed Gradient	67.03%	340 m	698 h	17,428 GB
(m=1.5M,p=200)	Mixture Weight	66.86%	300 m	710 h	65 GB
Gigaword	Distributed Gradient	51.16%	240 m	18,598 h	13,000 GB
(m=1M,p=1k)	Mixture Weight	50.12%	215 m	17,998 h	21 GB

	Training Method	Accuracy	Wall Clock	Cumulative CPU	Network Usage
English POS	Distributed Gradient	97.60%	17.5 m	11.0 h	652 GB
(m=100k,p=10)	Majority Vote	96.80%	12.5 m	18.5 h	0.686 GB
	Mixture Weight	96.80%	5 m	11.5 h	0.015 GB
Sentiment	Distributed Gradient	81.18%	104 m	123 h	367 GB
(m=900k,p=10)	Majority Vote	81.25%	131 m	168 h	3 GB
	Mixture Weight	81.30%	110 m	163 h	9 GB
RCV1-v2	Distributed Gradient	27.03%	48 m	407 h	479 GB
(m=2.6M,p=10)	Majority Vote	26.89%	54 m	474 h	3 GB
	Mixture Weight	27.15%	56 m	473 h	0.108 GB
Speech	Distributed Gradient	34.95%	160 m	511 h	200 GB
(m=100k,p=499)	Mixture Weight	34.99%	130 m	534 h	158 GB
Deja	Distributed Gradient	64.74%	327 m	733 h	5,283 GB
(m=1.5M,p=200)	Mixture Weight	65.46%	316 m	707 h	48 GB
Deja 250K	Distributed Gradient	67.03%	340 m	698 h	17,428 GB
(m=1.5M,p=200)	Mixture Weight	66.86%	300 m	710 h	65 GB
Gigaword	Distributed Gradient	51.16%	240 m	18,598 h	13,000 GB
(m=1M,p=1k)	Mixture Weight	50.12%	215 m	17,998 h	21 GB

Iterative Param Mixture

[McDonald et al., 2010]

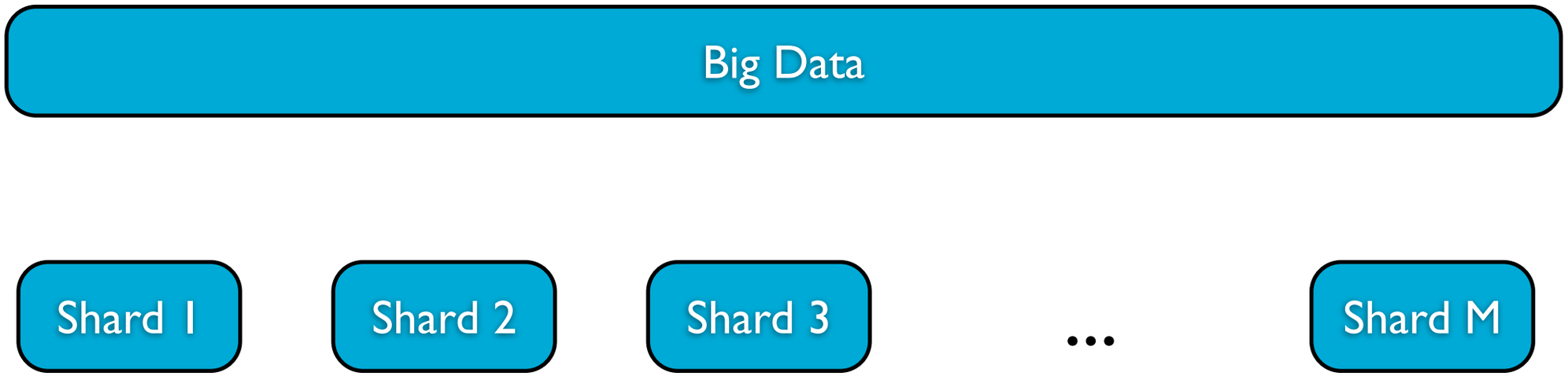
Iterative Param Mixture

[McDonald et al., 2010]

Big Data

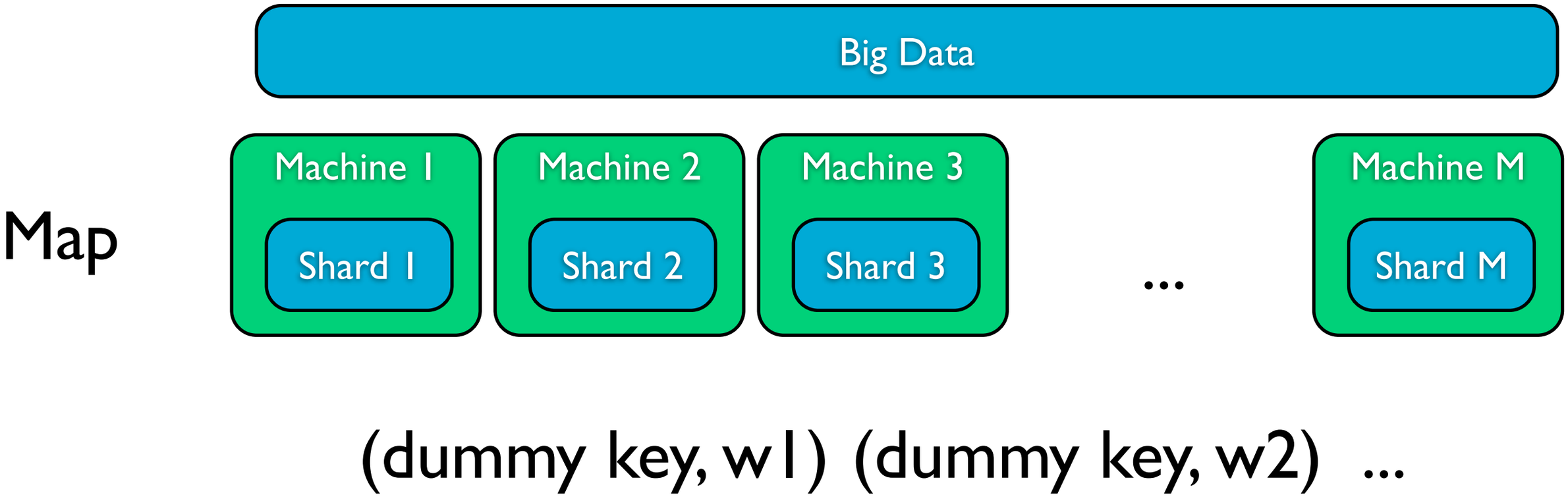
Iterative Param Mixture

[McDonald et al., 2010]



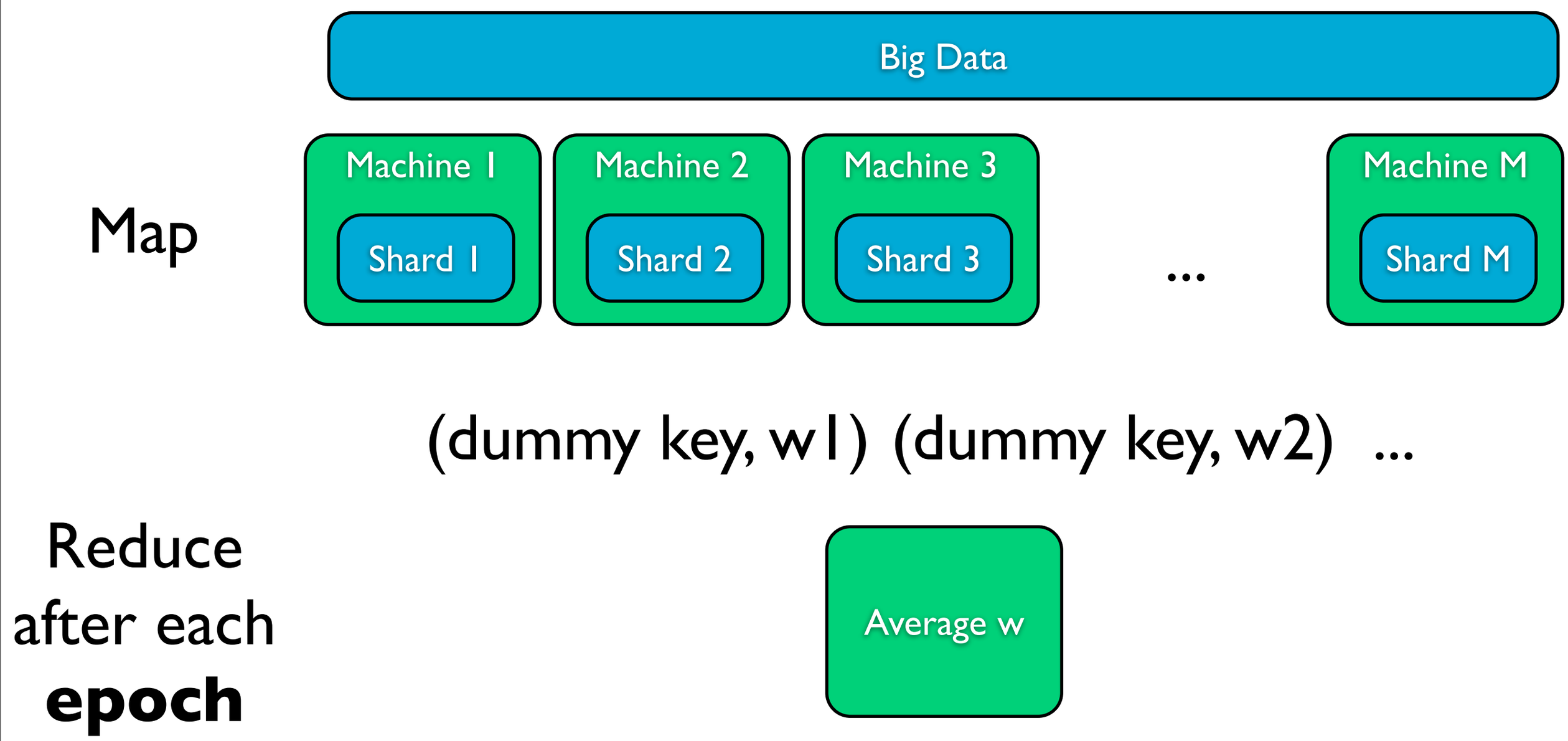
Iterative Param Mixture

[McDonald et al., 2010]



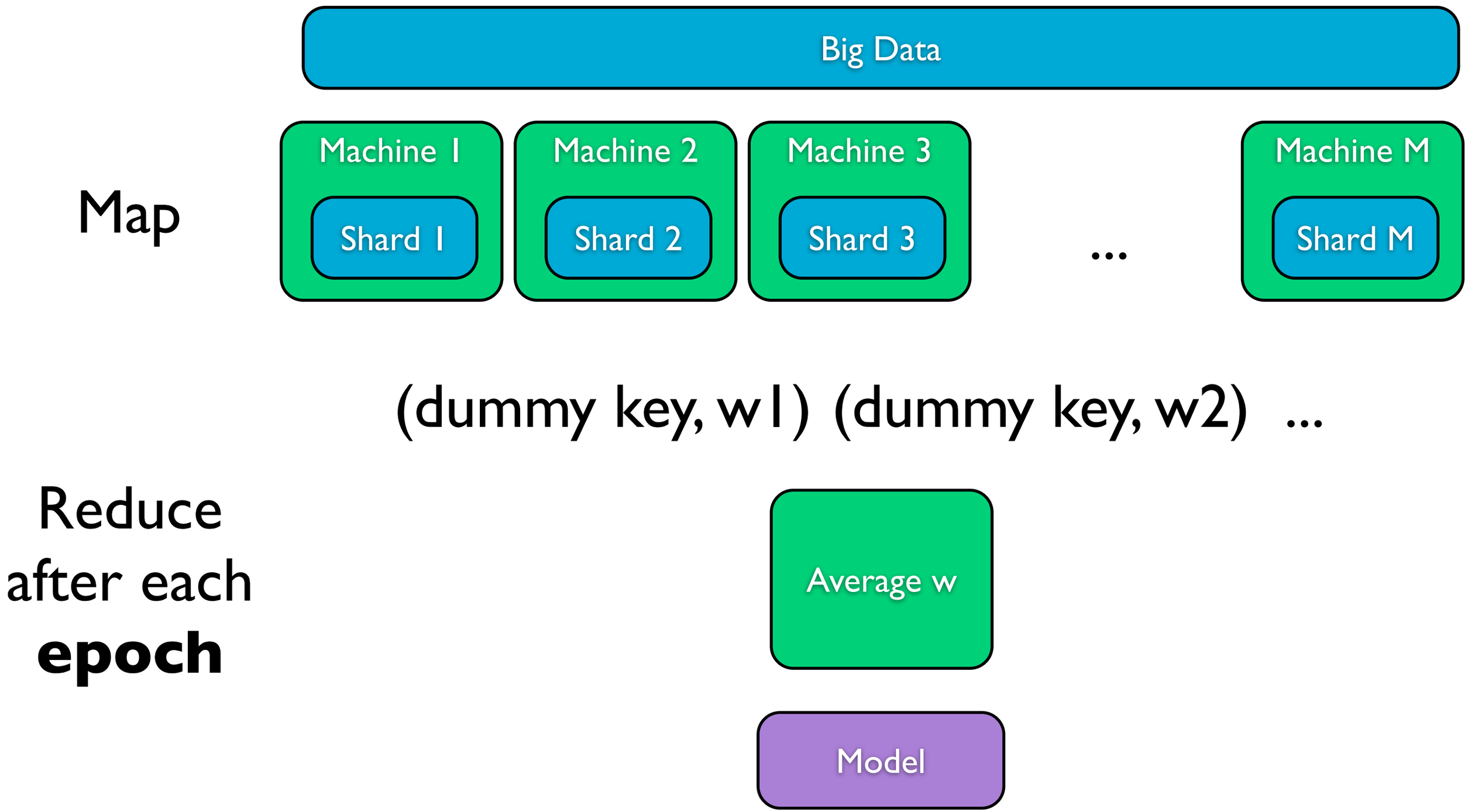
Iterative Param Mixture

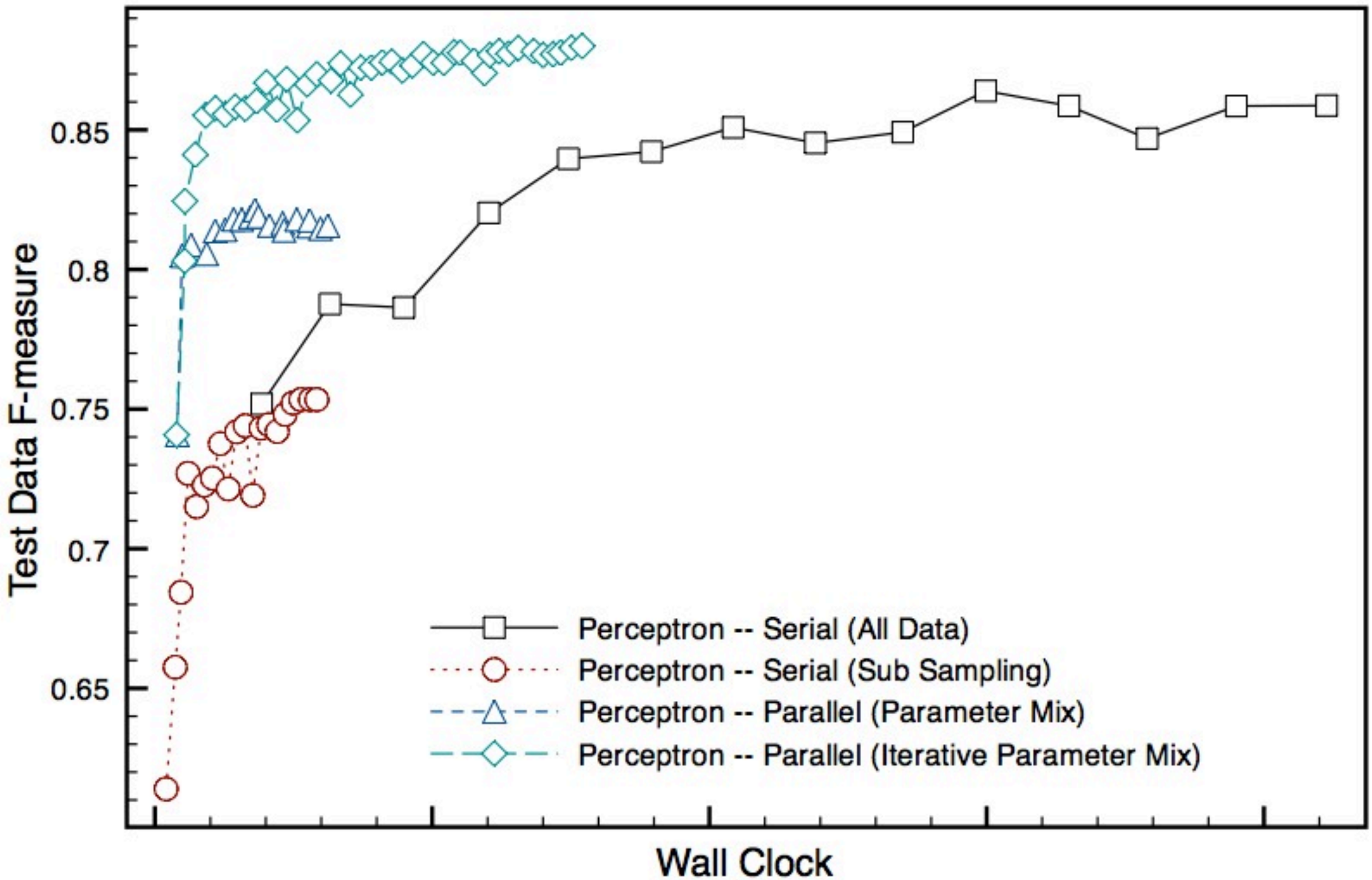
[McDonald et al., 2010]

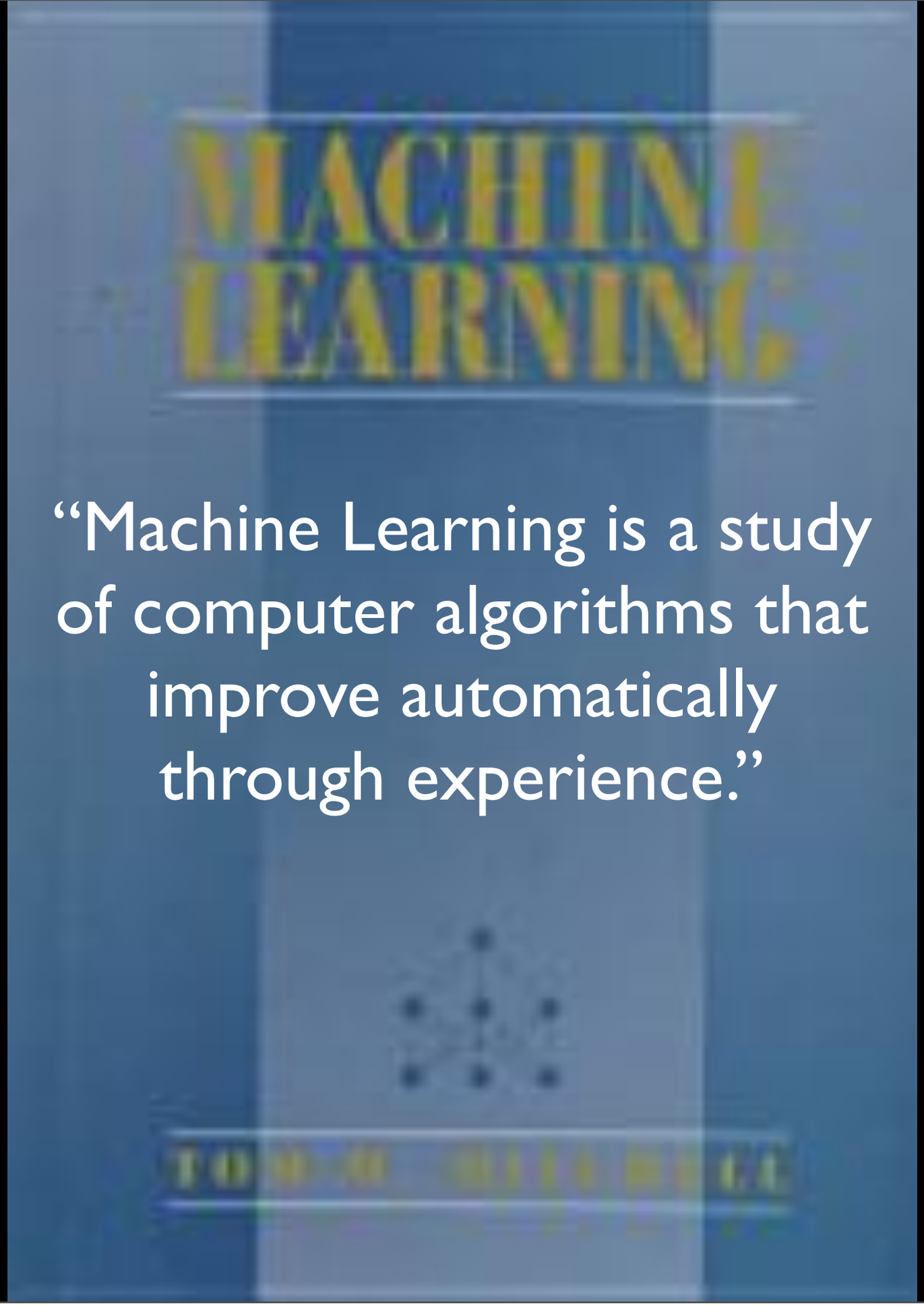


Iterative Param Mixture

[McDonald et al., 2010]




Test Data F-measure
0.85
0.8
0.75
0.7
0.65
Wall Clock
Perceptron -- Serial (All Data)
Perceptron -- Serial (Sub Sampling)
Perceptron -- Parallel (Parameter Mix)
Perceptron -- Parallel (Iterative Parameter Mix)



"Machine Learning is a study of computer algorithms that improve automatically through experience."

Google Prediction API

Machine Learning as a Web Service

Parallelize ML Algorithms

Parallelize ML Algorithms

- Embarrassingly parallel

Parallelize ML Algorithms

- Embarrassingly parallel
- Parallelize sub-routines

Parallelize ML Algorithms

- Embarrassingly parallel
- Parallelize sub-routines
- Distributed learning

Google APIs

Google APIs

- Prediction API
  - machine learning service on the cloud
  - http://code.google.com/apis/predict

Google APIs

- Prediction API
  - machine learning service on the cloud
  - http://code.google.com/apis/predict

- BigQuery
  - interactive analysis of massive data on the cloud
  - http://code.google.com/apis/bigquery