Machine Learning at Scale

James G. Shanahan[2]

Assistants: Jason Anastasopoulos[2] Liang Dai[1],

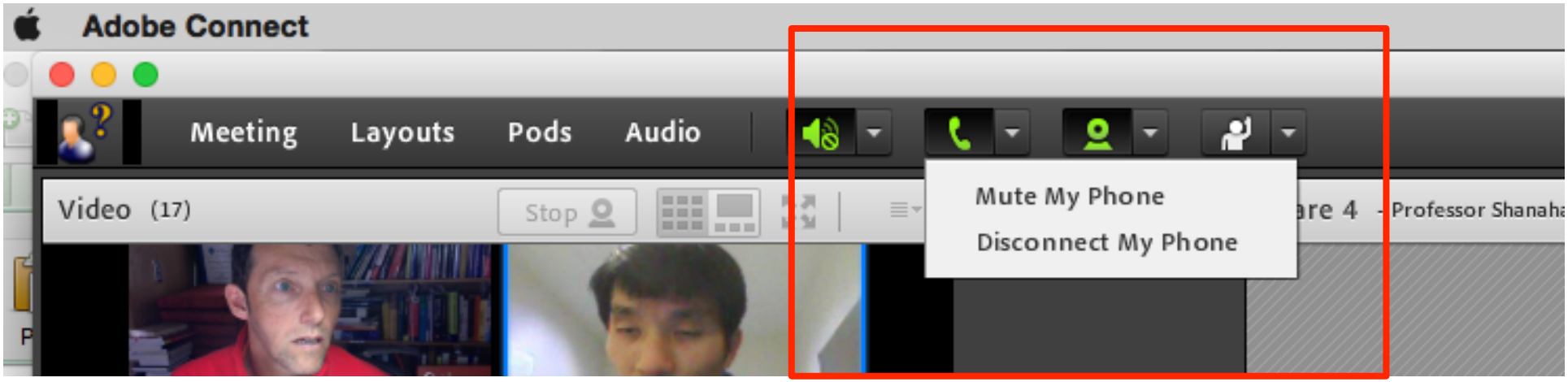
[1]NativeX, [2]iSchool UC Berkeley, CA, [3]UC Santa Cruz

EMAIL: James_DOT_Shanahan_AT_gmail_DOT_com

Live Session #2

January 19, 2016

Mute your mikes via Adobe Connect



Live Sessions Spring 2016

- Monday UCB-MIDS 1 W261 James Shanahan 7:00 PM - 8:30 PM EST
- **Tuesday UCB-MIDS 4 W261 Jason/James Shanahan 7:00 PM - 8:30 PM EST**
- **Wednesday UCB-MIDS 2 W261 James Shanahan 7:00 PM - 8:30 PM EST**
- **Wednesday UCB-MIDS 3 W261 James Shanahan 9:30 PM - 11:00 PM EST**

Starting a Live Session

1. **Start & Connect to the Audio**

2. **Start Your Webcam & Enable Webcam for Participants**

3. **Start Recording**

4. **Allow Participants Into the Room**

5. **Merge Speaker Audio (if necessary)**

This is also available on Page 3 of the “Live Session Essentials” document that you downloaded during Training Session 1.

Live sessions

- 80 minutes: directed discussion by instructors
- 10 minutes free talk

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
- **HW1: review and grading**
- **Q&A (WK02)**
- **Naive Bayes**
  - Various Naïve Bayes Flavours
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

- **Review HW1 together in Class**
- **Talk about Peer Grading Process**

HW1 Master Solution: group 4

- https://www.dropbox.com/s/q6moe8kp6suc955/JRW_hw1.ipynb?dl=0

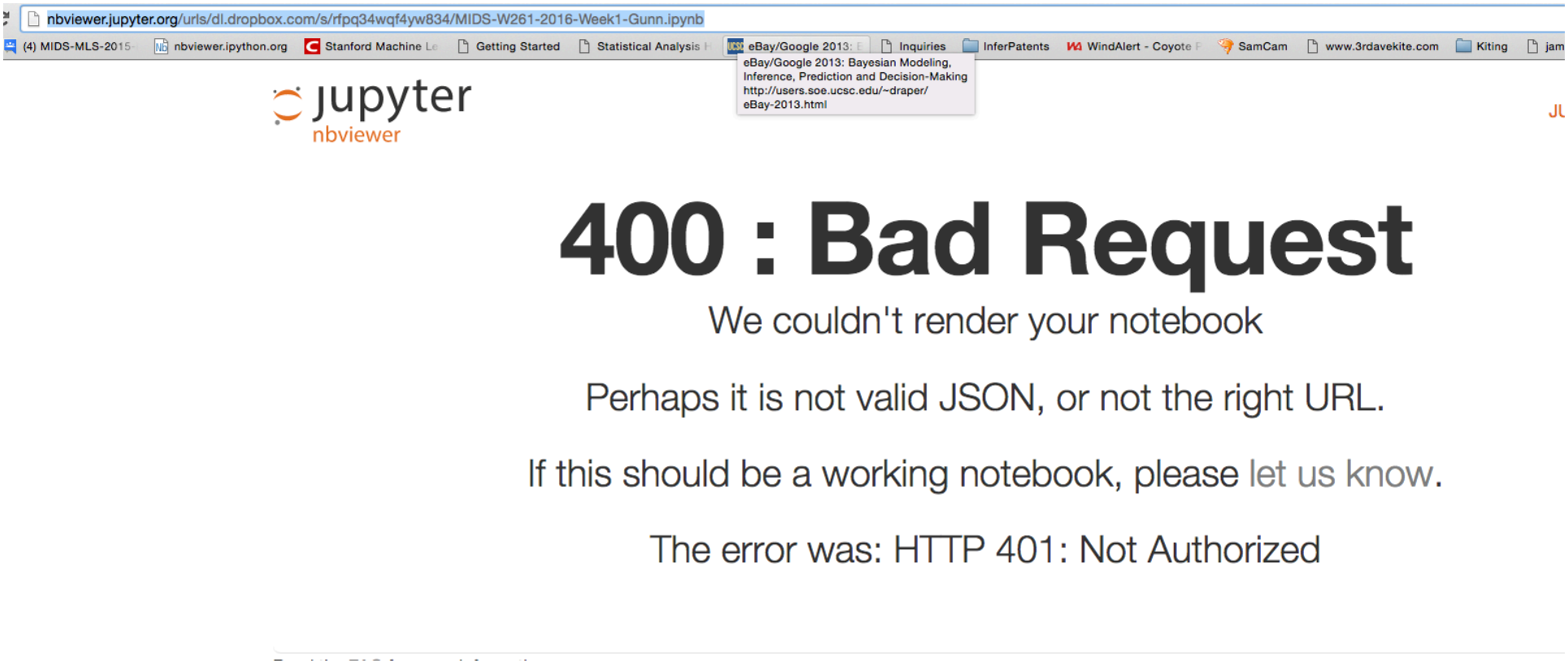
- http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/q6moe8kp6suc955/JRW_hw1.ipynb

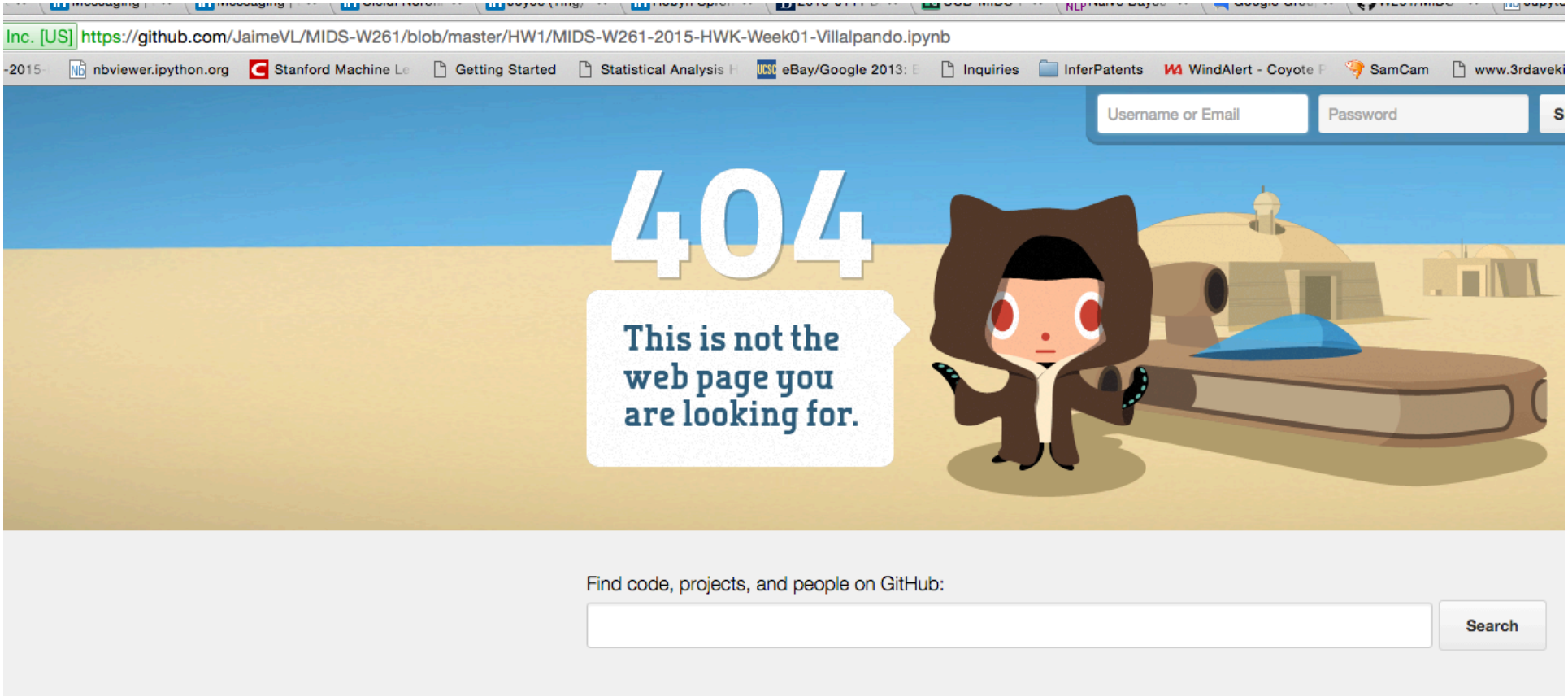
HW1: Group2

- http://nbviewer.jupyter.org/github/patng323/w261/blob/submitted/assignments/wk1/MIDS-W261-2015-HWK-Week01-Ng.ipynb

Group 3

- http://nbviewer.jupyter.org/github/jxu87/W261/blob/master/HW1/MIDS-W261-2015-HWK-Week01-Xu.ipynb


nbviewer.jupyter.org/urls/dl.dropbox.com/s/rfpq34wqf4yw834/MIDS-W261-2016-Week1-Gunn.ipynb
(4) MIDS-MLS-2015
nbviewer.ipython.org
Stanford Machine Le
Getting Started
Statistical Analysis H
eBay/Google 2013: E
Inquiries
InferPatents
WindAlert - Coyote P
SamCam
www.3rdavekite.com
Kiting
jam
eBay/Google 2013: Bayesian Modeling, Inference, Prediction and Decision-Making
http://users.soe.ucsc.edu/~draper/eBay-2013.html
jupyter
nbviewer
JU
400 : Bad Request
We couldn't render your notebook
Perhaps it is not valid JSON, or not the right URL.
If this should be a working notebook, please let us know.
The error was: HTTP 401: Not Authorized


Inc. [US] https://github.com/JaimeVL/MIDS-W261/blob/master/HW1/MIDS-W261-2015-HWK-Week01-Villalpando.ipynb
nbviewer.ipython.org
Stanford Machine Le
Getting Started
Statistical Analysis H
eBay/Google 2013: E
Inquiries
InferPatents
WindAlert - Coyote F
SamCam
www.3rdaveki
Username or Email
Password
404
This is not the web page you are looking for.
Find code, projects, and people on GitHub:
Search

Peer grading

- **Details of peer grading to follow in a separate email.**
- **Please feel free to revise HW1 and resubmit by 8AM tomorrow morning via this form:**
  - https://docs.google.com/forms/d/1ZOr9RnIe_A06AcZDB6K1mJN4vrLeSmS2PD6Xm3eOiis/viewform?usp=send_form

Submit homework HW1

- **Please feel free to revise HW1 and resubmit by 8AM Friday Morning (January 21, 2016) via this form:**
  - https://docs.google.com/forms/d/1ZOr9RnIe_A06AcZDB6K1mJN4vrLeSmS2PD6Xm3eOiis/viewform?usp=send_form

MIDS Machine Learning at Scale: Homework Submission Form

https://docs.google.com/forms/d/1ZOr9RnIe_A06AcZDB6K1mJN4vrLeSmS2PD6Xm3eOiis/viewform?usp=send_form

Select your Name from this list
If your name is not in this list please alert your instructor

NBviewer Link to Notebook

Link to Notebook (PDF Note timestamp will be checked)

Homework for Week
Select a week: Select HW1 for homework for week 1

Students names in this group (CSV style, firstName1 lastName1, firstName2 lastName2, ...)
Required for GROUP homeworks only

Emails (CSV style, e.g., tom@blah.com, jill@blah.com)
Required for GROUP homeworks only

Submit

Never submit passwords through Google Forms.

Big data Definition: use

Definition

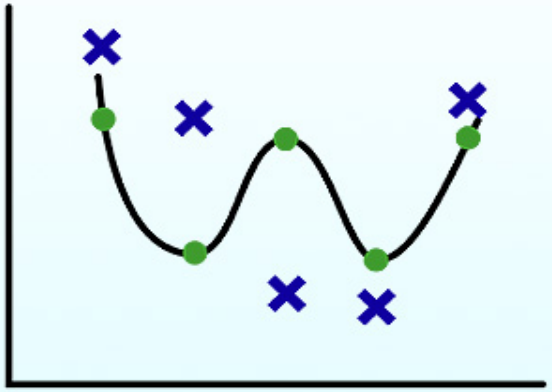
- **Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate.**

Plus intuition

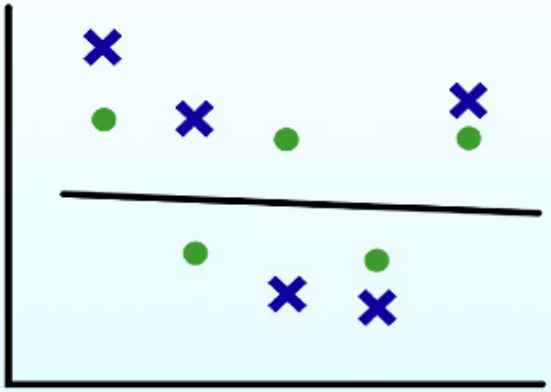
  - PROCESSING:
    - Think of your laptop that gets overwhelmed with 3-4 gig of data (disk space is 1TB)
  - STORAGE:
    - Laptop : 1 TB
  - THROUGH-PUT
    - 1TB would take 3 hours to read it using your laptop

- **Challenges**
  - Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, security, and information privacy.

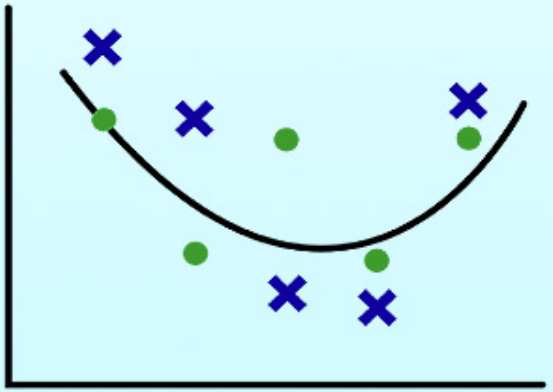
Bias-Variance Tradeoff in Model Selection in Simple Problem



(a) High variance/low bias.
4th-order polynomial ($p = 5$).

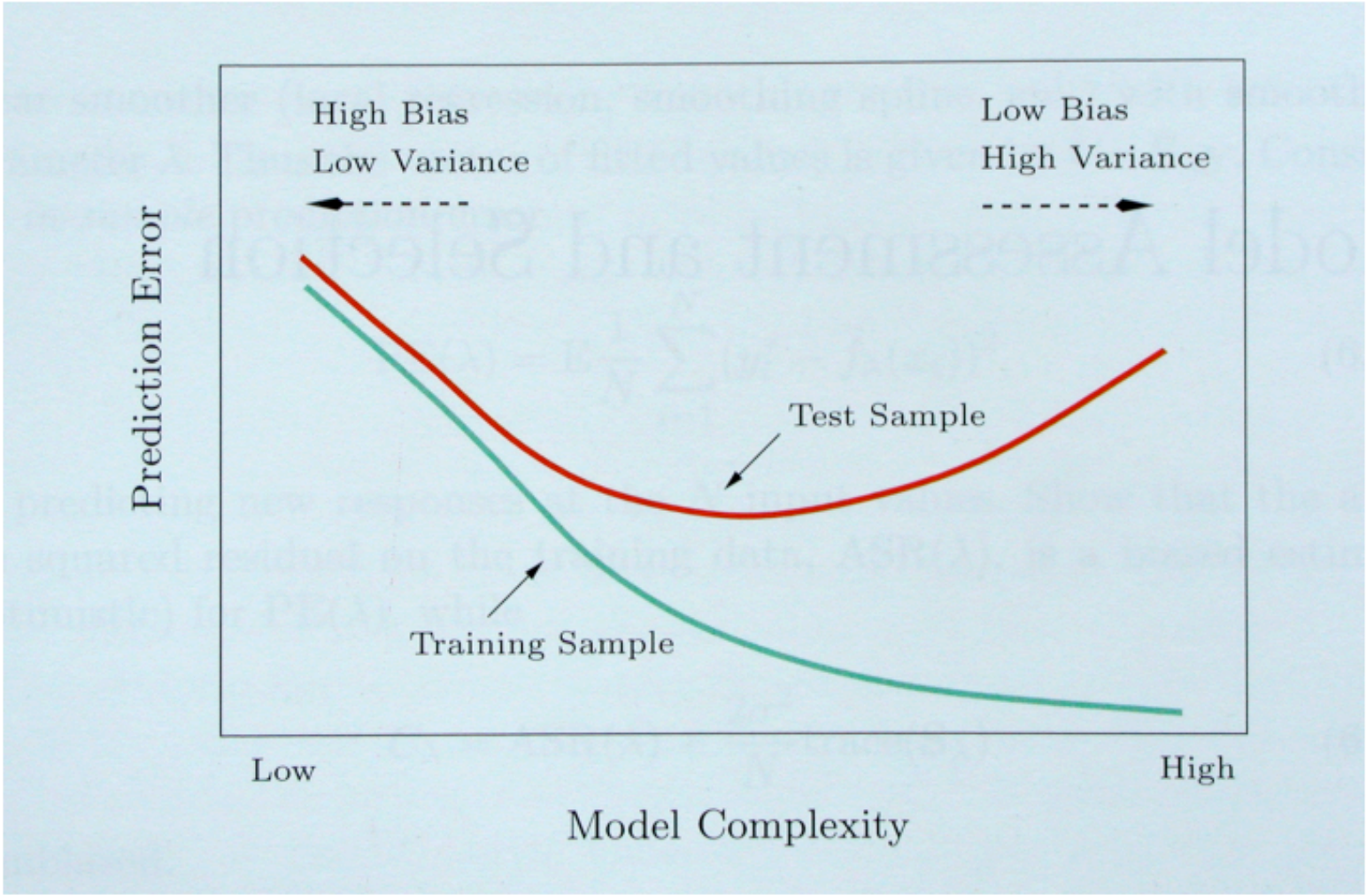


(b) Low variance/high bias.
1st-order polynomial ($p = 2$).



(c) Balanced variance & bias.
Minimum MSE.
2nd-order polynomial ($p = 3$).

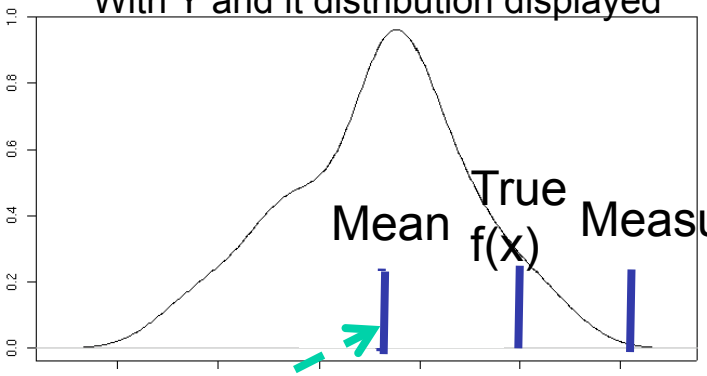
Bias/Variance Tradeoff



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

50 linear models (using different samples)

Graph Cross Section where X=1
With Y and it distribution displayed

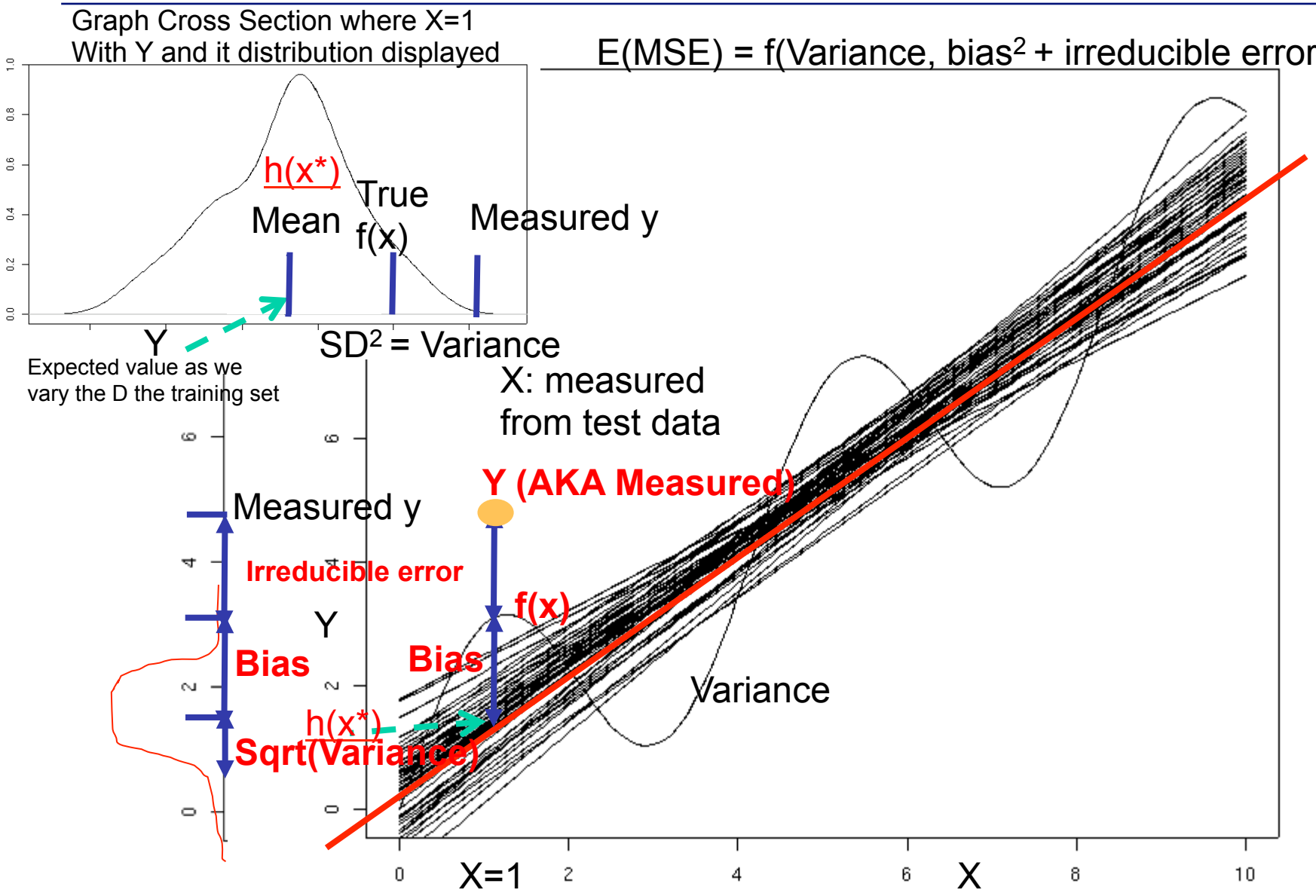


E(MSE) = f(Variance, bias^2 + irreducible error)

Expected value
as we vary the D
the training set

X: measured
from test data

50 linear models (using different samples)



Bias-Variance Analysis

- Given a new data point **x**, what is the **expected prediction error?**
- Assume that the data points are drawn i.i.d. from *a unique underlying probability distribution P*
- The goal of the analysis is to compute, for an arbitrary new point **x**,

$$E_P\left[(y - h(\mathbf{x}))^2\right]$$

where y is the value of **x** that could be present in a data set, and the expectation is over *all training sets* drawn according to P.

- We will decompose this expectation into three components:

bias, variance and noise

Bias-variance decomposition (2)

- Putting everything together, we have:

$$
\begin{aligned}
E_P\left[(y-h(\mathbf{x}))^2\right] &= E_P\left[(h(\mathbf{x})-\overline{h}(\mathbf{x}))^2\right] + \\
&\quad \overline{h}(\mathbf{x})^2 - 2f(\mathbf{x})\overline{h}(\mathbf{x}) + f(\mathbf{x})^2 + \\
&\quad E_P\left[(y-f(\mathbf{x}))^2\right] \\
&= E_P\left[(h(\mathbf{x})-\overline{h}(\mathbf{x}))^2\right] + \qquad \text{(variance)} \\
&\quad (h(\mathbf{x})-f(\mathbf{x}))^2 + \qquad \text{(bias)}^2 \\
&\quad E_P\left[(y-f(\mathbf{x}))^2\right] \qquad \text{(noise)} \\
&= \mathrm{Var}[h(\mathbf{x})] + \mathrm{Bias}[h(\mathbf{x})]^2 + E_P[\varepsilon^2] \\
&= \mathrm{Var}[h(\mathbf{x})] + \mathrm{Bias}[h(\mathbf{x})]^2 + \sigma^2
\end{aligned}
$$

- Expected prediction error = Variance + Bias^2 + Noise^2

Estimating Bias and Variance (continued)

- For each data point **x**, we will now have the observed corresponding value y and several predictions $y_1, \ldots, y_K$.
- Compute the average prediction h.
- Estimate **bias** as $(h - y)$
- Estimate **variance** as $\Sigma_k (y_k - h)^2/(K - 1)$
- Assume noise is 0

http://www-scf.usc.edu/~csci567/17-18-bias-variance.pdf

Bias, Variance, and Noise

- **Using a test data set with 20 data points**
- **For each data point x* in the test data set compute variance over the variance predictions (50 models give 50 predictions for each data point x*).**
- **For each data point x* calculate**
  - **Variance**: $E[(h(x^*) - \underline{h(x^*)})^2]$ ## $\Sigma(h(x^*) - \underline{h(x^*)})^2/50$
    - Describes how much h(x*) varies from one training set S to another
  - **Bias**: $[\underline{h(x^*)} - f(x^*)]$
    - Describes the <u>average</u> error of h(x*).
  - **Noise**: $E[(y^* - f(x^*))^2] = E[\varepsilon^2] = \sigma^2$
    - Describes how much y* varies from f(x*)

Bias-Variance written more formally for a single test point x*, using say 20 models

Using a test data set with 20 data points sum $(h(x^*) - y^*)^2$ over each of the 20 points and take the average

$$E_D[\,(h(x^*) - y^*)^2\,] =$$

Expected MSE wrt different models that are learned from different datasets D. E.g., 20 models yield 20 predictions $h_D(x^*)$

$$= E_D[\,(h_D(x^*) - \underline{h(x^*)})^2\,] + \textit{Variance} +$$

$$(\underline{h(x^*)} - f(x^*))^2 + \quad \textit{Bias}^2 +$$

$$E[\,(y^* - f(x^*))^2\,] \quad \textit{Noise}^2$$

$$= \mathrm{Var}(h(x^*)) + \mathrm{Bias}(h(x^*))^2 + E[\,\varepsilon^2\,]$$

$$= \mathrm{Var}(h(x^*)) + \mathrm{Bias}(h(x^*))^2 + \sigma^2$$

Expected prediction error = Variance + Bias^2 + Noise^2

Estimating Bias and Variance (continued)

- For each data point **x**, we will now have the observed corresponding value y and several predictions $y_1, \ldots, y_K$.
- Compute the average prediction h.
- Estimate **bias** as $(h - y)$
- Estimate **variance** as $\Sigma_k\,(y_k - h)^2/(K-1)$
- Assume noise is 0

$h_D(x^*)$ model prediction (assume 20 training datesets)
h(x*) Average model prediction
f(x*) TRUE (Actual function value)
Y* Observed target data (noisy)

http://www-scf.usc.edu/~csci567/17-18-bias-variance.pdf
Excellent Slides from Sofus

Bias-variance trade-off

- Consider fitting a logistic regression LTU to a data set vs. fitting a large neural net.
- Which one do you expect to have higher bias? Higher variance?
- Typically, *bias* comes from not having good hypotheses in the considered class
- *Variance* results from the hypothesis class containing too many hypotheses
- Hence, we are faced with a *trade-off:* choose a more expressive class of hypotheses, which will generate higher variance, or a less expressive class, which will generate higher bias.

Source of bias

- Inability to represent certain decision boundaries
  - E.g., linear threshold units, naïve Bayes, decision trees
- Incorrect assumptions
  - E.g, failure of independence assumption in naïve Bayes
- Classifiers that are "too global" (or, sometimes, too smooth)
  - E.g., a single linear separator, a small decision tree.

If the bias is high, the model is *underfitting* the data.

Source of variance

- Statistical sources
  - Classifiers that are "too local" and can easily fit the data
    - E.g., nearest neighbor, large decision trees
- Computational sources
  - Making decision based on small subsets of the data
    - E.g., decision tree splits near the leaves
  - Randomization in the learning algorithm
    - E.g., neural nets with random initial weights
  - Learning algorithms that make sharp decisions can be unstable (e.g. the decision boundary can change if one training example changes)

If the variance is high, the model is *overfitting* the data

Measuring Bias and Variance

- In practice (unlike in theory), we have only ONE training set S.
- We can simulate multiple training sets by bootstrap replicates
  - $S' = \{\mathbf{x} \mid \mathbf{x}$ is drawn at random with replacement from $S\}$ and $|S'| = |S|$.

Bias-Variance Tradeoff

Bias–variance decomposition of squared error [edit]

Suppose that we have a training set consisting of a set of points $x_1, \ldots, x_n$ and real values y_i associated with each point x_i. We assume that there is a functional, but noisy relation $y_i = f(x_i) + \epsilon$, where the noise, ϵ, has zero mean and variance σ^2.

We want to find a function $\hat{f}(x)$, that approximates the true function $y = f(x)$ as well as possible, by means of some learning algorithm. We make "as well as possible" precise by measuring the mean squared error between y and $\hat{f}(x)$: we want $(y - \hat{f}(x))^2$ to be minimal, both for $x_1, \ldots, x_n$ *and for points outside of our sample*. Of course, we cannot hope to do so perfectly, since the y_i contain noise ϵ; this means we must be prepared to accept an *irreducible error* in any function we come up with.

Finding an $\hat{f}$ that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function $\hat{f}$ we select, we can decompose its expected error on an unseen sample x as follows:[3]:34[4]:223

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = \mathrm{Bias}\left[\hat{f}(x)\right]^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

Where:

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x)\right] - f(x)$$

and

$$\mathrm{Var}\left[\hat{f}(x)\right] = \mathrm{E}\left[\left(\hat{f}(x) - \mathrm{E}[\hat{f}(x)]\right)^2\right]$$

The expectation ranges over different choices of the training set $x_1, \ldots, x_n, y_1, \ldots, y_n$, all sampled from the same distribution. The three terms represent:

- the square of the *bias* of the learning method, which can be thought of the error caused by the simplifying assumptions built into the method. E.g., when approximating a non-linear function $f(x)$ using a learning method for linear models, there will be error in the estimates $\hat{f}(x)$ due to this assumption;
- the *variance* of the learning method, or, intuitively, how much the learning method $\hat{f}(x)$ will move around its mean;
- the irreducible error σ^2. Since all three terms are non-negative, this forms a lower bound on the expected error on unseen samples.[3]:34

The more complex the model $\hat{f}(x)$ is, the more data points it will capture, and the lower the bias will be. However, complexity will make the model "move" more to capture the data points, and hence its variance will be larger.

using radial basis functions (blue). Several trials are shown in each graph. For each trial, a few noisy data points are provided as training set (top). For a wide spread (image 2) the bias is high: the RBFs cannot fully approximate the function (especially the central dip), but the variance between different trials is low. As spread decreases (image 3 and 4) the bias decreases: the blue curves more closely approximate the red. However, depending on the noise in different trials the variance between trials increases. In the lowermost image the approximated values for x=0 varies wildly depending on where the data points were located.

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

https://theclevermachine.wordpress.com/2013/04/21/model-selection-underfitting-overfitting-and-the-bias-variance-tradeoff/

Bias-Variance
Order 1, 2,3 polynomial

xGrid = linspace(-1,1,100);

```
% FIT MODELS TO K INDEPENDENT DATASETS
K = 50;
for iS = 1:K
    ySim = f(x) + noiseSTD*randn(size(x));
    for jD = 1:numel(degree)
        % FIT THE MODEL USING polyfit.m
        thetaTmp = polyfit(x,ySim,degree(jD));
        % EVALUATE THE MODEL FIT USING polyval.m
        simFit{jD}(iS,:) = polyval(thetaTmp,xGrid);
    end
end

% DISPLAY ALL THE MODEL FITS
h = [];
for iD = 1:numel(degree)
    figure(iD+1)
    hold on
    % PLOT THE FUNCTION FIT TO EACH DATASET
    for iS = 1:K
        h(1) = plot(xGrid,simFit{iD}(iS,:),'color',brighten(cols(iD,:),.6));
    end
    % PLOT THE AVERAGE FUNCTION ACROSS ALL FITS
    h(2) = plot(xGrid,mean(simFit{iD}),'color',cols(iD,:),'Linewidth',5);
    % PLOT THE UNDERLYING FUNCTION f(x)
    h(3) = plot(xGrid,f(xGrid),'color','k','Linewidth',3);
    % CALCULATE THE SQUARED ERROR AT EACH POINT, AVERAGED ACROSS ALL DATASETS
    squaredError = (mean(simFit{iD})-f(xGrid)).^2;
    % PLOT THE SQUARED ERROR
    h(4) = plot(xGrid,squaredError,'k--','Linewidth',3);
    uistack(h(2),'top')
    hold off
    axis square
    xlim([-1 1])
    ylim([-1 1])
    legend(h,{sprintf('Individual g_{%d}(x)',degree(iD)),'Mean of All Fits','f(x)','Squared Error'},'Location','WestOu
    title(sprintf('Model Order=%d',degree(iD)))
end
```

For polynomial =iD

iS Dataset Sample

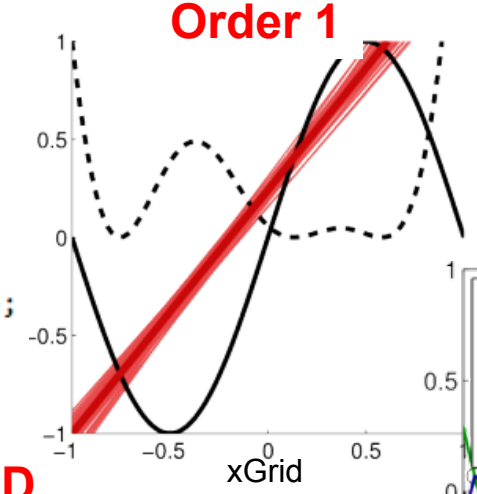
Predictions for sample set iS using model iD

Avg predictions of xGrid

BIAS

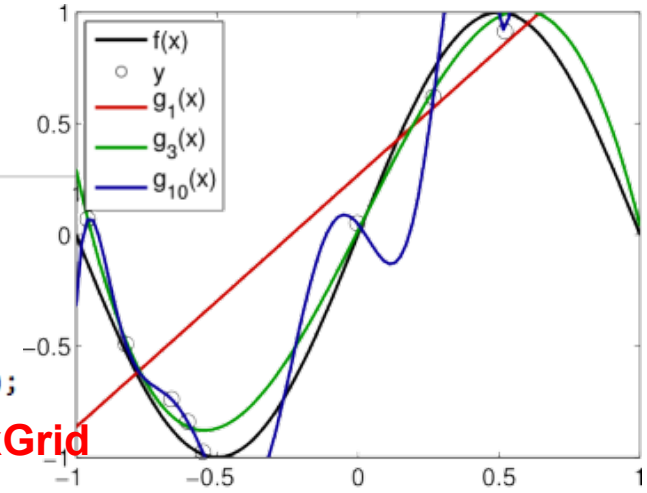
True error

Order 1



f(x) = sin(pi X x)

Order 1, 2, 3 with data

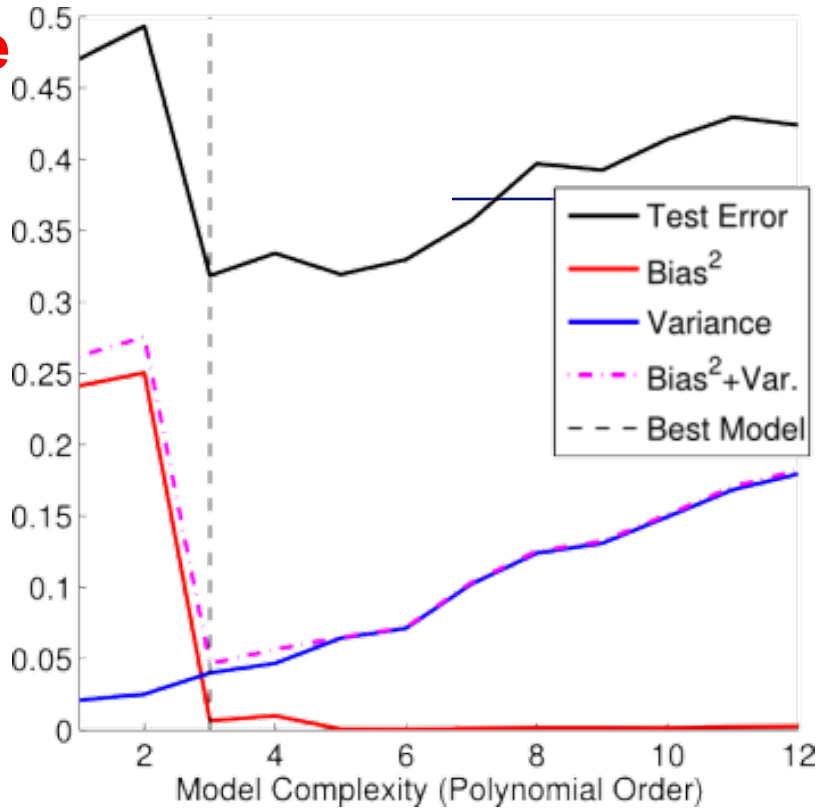


F(x) = true functi
ySim = noisy dat
simFit matrix of pr

Our original goal was to approximate f(x), not the data points per se.

Bias-Variance Order [1-12] polynomials

```
% # INITIALIZE SOME VARIABLES
xGrid = linspace(-1,1,N);
meanPrediction = zeros(K,N);
thetaHat = {};
x = linspace(-1,1,N);
x = x(randperm(N));
for iS = 1:K % LOOP OVER DATASETS
    % CREATE OBSERVED DATA, y
    y = f(x) + noiseSTD*randn(size(x));

    % CREATE TRAINING SET
    xTrain = x(1:nTrain);
    yTrain = y(1:nTrain);

    % CREATE TESTING SET
    xTest = x(nTrain+1:end);
    yTest = y(nTrain+1:end);

    % FIT MODELS
    for jD = 1:nPolyMax

        % MODEL PARAMETER ESTIMATES
        thetaHat{jD}(iS,:) = polyfit(xTrain,yTrain,jD);

        % PREDICTIONS
        yHatTrain{jD}(iS,:) = polyval([thetaHat{jD}(iS,:)],xTrain); TRAINING SET
        yHatTest{jD}(iS,:) = polyval([thetaHat{jD}(iS,:)],xTest);% TESTING SET

        % MEAN SQUARED ERROR
        trainErrors{jD}(iS) = mean((yHatTrain{jD}(iS,:) - yTrain).^2); % TRAINING
        testErrors{jD}(iS) = mean((yHatTest{jD}(iS,:) - yTest).^2); % TESTING
    end
end

% CALCULATE AVERAGE PREDICTION ERROR, BIAS, AND VARIANCE
for iD = 1:nPolyMax
    trainError(iD) = mean(trainErrors{iD});
    testError(iD) = mean(testErrors{iD});
    biasSquared(iD) = mean((mean(yHatTest{iD})-f(xTest)).^2);
    variance(iD) = mean(var(yHatTest{iD},1));
end
[~,bestModel] = min(testError);
```



Test Error = Variance(x_i)+ Bias(x_i)+irreducibleError

Avg(Bias (x_i))
Avg(Variance (x_i))

Best least squares fit for monomials of degree 1 to n.

- p = polyfit(x,y,n) returns the coefficients for a polynomial p(x) of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is n+1

polyfit

Polynomial curve

Syntax

```
p = polyfit(x,y,n)
[p,S] = polyfit(x,y,n)
[p,S,mu] = polyfit(x,y,n)
```

http://www.mathworks.com/help/matlab/ref/polyfit.html

Description

`p = polyfit(x,y,n)` returns the coefficients for a polynomial `p(x)` of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is n+1

$$p(x) = p_1 x^n + p_2 x^{n-1} + \ldots + p_n x + p_{n+1}.$$

`[p,S] = polyfit(x,y,n)` also returns a structure S that can be used as an input to `polyval` to obtain error estimates.

`[p,S,mu] = polyfit(x,y,n)` also returns `mu`, which is a two-element vector with centering and scaling values. `mu(1)` is `mean(x)`, and `mu(2)` is `std(x)`. Using these values, `polyfit` centers x at zero and scales it to have unit standard deviation

$$\hat{x} = \frac{x - \bar{x}}{\sigma_x}.$$

This centering and scaling transformation improves the numerical properties of both the polynomial and the fitting algorithm.

Examples

▼ Fit Polynomial to Trigonometric Function

Generate 10 points equally spaced along a sine curve in the interval `[0,4*pi]`.

```
x = linspace(0,4*pi,10);
y = sin(x);
```

Use `polyfit` to fit a 7th-degree polynomial to the po

```
p = polyfit(x,y,7);
```

Evaluate the polynomial on a finer grid and plot the re

```
x1 = linspace(0,4*pi);
y1 = polyval(p,x1);
figure
plot(x,y,'o')
hold on
plot(x1,y1)
hold off
```

```
> x1=0.1; x=c(x1^6, x1^5, x1^4,x1^3, x1^2, x1^1, 1)
> t(x) %*% ((c(0.0084,  -0.0983,    0.4217,   -0.7435,    0.1471,    1.1064,
0.00044117)))
          [,1]
[1,] 0.1118499
>
```

Fit a polynomial of degree 6

Determine the coefficients of the approximating polynomial of degree 6.

```
p = polyfit(x,y,6)
```

```
p =

    0.0084   -0.0983    0.4217   -0.7435    0.1471    1.1064    0.0004
```

To see how good the fit is, evaluate the polynomial at the data points and generate a table showing the data, fit, and error.

```
f = polyval(p,x);
T = table(x,y,f,y-f,'VariableNames',{'X','Y','Fit','FitError'})
```

```
T =

     X        Y         Fit         FitError
    ___    _______    _________    __________

      0          0    0.00044117    -0.00044117
    0.1    0.11246       0.11185     0.00060836
    0.2     0.2227       0.22231     0.00039189
    0.3    0.32863       0.32872    -9.7429e-05
    0.4    0.42839        0.4288    -0.0004066
    0.5     0.5205       0.52093    -0.0004256
    0.6    0.60386       0.60408    -0.0002282
    0.7     0.6778       0.67775     4.6383e-0
    0.8     0.7421       0.74183     0.0002699
    0.9    0.79691       0.79654     0.0003651
      1     0.8427       0.84238      0.000316
    1.1    0.88021       0.88005     0.0001594
    1.2    0.91031       0.91035    -3.9919e-0
    1.3    0.93401       0.93422      -0.00021
    1.4    0.95229       0.95258    -0.0002993
    1.5    0.96611       0.96639    -0.0002809
    1.6    0.97635       0.97652    -0.00016704
    1.7    0.98379       0.98379     8.3306e-07
    1.8    0.98909       0.98893     0.00016278
    1.9    0.99279       0.99253     0.00025791
      2    0.99532       0.99508     0.00024347
    2.1    0.99702       0.99691      0.0001131
    2.2    0.99814       0.99823    -8.8548e-05
    2.3    0.99886       0.99911    -0.00025673
    2.4    0.99931       0.99954    -0.00022451
    2.5    0.99959       0.99936     0.00023151
```

http://www.mathworks.com/help/matlab/ref/polyfit.html

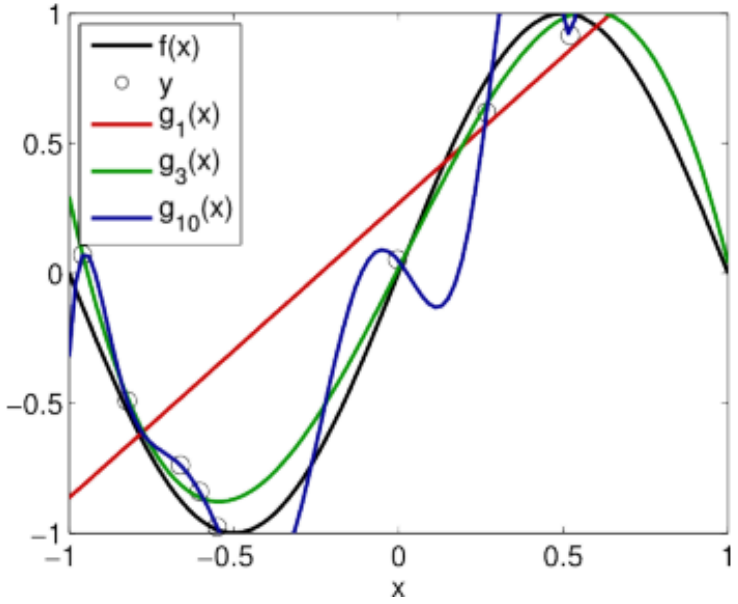
```
> x1=0.1; x=c(x1^6, x1^5, x1^4,x1^3, x1^2, x1^1, 1)
> t(x) %*% ((c(0.0084,  -0.0983,    0.4217,   -0.7435,    0.1471,    1.1
0.00044117)))
         [,1]
[1,] 0.1118499
>
```

ntact:James.Shanahan @ gmail.com

Fit a polynomial of degree 1, 3, 10

Below we estimate the parameters of three polynomial model functions of increasing complexity (using Matlab's `polyfit.m`) to the sampled data displayed above. Specifically, we estimate the functions $g_1(x)$, $g_3(x)$ and $g_{10}(x)$.

```
% FIT POLYNOMIAL MODELS & DISPLAY
% (ASSUMING PREVIOUS PLOT ABOVE STILL AVAILABLE)
degree = [1,3,10];
theta = {};
cols = [.8 .05 0.05; 0.05 .6 0.05; 0.05 0.05 .6];
for iD = 1:numel(degree)
    figure(1)
    theta{iD} = polyfit(x,y,degree(iD));
    fit{iD} = polyval(theta{iD},xGrid);
    h(end+1) = plot(xGrid,fit{iD},'color',cols(iD,:),'Linewidth',2);
    xlim([-1 1])
    ylim([-1 1])
end
legend(h,'f(x)','y','g_1(x)','g_3(x)','g_{10}(x)','Location','Northwest')
```



```
% FIT MODELS TO K INDEPENDENT DATASETS
K = 50;
for iS = 1:K
    ySim = f(x) + noiseSTD*randn(size(x));
    for jD = 1:numel(degree)
        % FIT THE MODEL USING polyfit.m
        thetaTmp = polyfit(x,ySim,degree(jD));
        % EVALUATE THE MODEL FIT USING polyval.m
        simFit{jD}(iS,:) = polyval(thetaTmp,xGrid);
    end
end

% DISPLAY ALL THE MODEL FITS
h = [];
for iD = 1:numel(degree)
    figure(iD+1)
    hold on
    % PLOT THE FUNCTION FIT TO EACH DATASET
    for iS = 1:K
        h(1) = plot(xGrid,simFit{iD}(iS,:),'color',brighten(cols(iD,:),.6));
    end
    % PLOT THE AVERAGE FUNCTION ACROSS ALL FITS
    h(2) = plot(xGrid,mean(simFit{iD}),'color',cols(iD,:),'Linewidth',5);
    % PLOT THE UNDERLYING FUNCTION f(x)
    h(3) = plot(xGrid,f(xGrid),'color','k','Linewidth',3);
    % CALCULATE THE SQUARED ERROR AT EACH POINT, AVERAGED ACROSS ALL DATASETS
    squaredError = (mean(simFit{iD})-f(xGrid)).^2;
    % PLOT THE SQUARED ERROR
    h(4) = plot(xGrid,squaredError,'k--','Linewidth',3);
    uistack(h(2),'top')
    hold off
    axis square
    xlim([-1 1])
    ylim([-1 1])
    legend(h,{sprintf('Individual g_{%d}(x)',degree(iD)),'Mean of All Fits','f(x)','Squared Error'},'Location','West
    title(sprintf('Model Order=%d',degree(iD)))
end
```

HW 1: Sample Submission

https://github.com/rocket-ron/MIDS-W261/blob/master/week01/HW1/MIDS-W261-2015-HWK-Week01-Cordell.ipynb

Homework HW1: Data set:Enron SPAM Mail

- The Enron Corpus is a large database of over 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission during its investigation after the company's collapse.
- The Enron data was originally collected at Enron Corporation headquarters in Houston during two weeks in May 2002 by Joe Bartling,[3] a litigation support and data analysis contractor working for Aspen Systems, now Lockheed Martin, whom the Federal Energy Regulatory Commission (FERC) had hired to preserve and collect the vast amounts of data in the wake of the Enron Bankruptcy in December 2001.

ENRON SPAM Data

- **This SPAM/HAM dataset for HW1 contains 100 records from the Enron SPAM/HAM corpus.**
- **There are about 93,000 emails in the original SPAM/HAM corpus. There are several versions of the SPAM/HAM corpus.**
- **Other Enron-Spam datasets are available from**
  - http://www.iit.demokritos.gr/skel/i-config/ and
  - http://www.aueb.gr/users/ion/publications.html in both raw and pre-processed form.

Enron Spam Data: Examples

Date and employee\tSPAM Tag \t Subject \t Email body

	A	B	C	D	E	F	G	H	I	J	K
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA							
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.							
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the							
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks, project apollo and beyond will conduct its final sap implementatio								
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs generating like mid-year documenting business unit performance							
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get some coffee with you, but remember that annoying project that m							
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high island larger block a-1 # 2 commenced on saturday at 2:00 p.m. at a							
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution of the central maine sos deal! this is another great example of wha							
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email address was obtained from a purchased list, reference # 2020 mid = 330							
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! space saving computer in a keyboard eliminate that big box comp							
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe hendricks (mrs) of rsa. i am writing this letter to you with the hop							
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine. i have talked to shirley and have directions. thanks, bob vince j							
13	0003.1999-12-14.farmer	0	calpine daily gas nomination	-calpine daily gas nomination 1. doc							
14	0003.2000-01-17.beck	0	re: additional responsibility	congratulations on this additional responsibility! i will be more than happy to help support your ne							
15	0003.2001-02-08.kitchen	0	re: key hr issues going forward	all is under control: a-we've set up a " work-out " group under cindy skinner and will be producing t							
16	0003.2003-12-18.GP	1	fw: account over due wfxu ppmfztdtet	eliminate your credit card debt without bankruptcy! tired of making minimum payments and barely							
17	0003.2004-08-01.BG	1	whats new in summer? bawled	carolyn regretful watchfully procrustes godly summer 2004 was too hot for the software manufactu							
18	0004.1999-12-10.kaminski	0	research group move to the 19 th floor	hello all: in case any of you feel energetic, " the boxes are here ". they are located at 2963 b (micha							
19	0004.1999-12-14.farmer	0	re: issue	fyi-see note below-already done. stella ----------forwarded by stella l morris/hou/ect on 12							
20	0004.2001-04-02.williams	0	enrononline desk to desk id and password	bill, the epmi-st-wbom book has been set up as an internal counterparty for desk-to-desk trading o							
21	0004.2001-06-12.SA_and_HP	1	spend too much on your phone bill? 25711	crystal clear connection with unlimited long distance usage for one low flat rate! now try it for fre							
22	0004.2004-08-01.BG	1	NA	h$ ello dea 54 r home owner, we have beetcn notiffiyved that your morayt " goage r [ate is fixed a							
23	0005.1999-12-12.kaminski	0	christmas baskets	the christmas baskets have been ordered. we have ordered several baskets. individual earth-sat fr							
24	0005.1999-12-14.farmer	0	meter 7268 nov allocation	fyi. ----------forwarded by lauri a allen/hou/ect on 12/14/99 12:17 pm----------							
25	0005.2000-06-06.lokay	0	transportation to resort	please be informed, a mini-bus has been reserved for your convenience in transporting you to the s							

Examples 1-4 (not SPAM) and 9 (SPAM)

1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. your vendor selection team will receive an update and even more information later in the week. on the lunch & learn for energy operations, the audience and focus will be your group. …….[TRUNCATED]
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks, project apollo and beyond will conduct its final sap implementation) [TRUNCATED] NOTE: No Body text for this email	
9				oo thank you, your email address was obtained from a purchased list, reference # 2020 mid = 3300. if you wish to unsubscribe from this list, please click here and enter your name into the remove box. if you have previously unsubscribed and are still receiving this message, you may email our abuse control center, or call 1-888-763-2497, or write us at: nospam, 6484

WordCount Example 3

Simple Spark Apps: *WordCount*

Definition:

count how often each word appears in a collection of text documents

This simple program provides a good test case for parallel processing, since it:

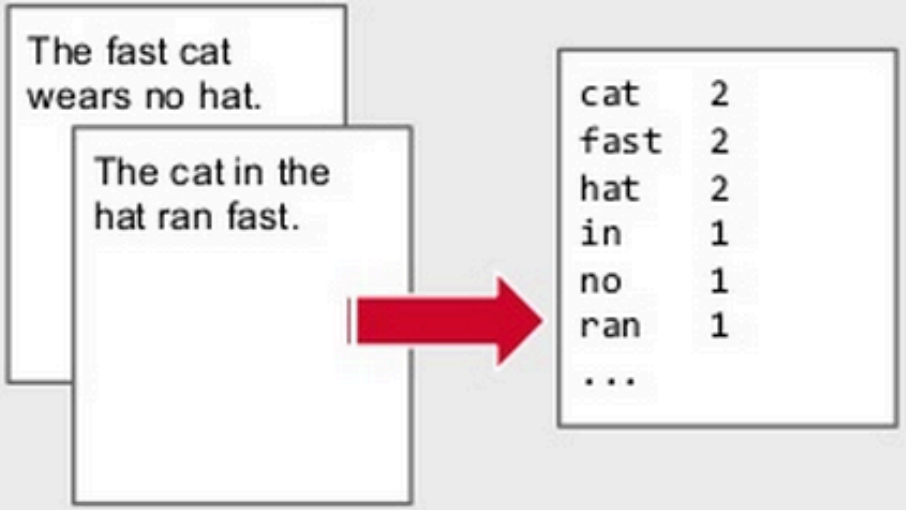
- requires a minimal amount of code
- demonstrates use of both symbolic and numeric values
- isn't many steps away from search indexing
- serves as a "Hello World" for Big Data apps

```
void map (String doc_id, String text):
  for each word w in segment(text):
    emit(w, "1");


void reduce (String word, Iterator group):
  int count = 0;

  for each pc in group:
    count += Int(pc);

  emit(word, String(count));
```

A distributed computing framework that can run WordCount **efficiently in parallel at scale** can likely handle much larger and more interesting compute problems

Word Count in Map-Reduce

```
def map(key, value):
    [redacted]
        emit(word, 1)

def reduce(key, values):
    [redacted]
        count += val
    emit(key, count)

# emit is a function that performs distributed I/O
```



Each document is passed to a mapper, which does the tokenization. The output of the mapper is reduced by key (word) and then counted.

What is the data flow for word count?

Word Count broken down

```
1 hello hi hi hallo
2 bonjour hola hi ciao
3 nihao konnichiwa ola
4 hola nihao hello
```

[hello, hi, hi, hallo, bonjour, hola, hi, ciao, ao, konnichiwa, ola, hola, nihao, hello]

Mapper

```
(hello,1),
(hi,1),
(hi,1),
(hallo,1),
(bonjour,1),
(hola,1),
(hi,1),
(ciao,1),
(nihao,1),
(konnichiwa,1),
(ola,1),
(hola,1),
(nihao,1),
(hello,1)
```

reduce ByKey

```
(u'ciao', 1)
(u'bonjour', 1)
(u'nihao', 2)
(u'hola', 2)
(u'konnichiwa', 1)
(u'hallo', 1)
(u'hi', 3)
(u'hello', 2)
(u'ola', 1)
```

SPAM Data Word Count : HW1.2

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! s
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. You… [TRUNCATED BODY]**

WordCount example reads text files and counts how often words occur. The input is text files and the output is text files, each line of which contains a word and the count of how often it occured, separated by a tab.

MAPPER: Each mapper takes a line as input and breaks it into words. It then emits a key/value pair of the word and 1.

REDUCER: Each reducer sums the counts for each word and emits a single key/value with the word and sum.

SPAM Data Word Count : HW1.2

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!!
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below, is an update on the leadership pilot. You… [TRUNCATED BODY]**

As an optimization, the reducer is also used as a combiner on the map outputs. This reduces the amount of data sent across the network by combining each word into a single record. [See Lecture 3]

WordCount example … ccur. The input is text files and the outp… word and the count of how often it …

MAPPER: Each map… rds. It then emits a key/value pair of the word and 1.

REDUCER: Each reducer sums the counts for each word and emits a single key/value with the word and sum.

Word Count : HW1.2: Mapper

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D	E
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA	
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.	
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s	
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,		
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs	
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get	
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is	
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution	
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres	
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! s	
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe	
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine	

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. You… [TRUNCATED BODY]**

For each record

- extract field 3 and 4 (Subject and Email Body) and split into tokens (words/numbe
- Out a list of token-count pairs

Output from Mapper

Mapper Output	Key=Word	Value=Count
Line 1	leadership	1
2	development	1
3	you	5

From Word Counts to Mulitnomial Naïve Bayes HW1.3

- **Mapper(Key=DocID, Value=(SPAM, Subject, Body)**
- **Output: 3 different types of information**
  - Words and their class conditional counts (partial)
    - E.g., Assistance SPAM, 3
  - Class word counts
    - E.g., SPAM Word Count, 45
    - E.g., SPAM Doc Count, 1

- **Reducer(Key, Value)**
- **Output: Naïve Bayes Model**
  - Class conditionals Pr(X|Y); E.g., SPAM, Assistance 0.0001
  - Class prior Pr(Y); E.g., SPAM Prior =0.5

Peer Grading

- **Details to follow**

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
- **HW1: review and grading**
- **Q&A (WK02)**
- **HW2:**
- **Naive Bayes**
  - Various Naïve Bayes Flavours
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Parallel computing, MapReduce, Hadoop (Data Storage and Algorithms)

James G. Shanahan[1]

[1]NativeX and iSchool, UC Berkeley, CA

EMAIL: James_DOT_Shanahan_AT_gmail_DOT_com

Week 2 review


hadoop
"Swiss army knife of the 21st century"
Media Guardian Innovation Awards
http://www.guardian.co.uk/technology/2011/mar/25/media-guardian-innovation-awards-apache-hadoop

Live Session #2

- **Class logistics (office hours, homework)**
- **Naïve Bayes review**
- **Hadoop Async summary**
  - Hadoop: detailed example wordcount (Michael Nolls)
  - Character count
- **Q&A**
- **Install Hadoop**
- **Run Word Count NoteBook**
- **Start homework**

Naïve Bayes

- Naïve Bayes (see live session #1)
- Is it Bernoulli versus Multinomial?
- Why smoothing?
- Underflow issues

Unit 2 | Parallel Computing, MapReduce, and Hadoop (Data Storage and Algorithms)

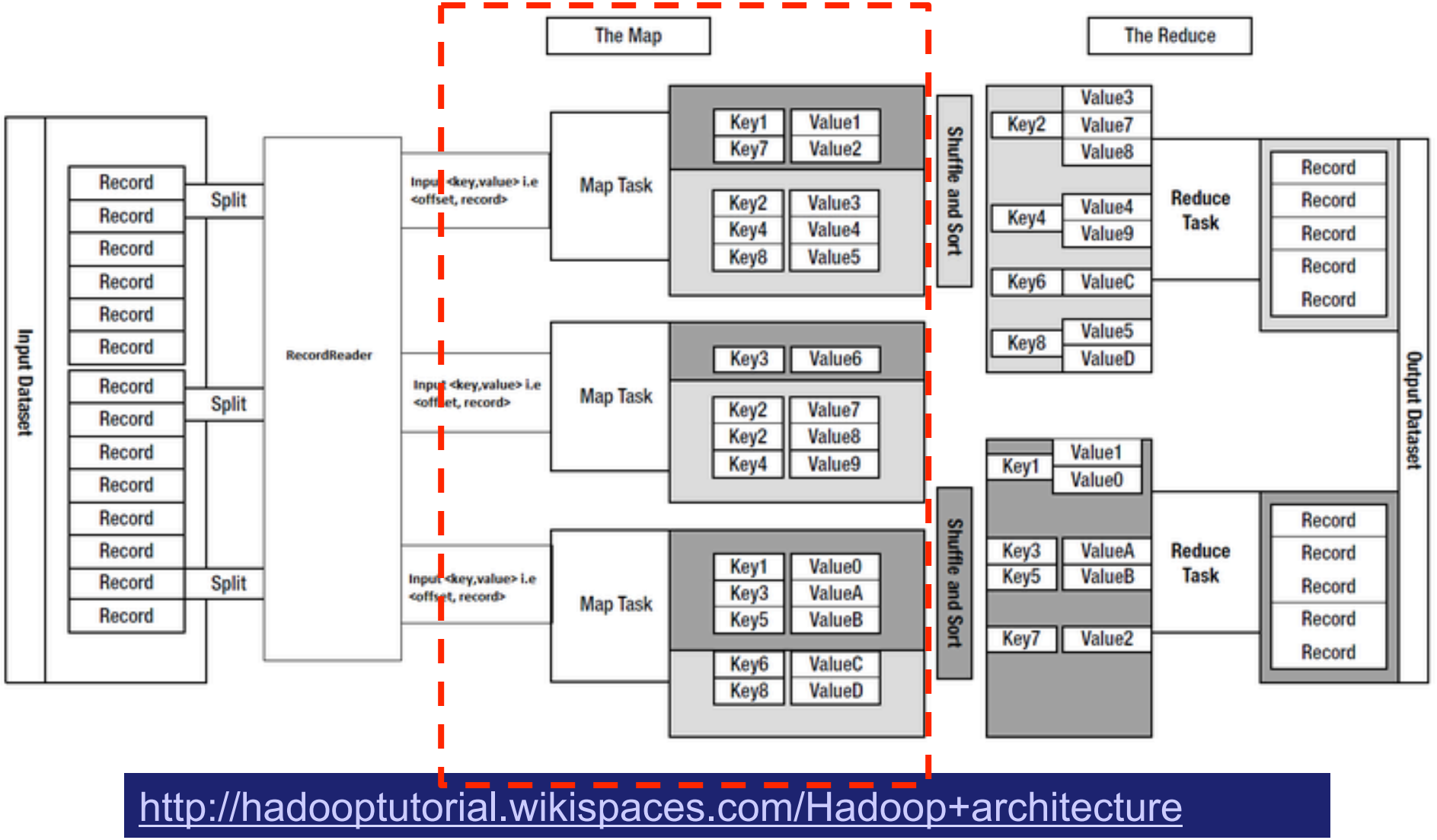
Hide Contents

- 2.1 Weekly Introduction (2 mins)
- 2.2 Assigned Readings
- 2.3 Motivation for Parallel Computing (8 mins)
- 2.4 Parallel Computing Definition and Communication Synchronization Types of PC Tasks (16 mins)
  - 2.4.1 Quiz: Embarrassingly Parallel Problems
- 2.5 Architectures for Parallel Computation (11 mins)
- 2.6 Developer Frameworks for Parallel Computation (13 mins)
  - 2.6.1 Quiz: Shared Nothing
- 2.7 Hadoop Background and History (8 mins)
- 2.8 Hadoop File System (8 mins)
- 2.9 MapReduce: Functional Programming (9 mins)
- 2.10 Hadoop: MapReduce (9 mins)
- 2.11 Animated Examples (13 mins)
- 2.12 Summary (2 mins)

MapReduce a framework for big data

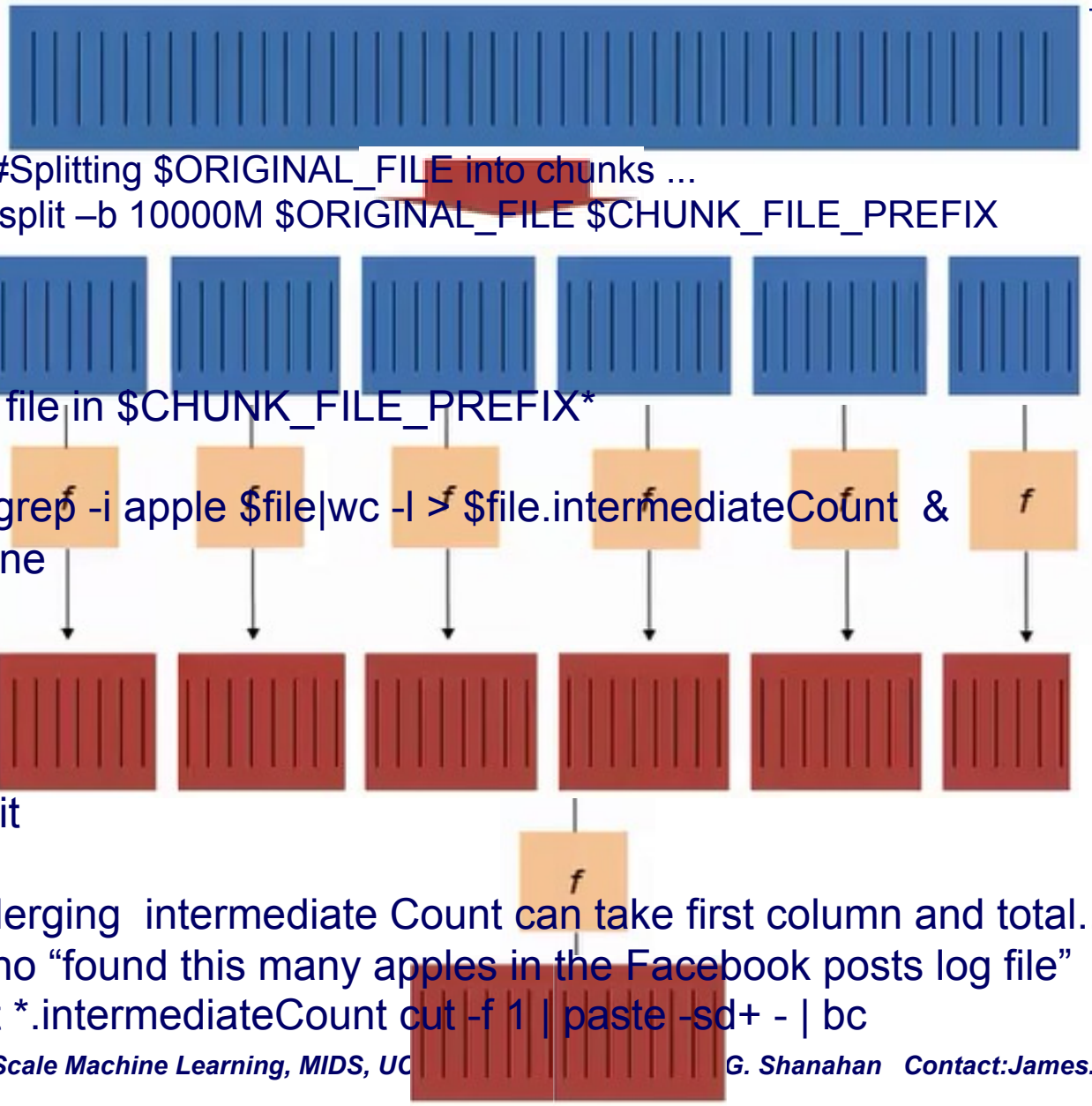
- **MapReduce codifies a generic recipe for processing large datasets that consists of two stages.**
  - In the first stage, a user-specified computation is applied over all input records in a dataset.
  - These operations occur in parallel and yield intermediate output that is then aggregated by another user-specified computation.
- **Programmer and execution framework synergy**
- **Just provide the mapper and reducer functions**
  - The programmer defines these two types of computations, and the execution framework coordinates the actual processing (very loosely, MapReduce provides a functional abstraction).
- **Very powerful: many interesting algorithms can be expressed quite concisely**
  - Although such a two-stage processing structure may appear to be very restrictive, many interesting algorithms can be expressed quite concisely, especially if one decomposes complex algorithms into a sequence of MapReduce jobs

Map Reduce: Data flow



http://hadooptutorial.wikispaces.com/Hadoop+architecture

Schematic of Parallel Processing



A big data file

File is split into smaller 100Gig chunks and distributed to k nodes

f is a function that operates on each data item

A distributed set of answers

Merge outputs

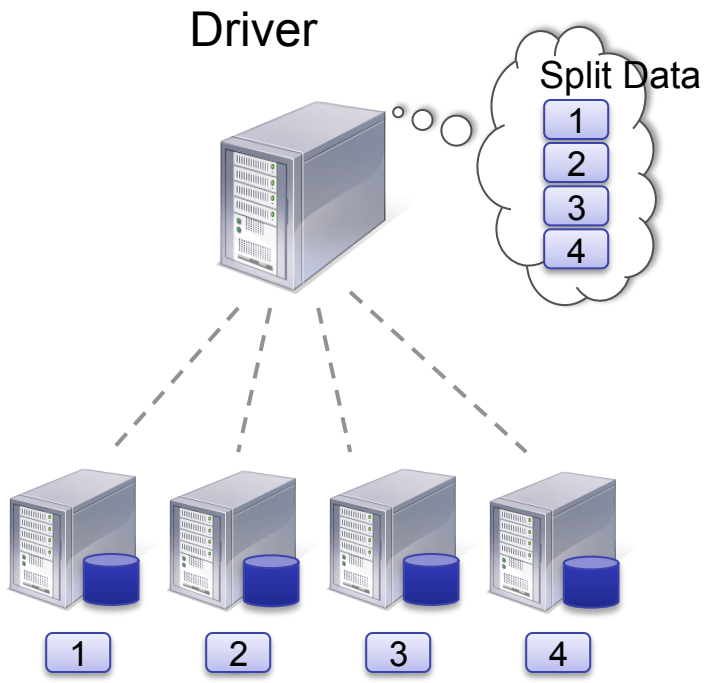
```
1. #Splitting $ORIGINAL_FILE into chunks ...
2.  split –b 10000M $ORIGINAL_FILE $CHUNK_FILE_PREFIX

for file in $CHUNK_FILE_PREFIX*
do
   grep -i apple $file|wc -l > $file.intermediateCount  &
Done

wait

#Merging  intermediate Count can take first column and total...
echo “found this many apples in the Facebook posts log file”
cat *.intermediateCount cut -f 1 | paste -sd+ - | bc
```

Command Line: Divide and Conquer

- **Partitions the data**
- **The framework processes the objects within a partition in sequence, and can processes multiple partitions in parallel**

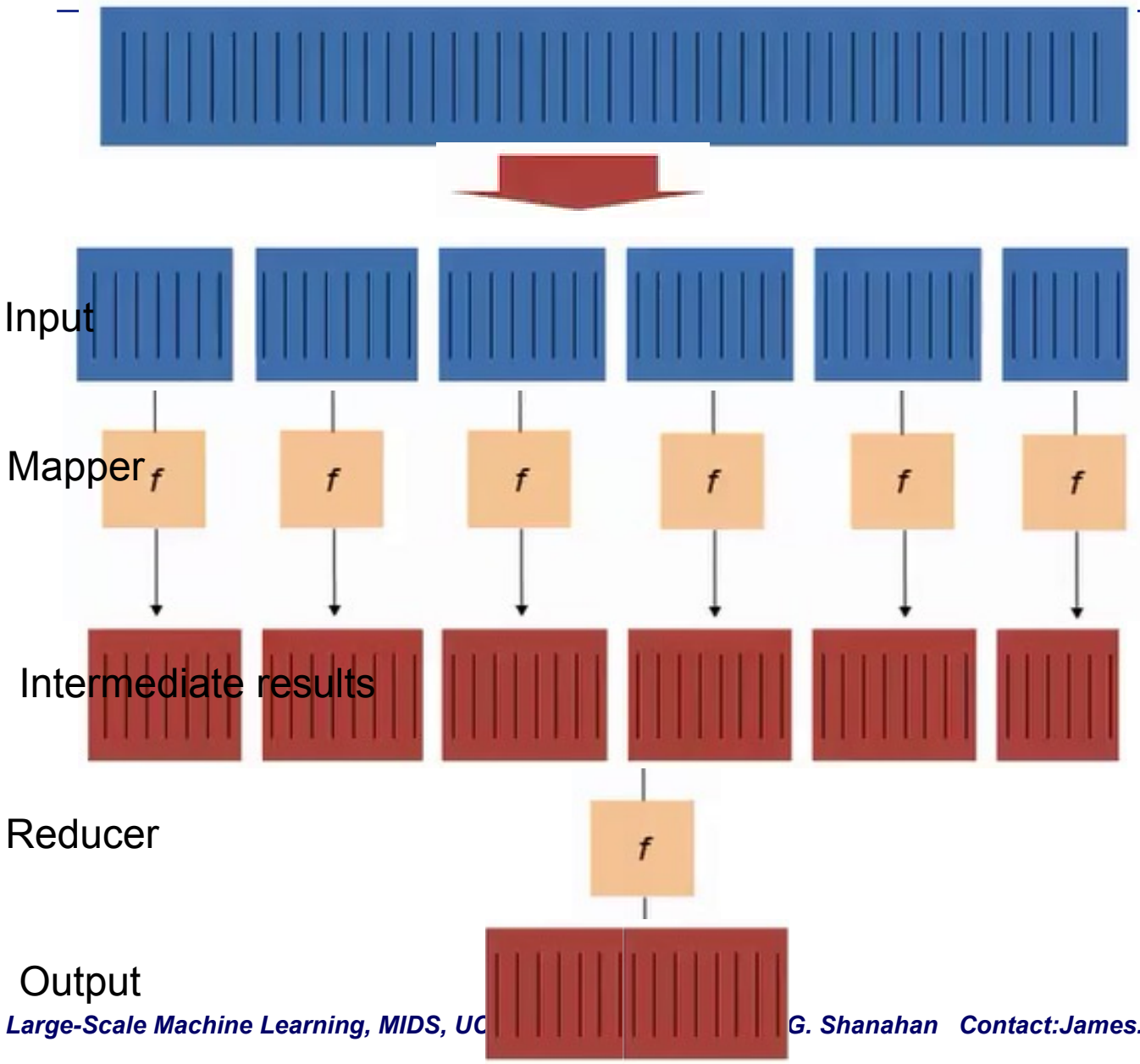
Challenges?

Partition and distribute data



Slave Processors or Computers
Executors

Schematic of Parallel Processing



A big data file

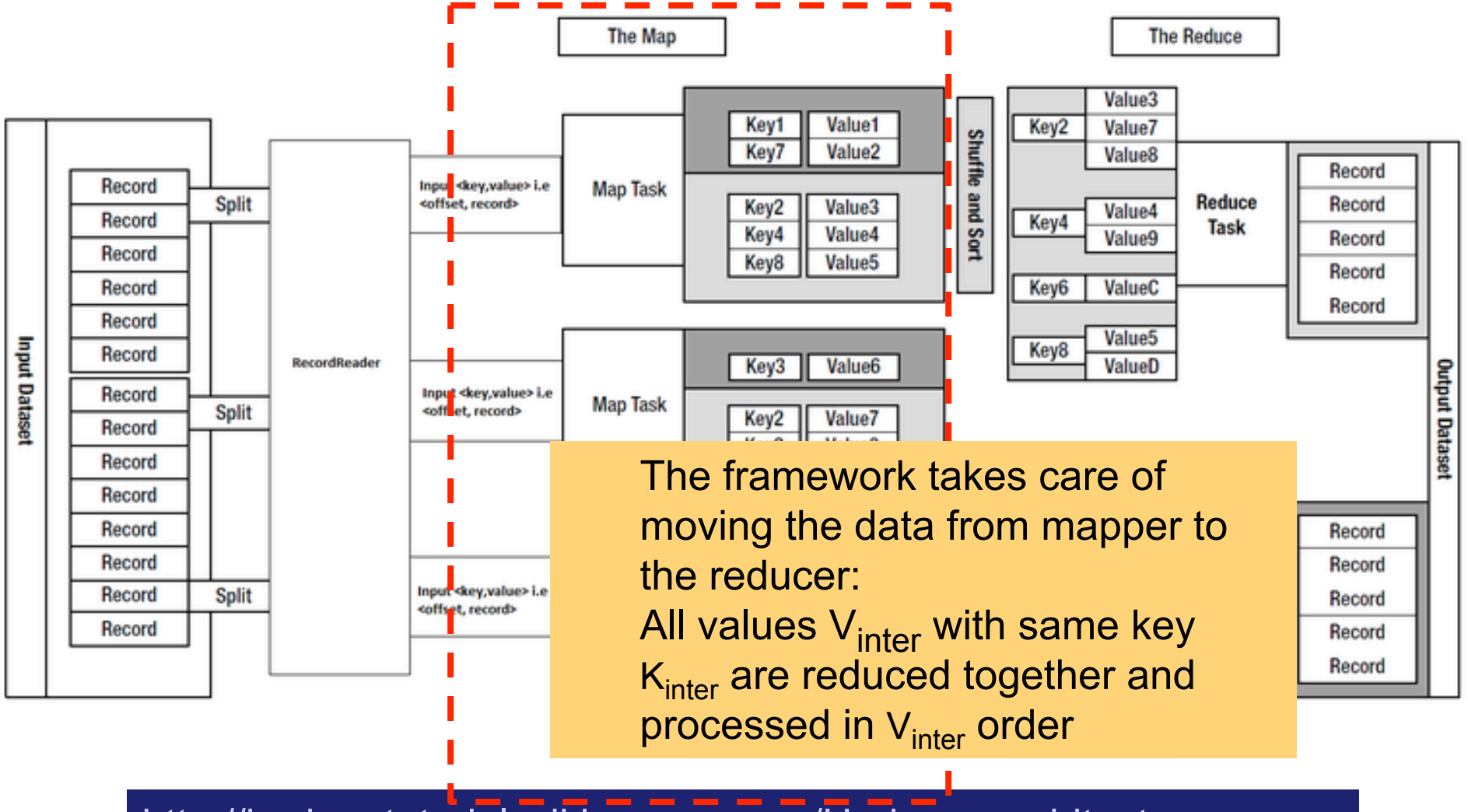
File is split into smaller chunks and distributed to k nodes

f is a function that operates on each data item

A distributed set of answers

Merge outputs

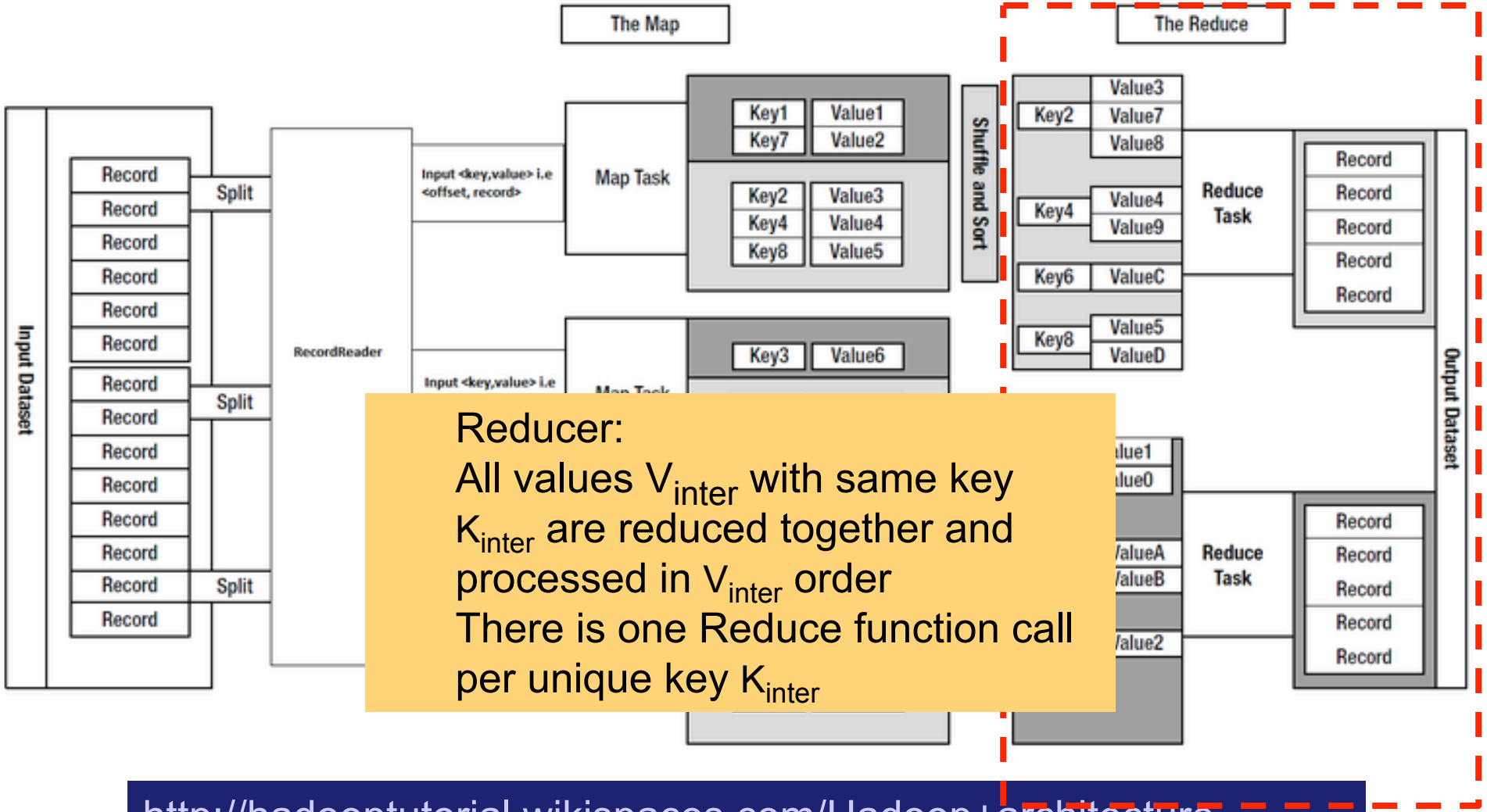
Map Reduce: Mapper→ Reducer



The framework takes care of moving the data from mapper to the reducer:
All values V_{inter} with same key K_{inter} are reduced together and processed in V_{inter} order

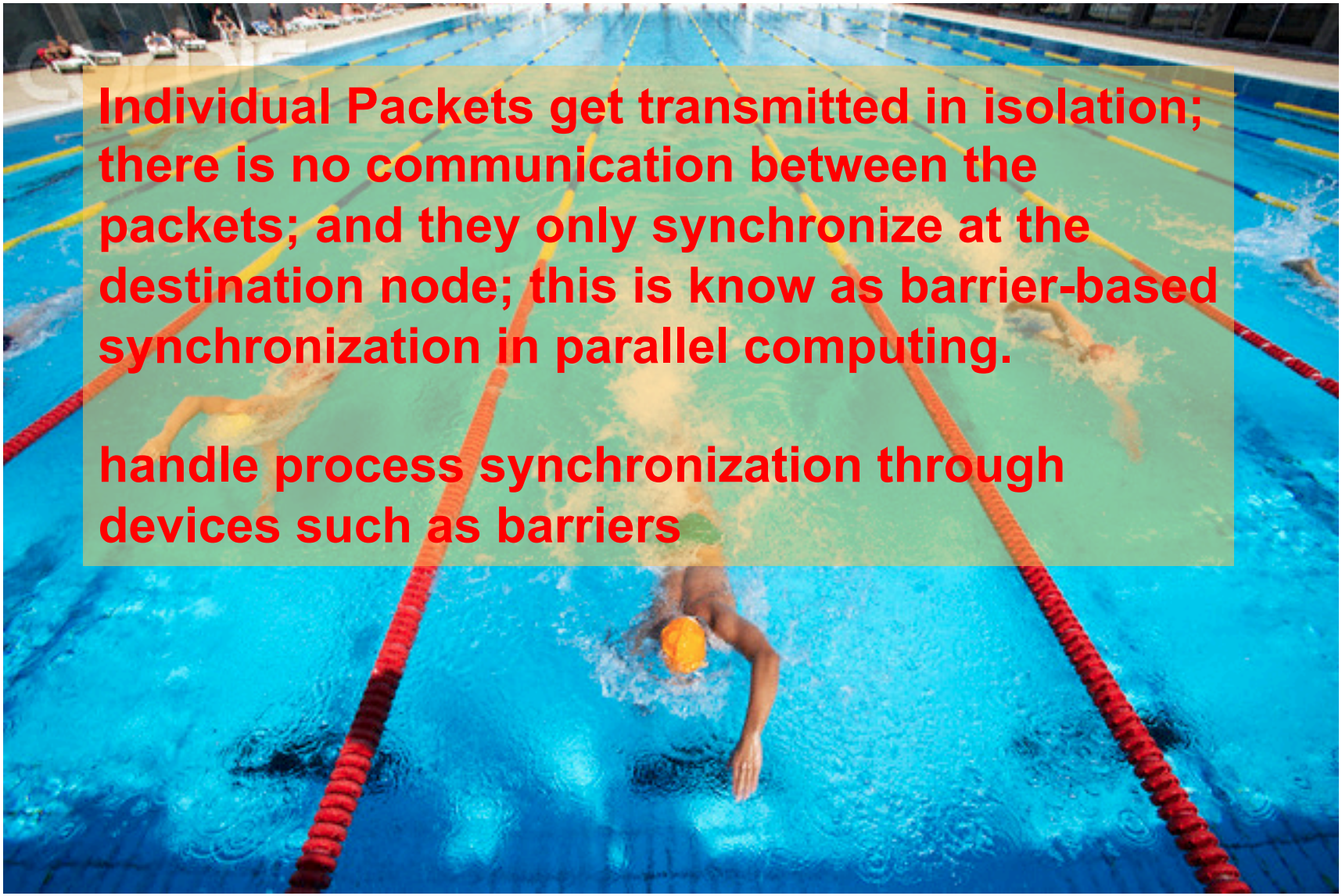
http://hadooptutorial.wikispaces.com/Hadoop+architecture

Map Reduce: Reduce



http://hadooptutorial.wikispaces.com/Hadoop+architecture

Individual Packets get transmitted in isolation



- **The key to success here was Divide and Conquer**
- **Decompose a large task into smaller ones**

- **We came up with a very nice framework for parallelizing tasks on the command line!!**
  - But it is limited
  - Granularity of task is somewhat coarse
  - No fault tolerance
  - No control over shared filespace
- **Divide and conquer does not come for free: there are obligations in terms of communication, synchronization, and fault tolerance**

Issues to be addressed

- How to break large problem into smaller problems? Decomposition for parallel processing
- How to assign tasks to workers distributed around the cluster?
- How do the workers get the data?
- How to synchronize among the workers?
- How to communicate with works?
- How to share partial results among workers?
- How to do all these in the presence of errors and hardware failures?

Divide and conquer does not come for free: there are obligations in terms of communication, synchronization, and fault tolerance

- **What does Embarrassingly Parallel mean?**

- **Give an example in Machine Learning of an Embarrassingly Parallel problem.**

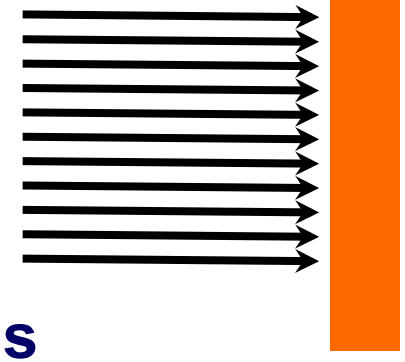
Data Parallelism – Embarrassingly Parallel Tasks

- Little or no effort is required to break up the problem into a number of parallel tasks, and there exists no dependency (or communication) between those parallel tasks.
- Examples:
  - map() function in Python:

```
>>> x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> map(lambda e: e*e, x)
>>> [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Synchronization thru a Barrier

- A barrier for a group of threads or processes in the source code means any thread/process must stop at this point and cannot proceed until all other threads/ processes reach this barrier.
- Another popular way of syncing is the barrier method;
- Explicitly handle process synchronization through devices such as barriers
- it is pretty effective, and very coarse grained and can be great in certain types of problems.
- In parallel computing, a barrier is a type of synchronization method.



- Better than MPI and shared memory solutions

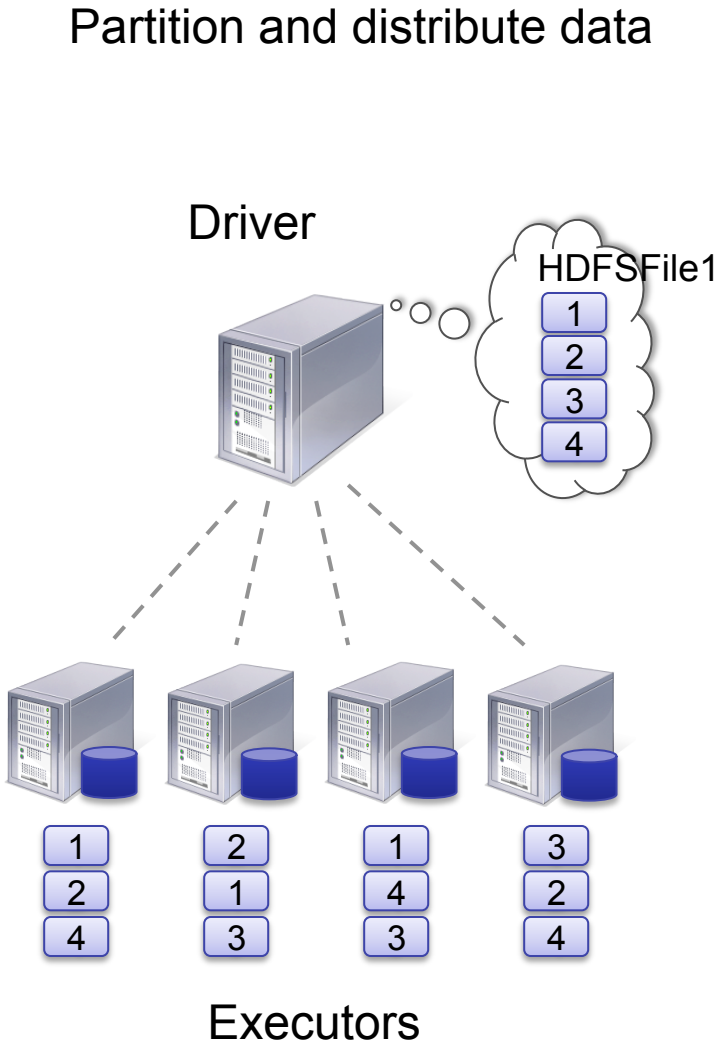
Embarrassingly Parallel Candidate Operations

- **Operations on our data**
  - Commutative, associative
  - In Naïve Bayes: Focus on say the Summation operation for calculating Pr(X|Y)
  - Numerator Total = Count from mapper1 + Count from mapper2, CM3+CM4

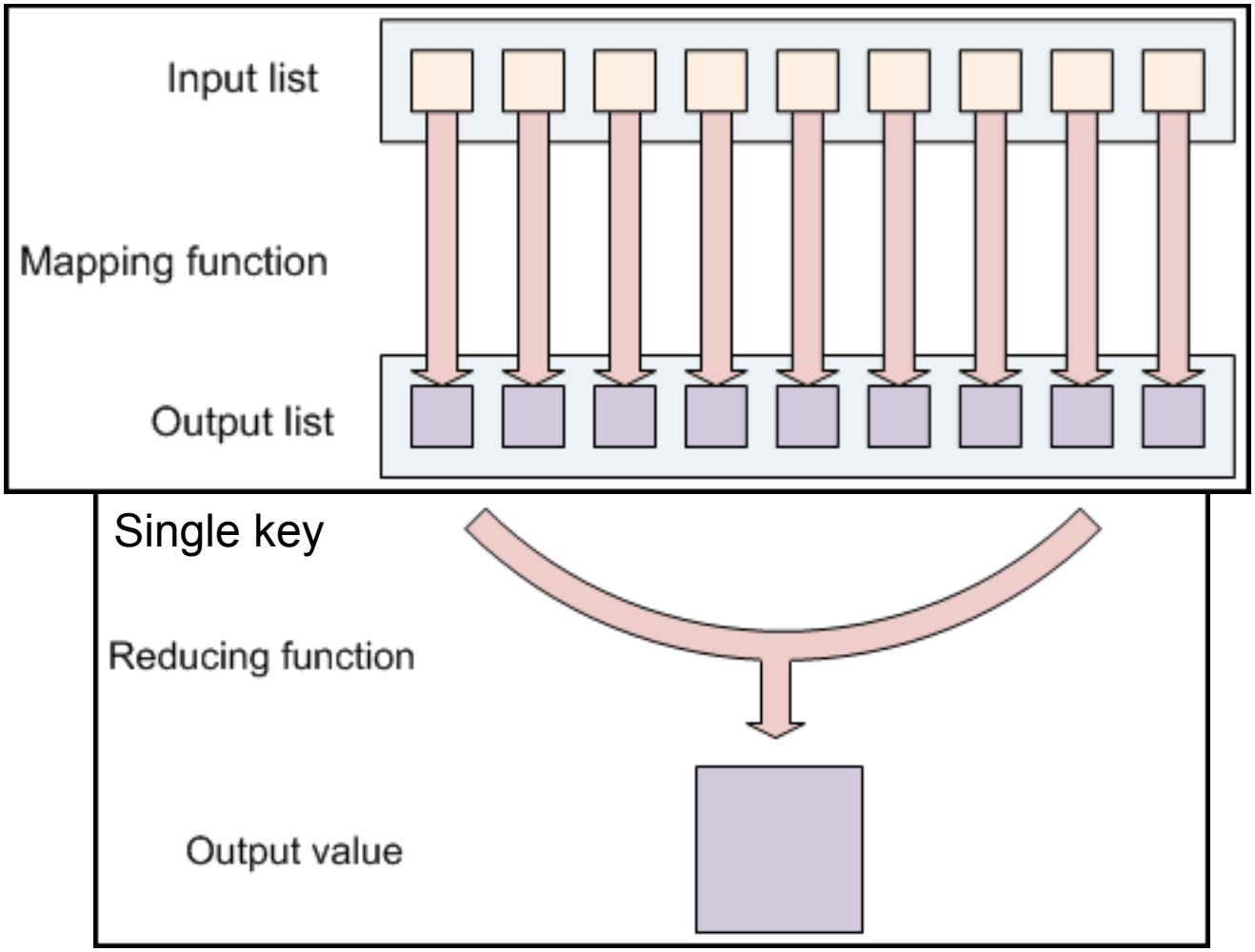
  - Operation needs to be associative
    - E.g., (1, 0, 4, 5, 6,8) + (10, 20, 0, 5) + (0, 30, 0, 3)
    - Also the following grouping is fine (1, 0, 4, 5) + (6,8, 10, 20, 0, 5) + (0, 30, 0, 3)
  - And Operation needs to be commutative

Data is a distributed collection of blocks

- **Data is laid out across a cluster of machines as collection of Partitions, each including a subset of the data**
- **The framework processes the objects within a partition in sequence, and processes multiple partitions in parallel.**
- **Data (e.g., clicks, or record linkage) is stored in a text file, with one observation on each line.**
  - JSON, zipped, AVRO, Parquet



MapReduce: Example Sum(X^2)



MapReduce: Example Sum(X^2)

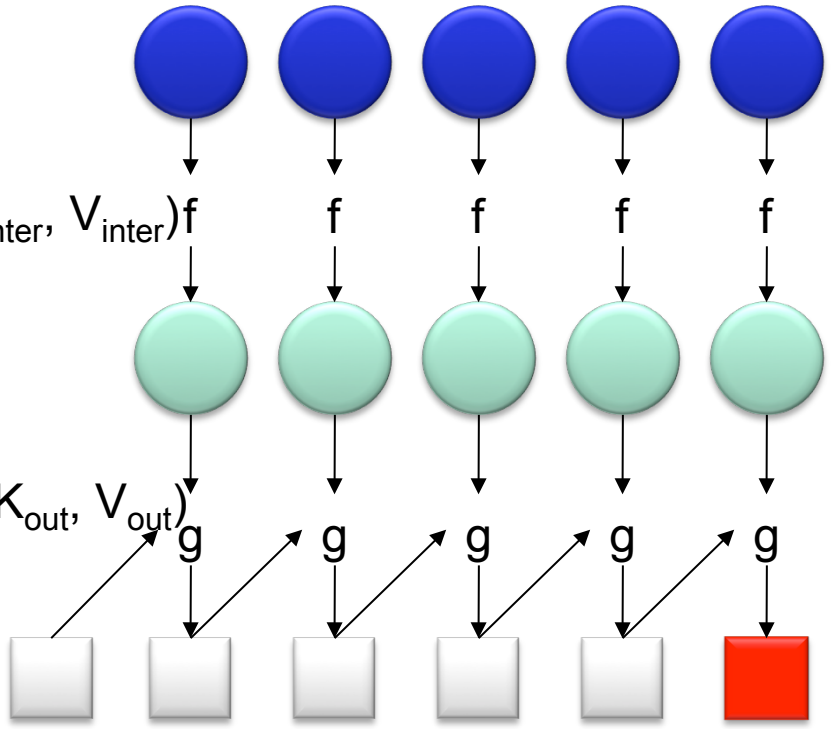
- **MapReduce is oriented around Key-Value pairs records**
- **Data is stored as key-value *records* in Hadoop**
  - output= Run(Mapper=f(x)=x × x, Reducer=sum(), input=inFile)

- **Map function:**

$Map(Key_{in}, Value_{in}) \rightarrow list(K_{inter}, V_{inter})$

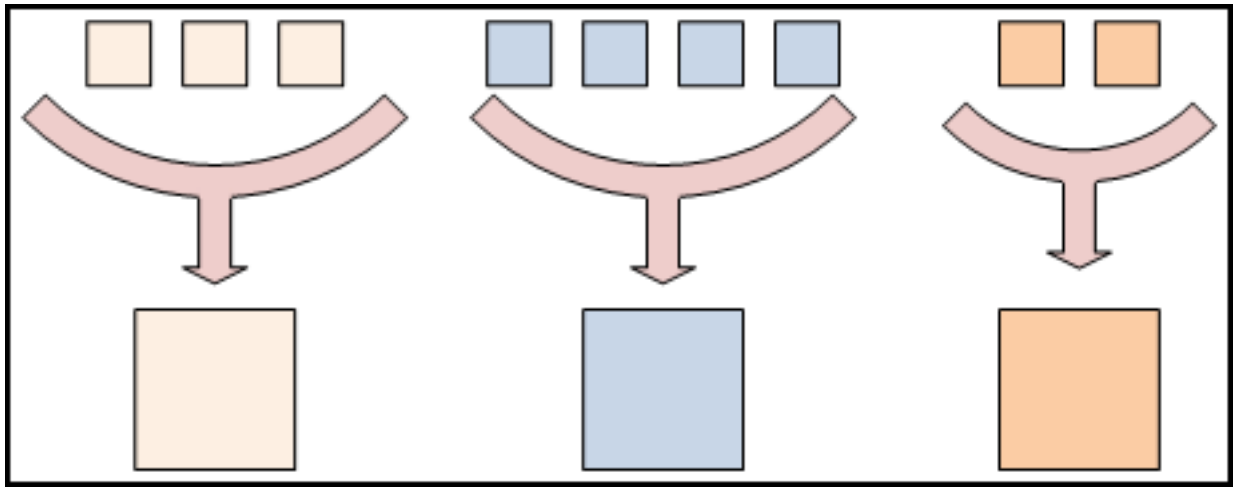
- **Reduce function:**

$Fold(K_{inter}, list(V_{inter})) \rightarrow list(K_{out}, V_{out})$

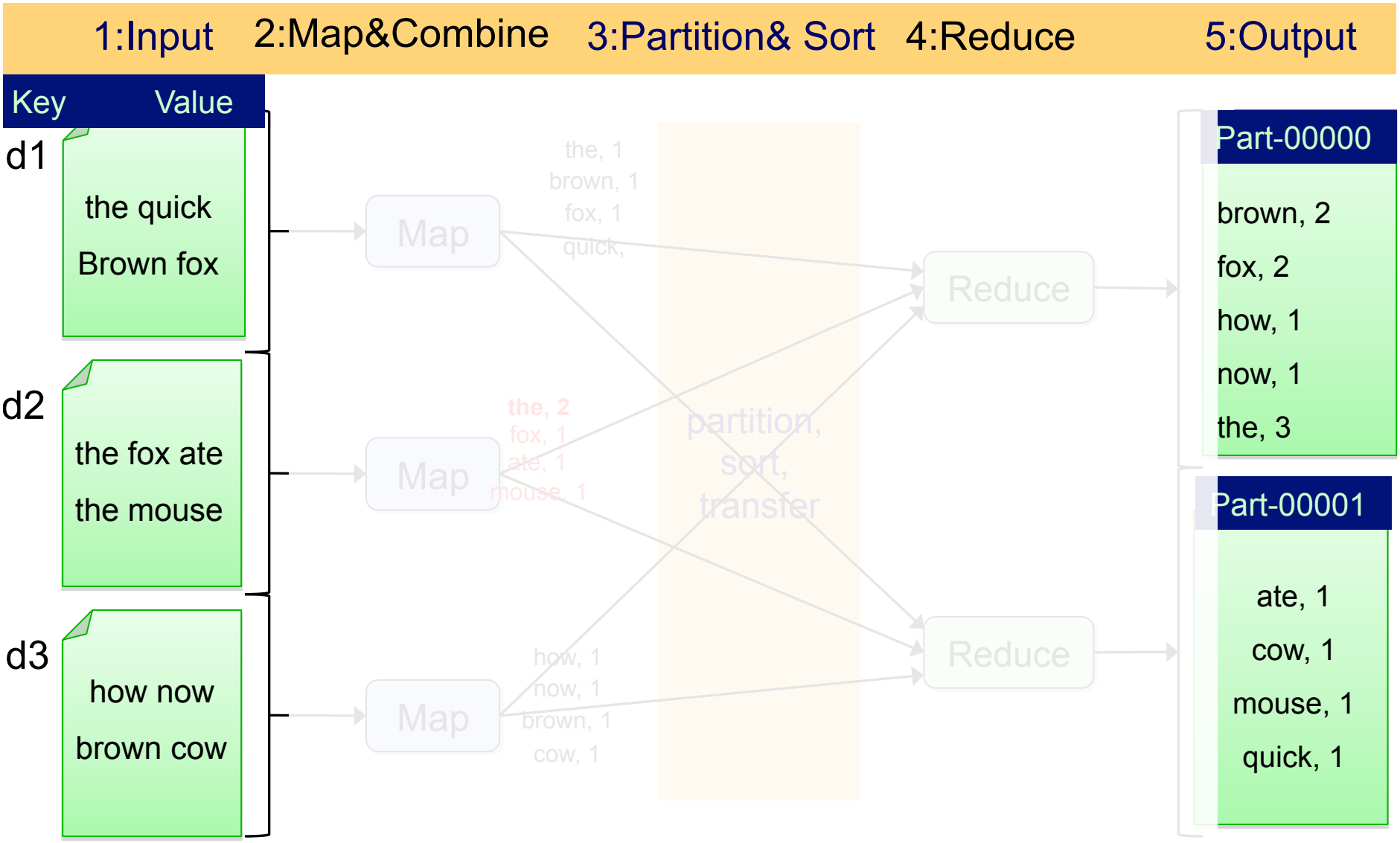


Reduce groups data with common keys

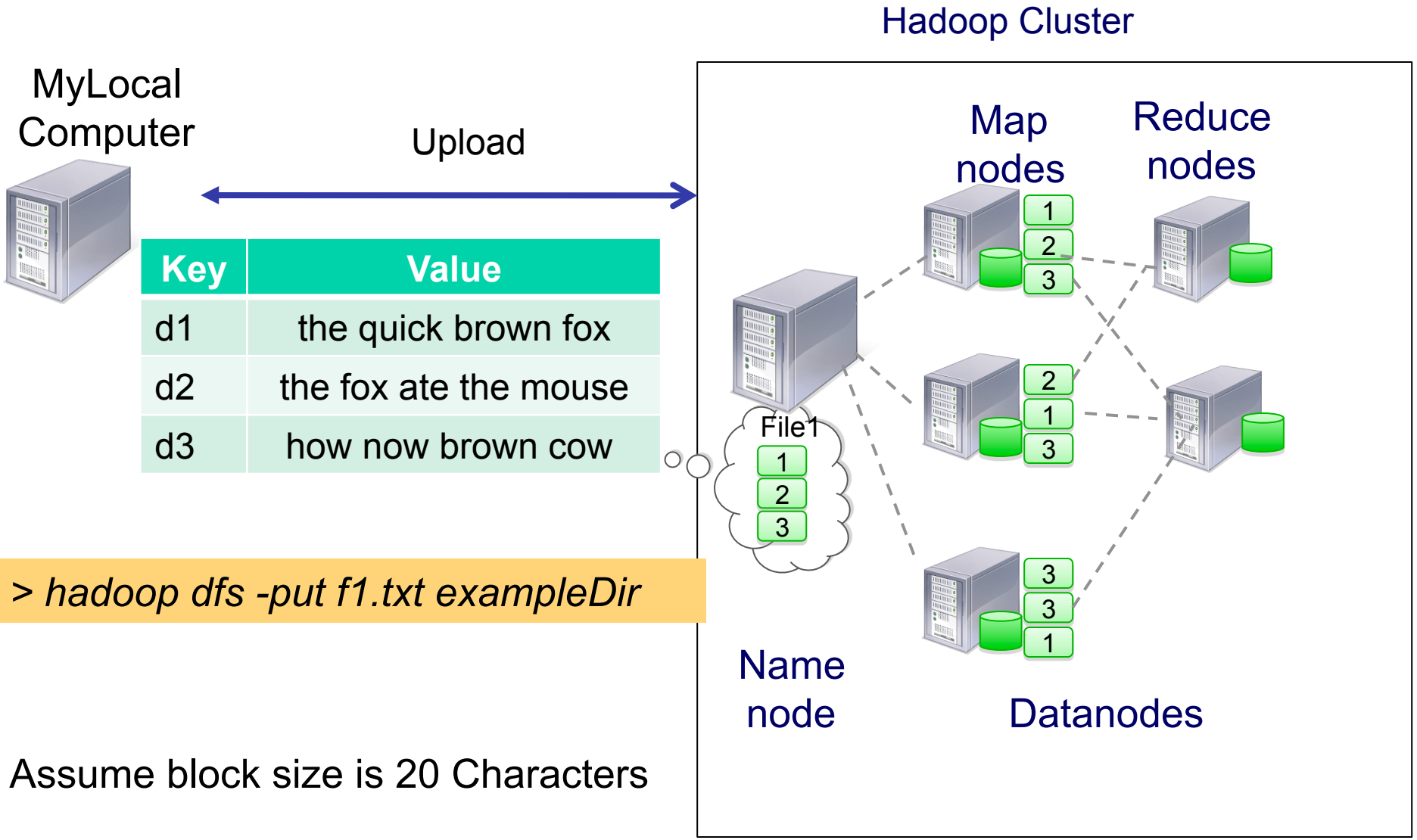
- ..



Word Count Workflow



Hadoop Cluster: 1 Name; 3 data node+2 task nodes

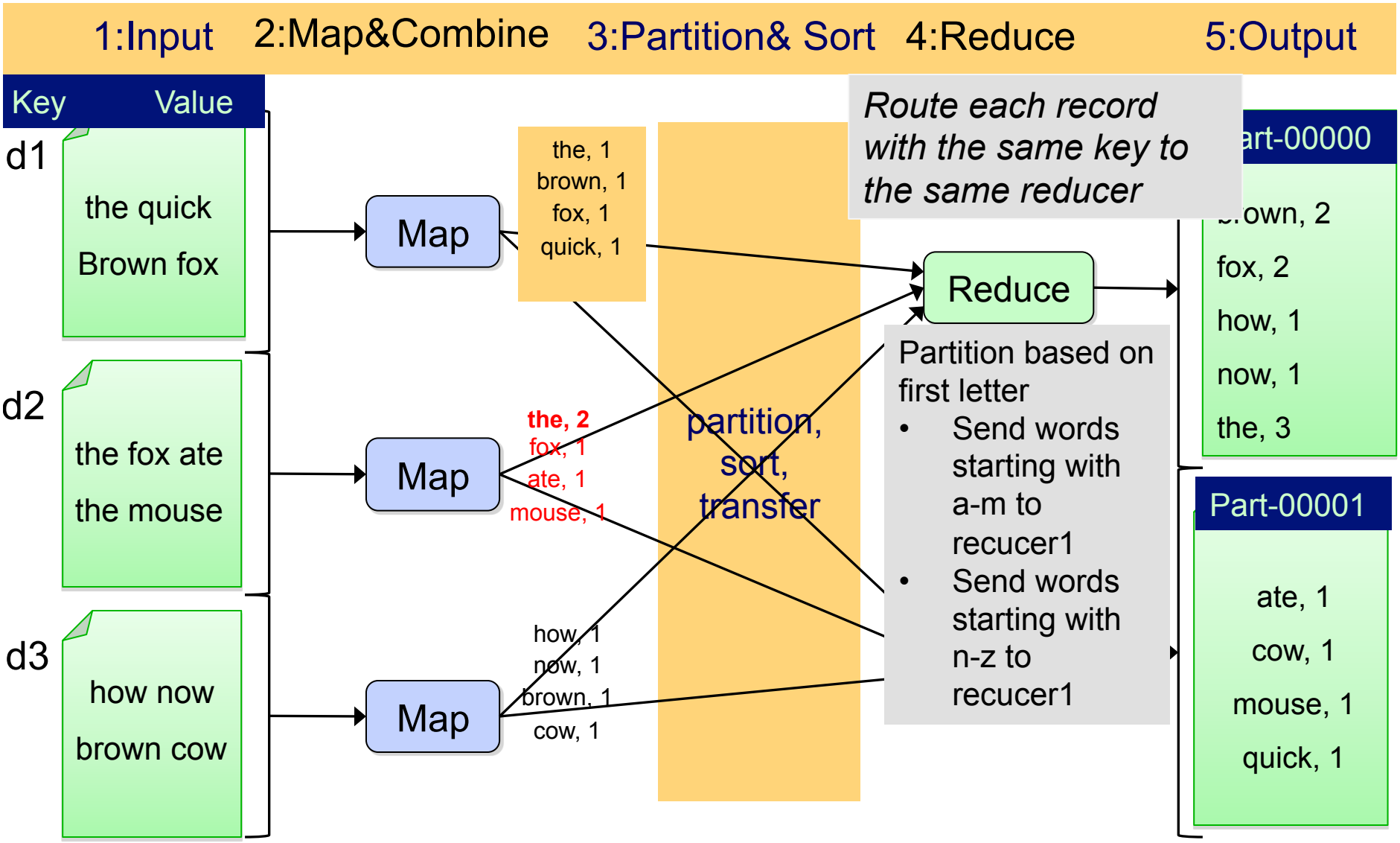


Key	Value
d1	the quick brown fox
d2	the fox ate the mouse
d3	how now brown cow

> hadoop dfs -put f1.txt exampleDir

Assume block size is 20 Characters

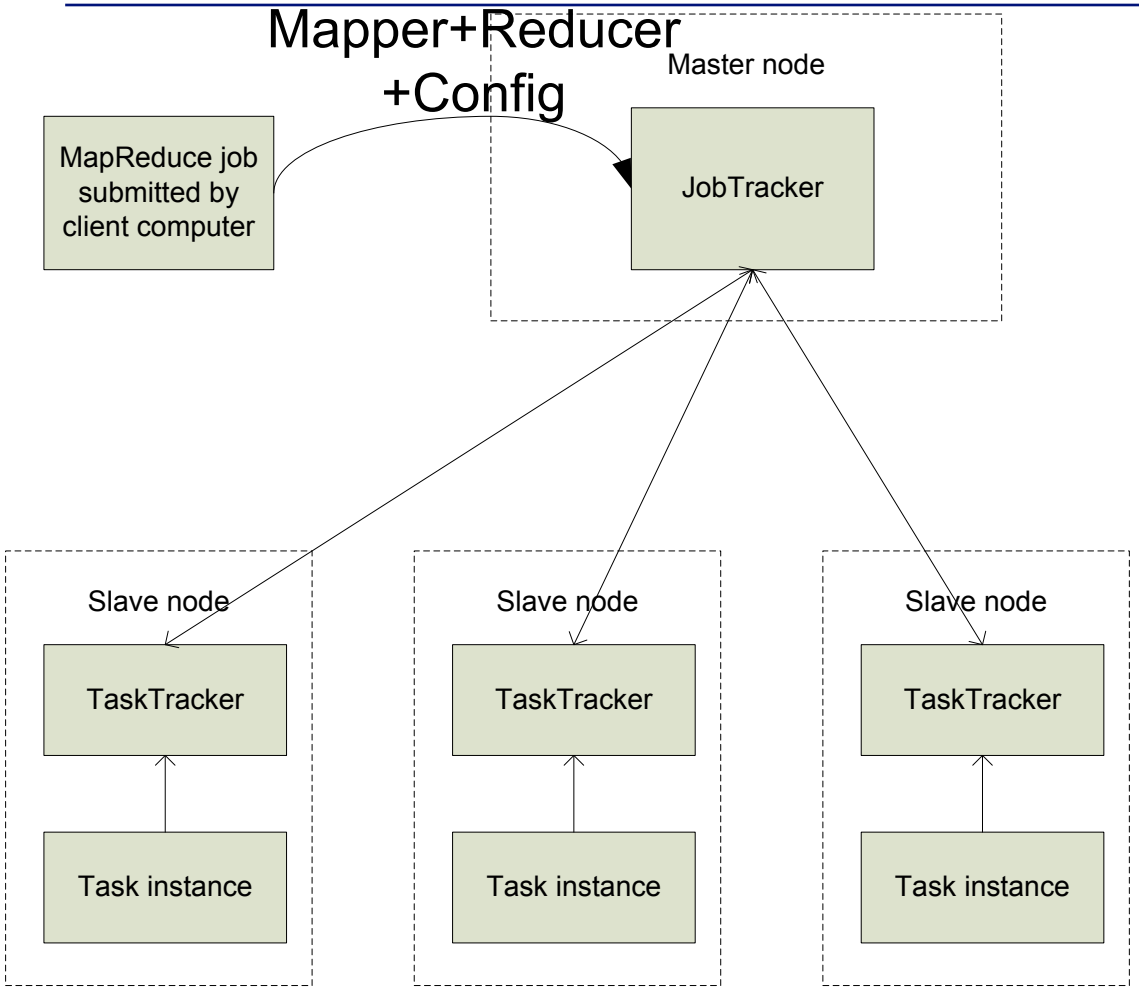
Word Count Workflow

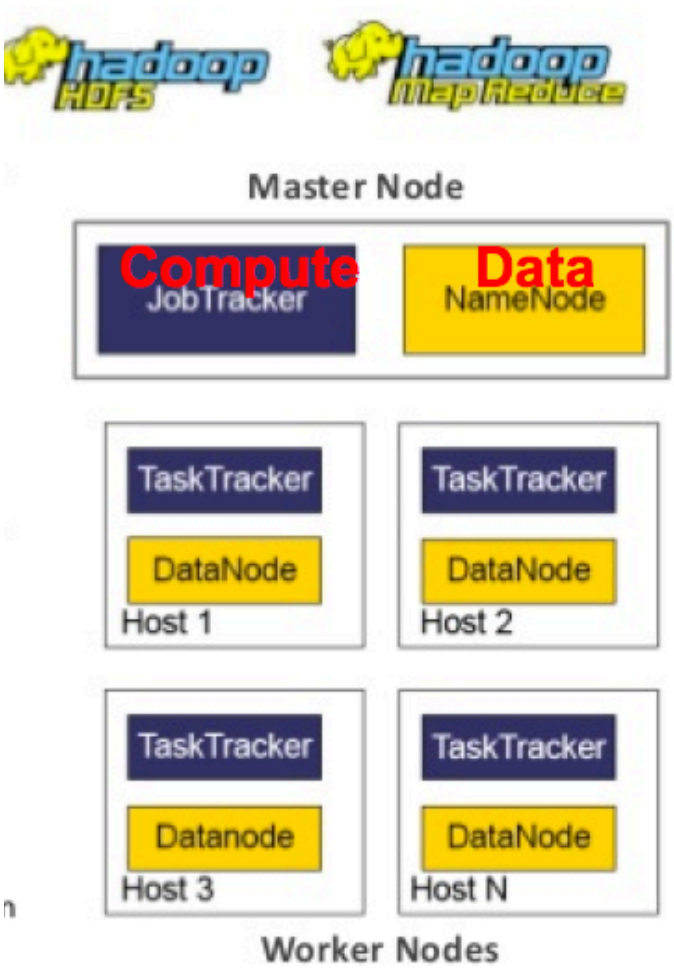


Execution on Clusters

1. Input files split (*M* splits)
2. Assign Master & Workers
3. Map tasks
4. Writing intermediate data to disk (*R* regions)
5. Intermediate data read & sort
6. Reduce tasks
7. Return partition files

MapReduce: High Level





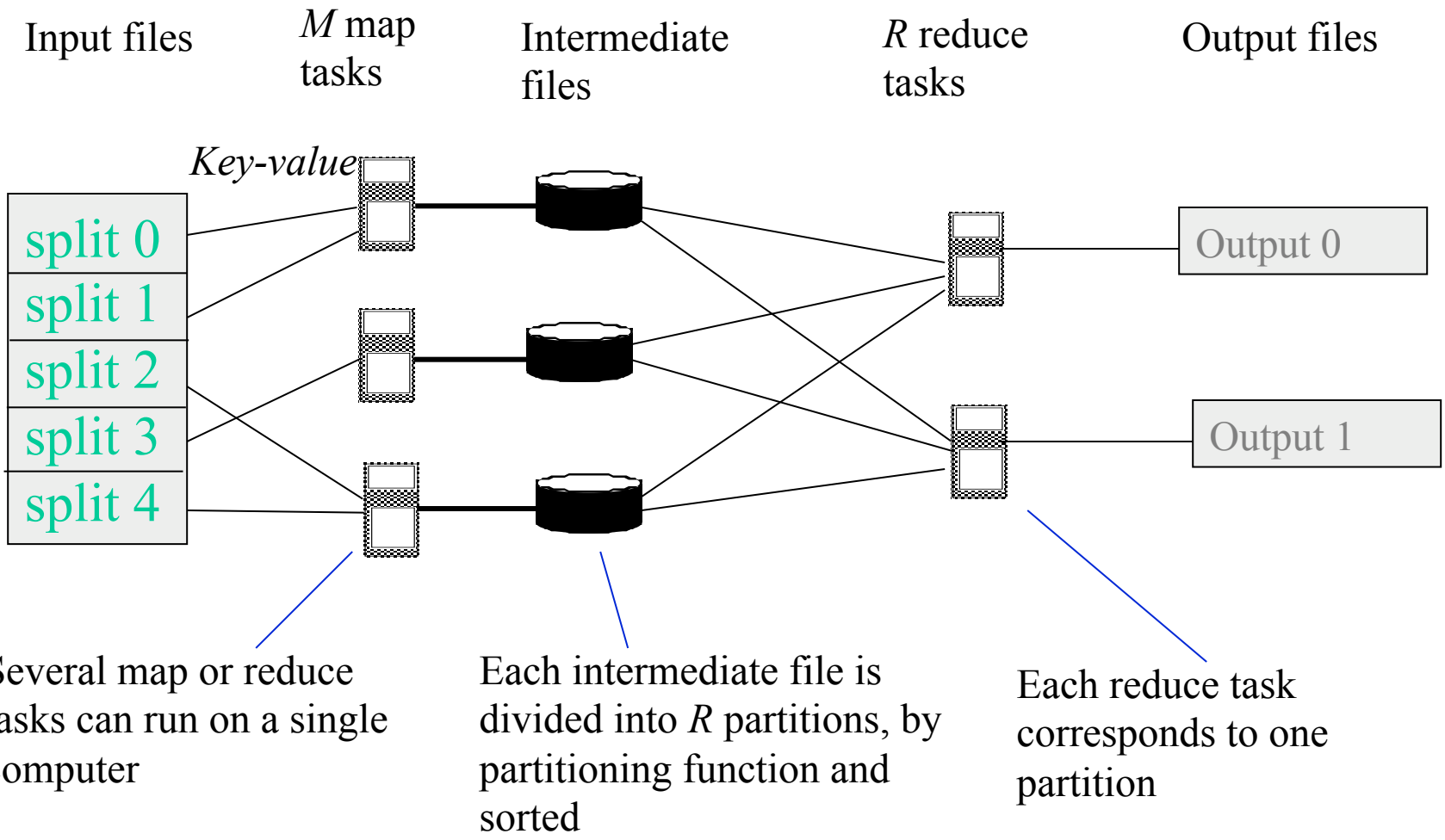
One instance of your Mapper is initialized by the *MapTaskRunner* for a *input block*

Exists in separate process from all other instances of Mapper – no data sharing!

MapReduce a framework for big data

- **MapReduce codifies a generic recipe for processing large datasets that consists of two stages.**
  - In the first stage, a user-specified computation is applied over all input records in a dataset.
  - These operations occur in parallel and yield intermediate output that is then aggregated by another user-specified computation.
- **Programmer and execution framework synergy**
- **Just provide the mapper and reducer functions**
  - The programmer defines these two types of computations, and the execution framework coordinates the actual processing (very loosely, MapReduce provides a functional abstraction).
- **Very powerful: many interesting algorithms can be expressed quite concisely**
  - Although such a two-stage processing structure may appear to be very restrictive, many interesting algorithms can be expressed quite concisely, especially if one decomposes complex algorithms into a sequence of MapReduce jobs

In summary: Map/Reduce Cluster



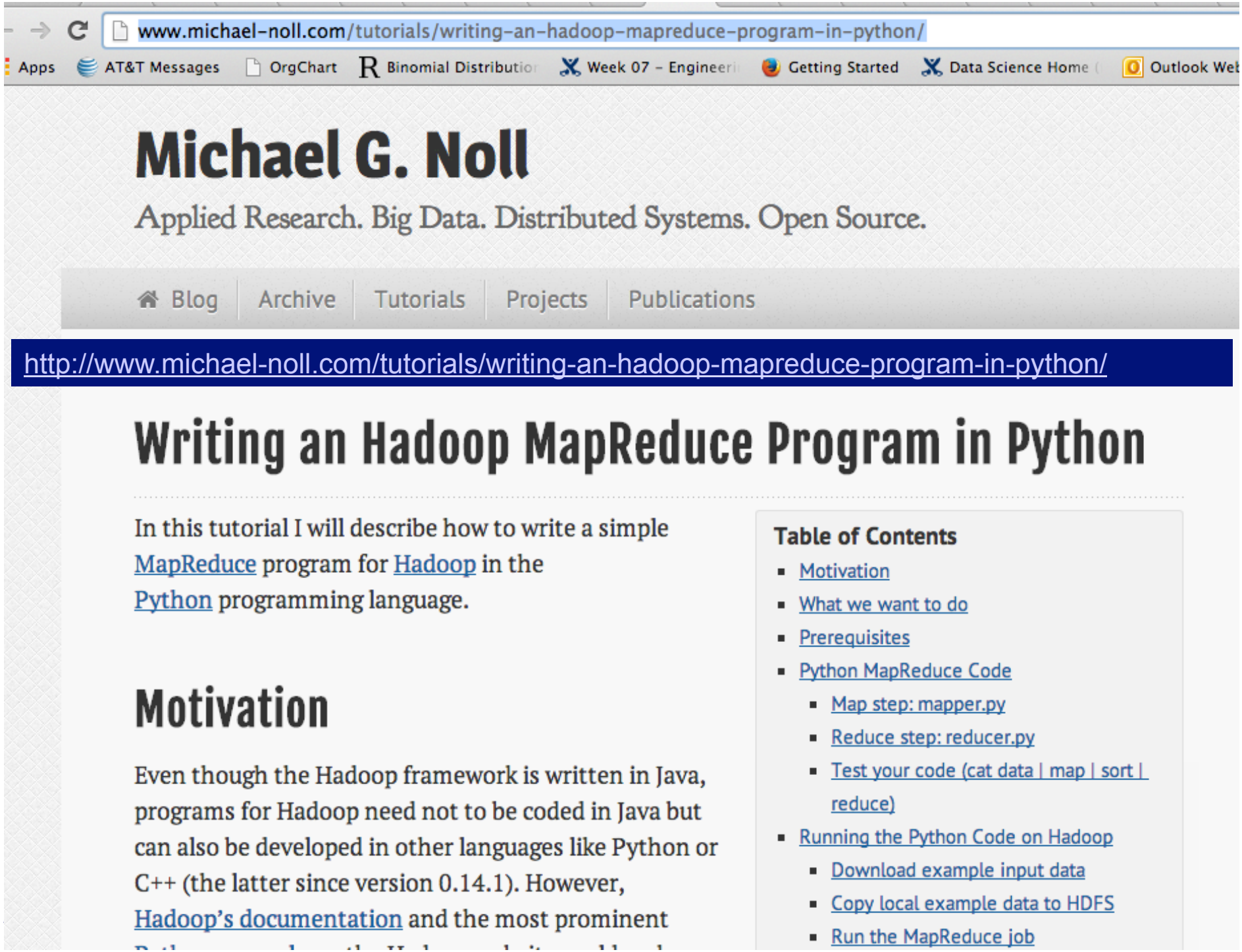
In summary: Data Records flow from Map to Reduce

- **Framework will convert each record of input into a key/value pair**
  - The framework will convert each record of input into a key/value pair, and each pair will be input to the map function once.
- **The map output pairs are grouped and sorted by key.**
  - The map output is a set of key/value pairs—nominally one pair that is the transformed input pair, but it is perfectly acceptable to output multiple pairs.
- **The reduce function is called one time for each key, in sort sequence, with the key and the set of values that share that key.**
- **Reduce outputs to file**
  - The reduce method may output an arbitrary number of key/value pairs, which are written to the output files in the job output directory.
  - *If the reduce output keys are unchanged from the reduce input keys, the final output will be sorted.*

Full Example: Word Count

- **Full code Examples (20)**
  - WordCount example on local machine (10)

http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/


www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
Apps
AT&T Messages
OrgChart
Binomial Distributio
Week 07 - Engineeri
Getting Started
Data Science Home
Outlook Wet
Michael G. Noll
Applied Research. Big Data. Distributed Systems. Open Source.
Blog
Archive
Tutorials
Projects
Publications
http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
Writing an Hadoop MapReduce Program in Python
In this tutorial I will describe how to write a simple MapReduce program for Hadoop in the Python programming language.
Table of Contents
Motivation
What we want to do
Prerequisites
Python MapReduce Code
Map step: mapper.py
Reduce step: reducer.py
Test your code (cat data | map | sort | reduce)
Running the Python Code on Hadoop
Download example input data
Copy local example data to HDFS
Run the MapReduce job
Motivation
Even though the Hadoop framework is written in Java, programs for Hadoop need not to be coded in Java but can also be developed in other languages like Python or C++ (the latter since version 0.14.1). However, Hadoop's documentation and the most prominent

Installing Hadoop

- **Mac:**
  - http://amodernstory.com/2014/09/23/installing-hadoop-on-mac-osx-yosemite/
  - This link is for haddoop 2.6. I follow the instructions and easily get hadoop installed.

- **Windows:**
  - Hadoop 1.0
  - http://saphanatutorial.com/hadoop-installation-on-windows-7-using-cygwin/

  - Hadoop 2.0 (Hortonworks Data Platform 2.0 for Windows)
  - http://hortonworks.com/blog/install-hadoop-windows-hortonworks-data-platform-2-0/

- **Linux (Ubuntu):**
  - http://www.bogotobogo.com/Hadoop/BigData_hadoop_Install_on_ubuntu_single_node_cluster.php

On Mac install HomeBrew

- **Install HomeBrew**
  - Download it from the website at http://brew.sh/ or simply paste the script inside the terminal
  - $ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

On Windows install CygWin

- **Cygwin is:**
  - a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows.

Current Cygwin DLL version

The most recent version of the Cygwin DLL is 2.2.1. Install it by running setup-x86.exe (32-bit installation) or setup-x86_64.exe (64-bit installation).

Use the setup program to perform a fresh install or to update an existing installation.

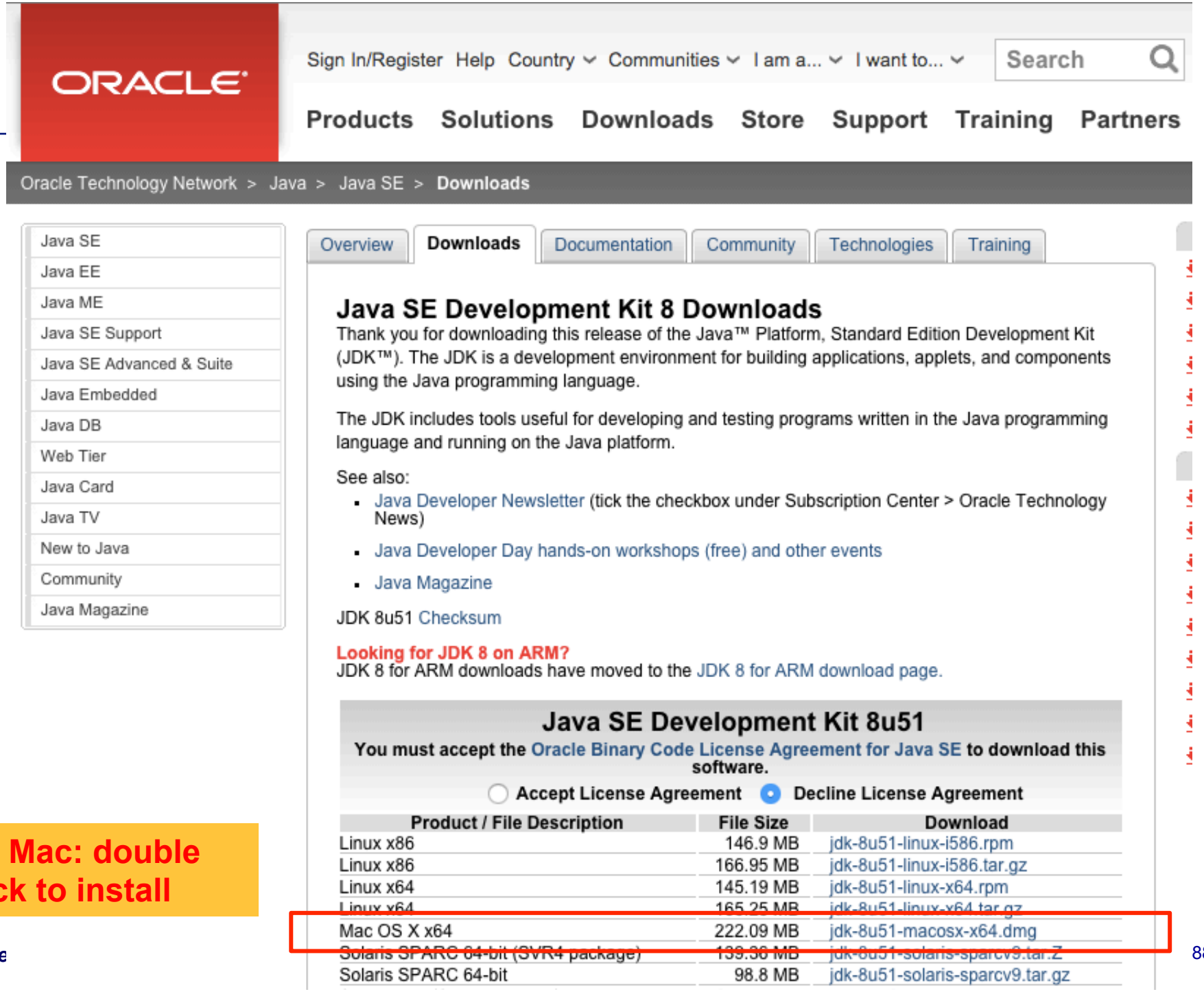
Note that individual packages in the distribution are updated separately from the DLL so the Cygwin DLL version is not useful as a general Cygwin release number.

Make sure Java JDK is installed

- **Click here:** http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

- **On windows machine set up $JAVA_HOME**

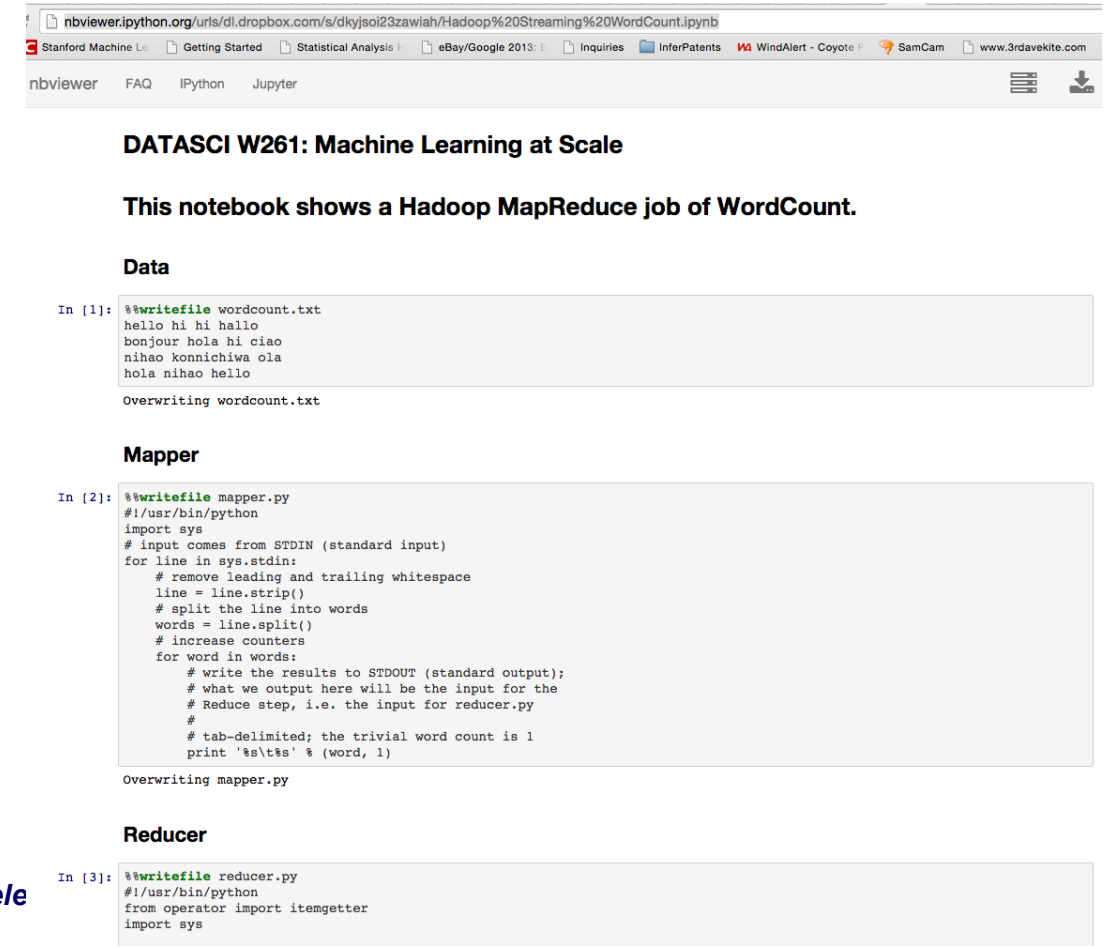
- **Right-click the My Computer icon on your desktop and select Properties**
- **Click the Advanced tab**
- **Click the Environment Variables button**
- **Under System Variables, click New**
- **Enter the variable name as JAVA_HOME**
- **Enter the variable value as the installation path for the Java Development Kit**



On Mac: double click to install

Word Count Notebook

http://nbviewer.ipython.org/urls/dl.dropbox.com/s/dkyjsoi23zawiah/Hadoop%20Streaming%20WordCount.ipynb



nbviewer.ipython.org/urls/dl.dropbox.com/s/dkyjsoi23zawiah/Hadoop%20Streaming%20WordCount.ipynb

Stanford Machine Le | Getting Started | Statistical Analysis | eBay/Google 2013: E | Inquiries | InferPatents | WindAlert - Coyote | SamCam | www.3rdavekite.com

nbviewer FAQ IPython Jupyter

DATASCI W261: Machine Learning at Scale

This notebook shows a Hadoop MapReduce job of WordCount.

Data

In [1]:
```
%%writefile wordcount.txt
hello hi hi hallo
bonjour hola hi ciao
nihao konnichiwa ola
hola nihao hello
```

```
Overwriting wordcount.txt
```

Mapper

In [2]:
```
%%writefile mapper.py
#!/usr/bin/python
import sys
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print '%s\t%s' % (word, 1)
```

```
Overwriting mapper.py
```

Reducer

In [3]:
```
%%writefile reducer.py
#!/usr/bin/python
from operator import itemgetter
import sys
```

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
- **HW1: review and grading**
- **Q&A (WK02)**
- **HW2:**
- **Naive Bayes**
  - Various Naïve Bayes Flavours
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

HW2

- **Repeat of HW1 modulo do everything in Hadoop and other minor tweaks**
  - Lift and shift mappers and reducers from HW1!

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
- **HW1: review and grading**
- **Q&A (WK02)**
- **HW2:**
- **Naive Bayes Continued**
  - Various Naïve Bayes Flavours
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case) **REVIEW Briefly**
  - Case Study: Spam detector in Naïve Bayes
- **Naïve Bayes (this session)**
  - Discrete input variables
  - Continuous input variables
  - Discrete Inputs: Bernoulli versus multinomial)

Live Session Outline

- **Welcome & Class Introductions**
  - Please mute you microphones
  - Start RECORDING (bonus points for reminding me!)
  - Class, homework, project Logistics + Office hours
- **HW1: review and grading**
- **Q&A (WK02)**
- **Naive Bayes**
  - Various Naïve Bayes Flavours
- **Wrapup**
  - Finish RECORDING (bonus points for reminding me!)
  - Click End Meeting

Derive NB Algorithm

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j)P(X_1, X_2, .., X_N | Y=y_j)}$$

$$= \frac{P(Y=y_k)\Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i | Y=y_j)}$$

- ..

The Naive Bayes algorithm is a classification algorithm based on Bayes rule, that assumes the attributes $X_1 \ldots X_n$ are all conditionally independent of one another, given Y. The value of this assumption is that it dramatically simplifies the representation of $P(X|Y)$, and the problem of estimating it from the training data. Consider, for example, the case where $X = \langle X_1, X_2 \rangle$. In this case

Independence
P(A|B)=P(A)
P(*A*, *B*)=P(*A*)P(*B*)
P(A|B, C)= P(A|C)

$$P(X|Y) = P(X_1, X_2|Y)$$
$$= P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

Use Product Rule $P(A, B) = P(A \mid B)P(B$

Naïve Bayes

Where the second line follows from a general property of probabilities (product Rule), and the third line follows directly from our above definition of conditional independence.

Naïve Bayes Classifier for Text

100 business docs = Pr(Business) =0.1+/- CI 2^N

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j)P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1

Y_2

$$= \frac{P(Y=y_k)\Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i | Y=y_j)}$$

Pr(X="corporation"|Class=Business) = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k)\Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

Conditional independence dramatically reduces model complexity: to 2n parameters from 2^n

- More generally, when X contains n attributes which are conditionally independent of one another given Y, we have

$$P(X_1 \ldots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \quad (1)$$

Notice that when Y and the X_i are boolean variables, we need only $2n$ parameters to define $P(X_i = x_{ik}|Y = y_j)$ for the necessary i, j, k. This is a dramatic reduction compared to the $2(2^n - 1)$ parameters needed to characterize $P(X|Y)$ if we make no conditional independence assumption.

Naïve Bayes

- A generative, parametric model
- Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.
- The standard naive Bayes classifier (at least the R implementation) assumes independence of the predictor variables, and Gaussian distribution (given the target class) of metric predictors.
- For attributes with missing values, the corresponding table entries are omitted for prediction.

Naïve Bayes and Conditional Independence

- **Make a conditional independence assumption; this leads to a Naïve Bayes classifier**
  - Reduces the number of parameters from 2n -1 * 2 parameters to 2n
- ***Definition: Given random variables X;Y and Z, we say X is conditionally* independent of *Y given Z, if and only if the probability* distribution governing *X is independent of the value of Y given Z;***
  - $(\forall i; j; k)\ P(X = x_i \mid Y = y_j, Z = z_k) = P(X = x_i \mid Z = z_k)$

Naïve Bayes Classifier for Text

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j)P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1

Y_2

$$= \frac{P(Y=y_k)\Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i | Y=y_j)}$$

Pr("corporation"|Class=Business) = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k)\Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

Probability Basics

- Prior, conditional and joint probability
  - Prior probability: $P(X)$
  - Conditional probability: $P(X_1 \mid X_2), P(X_2 \mid X_1)$
  - Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
  - Relationship: $P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) = P(X_1 \mid X_2)P(X_2)$
  - Independence: $P(X_2 \mid X_1) = P(X_2), P(X_1 \mid X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$
- Bayesian Rule

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \qquad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Probabilistic Classification

- Establishing a probabilistic model for classification
  - Discriminative model

$$P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

  - Generative model

$$P(\mathbf{X} \mid C) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

- MAP classification rule
  - MAP: Maximum A Posterior
  - Assign x to c^* if $P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*, \; c = c_1, \cdots, c_L$
- Generative classification with the MAP rule
  - Apply Bayesian rule to convert $P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \propto P(\mathbf{X} \mid C)P(C)$

Naive Bayes for Discrete-Valued Inputs

When the n input attributes X_i each take on J possible discrete values, and Y is a discrete variable taking on K possible values, then our learning task is to estimate two sets of parameters. The first is

$$\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k) \tag{4}$$

for each input attribute X_i, each of its possible values x_{ij}, and each of the possible values y_k of Y. Note there will be nJK such parameters, and note also that only $n(J-1)K$ of these are independent, given that they must satisfy $1 = \sum_j \theta_{ijk}$ for each pair of i, k values.

In addition, we must estimate parameters that define the prior probability over Y:

$$\pi_k \equiv P(Y = y_k) \tag{5}$$

Note there are K of these parameters, $(K-1)$ of which are independent.

We can estimate these parameters using either maximum likelihood estimates (based on calculating the relative frequencies of the different events in the data), or using Bayesian MAP estimates (augmenting this observed data with prior distributions over the values of these parameters).

Maximum likelihood estimates for θ_{ijk} given a set of training examples D are given by

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}} \quad (6)$$

where the $\#D\{x\}$ operator returns the number of elements in the set D that satisfy property x.

One danger of this maximum likelihood estimate is that it can sometimes result in θ estimates of zero, if the data does not happen to contain any training examples satisfying the condition in the numerator. To avoid this, it is common to use a "smoothed" estimate which effectively adds in a number of additional "hallucinated" examples, and which assumes these hallucinated examples are spread evenly over the possible values of X_i. This smoothed estimate is given by

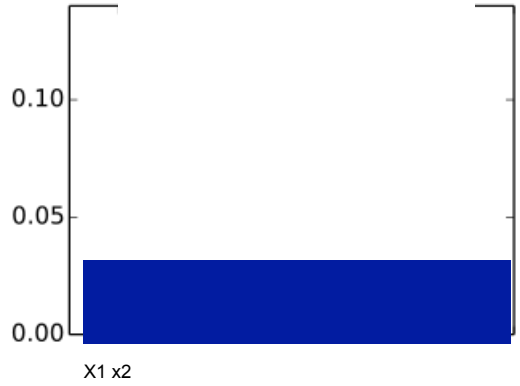
$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\} + l}{\#D\{Y = y_k\} + lJ} \quad (7)$$

where J is the number of distinct values X_i can take on, and l determines the strength of this smoothing (i.e., the number of hallucinated examples is lJ). This expression corresponds to a MAP estimate for θ_{ijk} if we assume a Dirichlet prior distribution over the θ_{ijk} parameters, with equal-valued parameters. If l is set to 1, this approach is called Laplace smoothing.

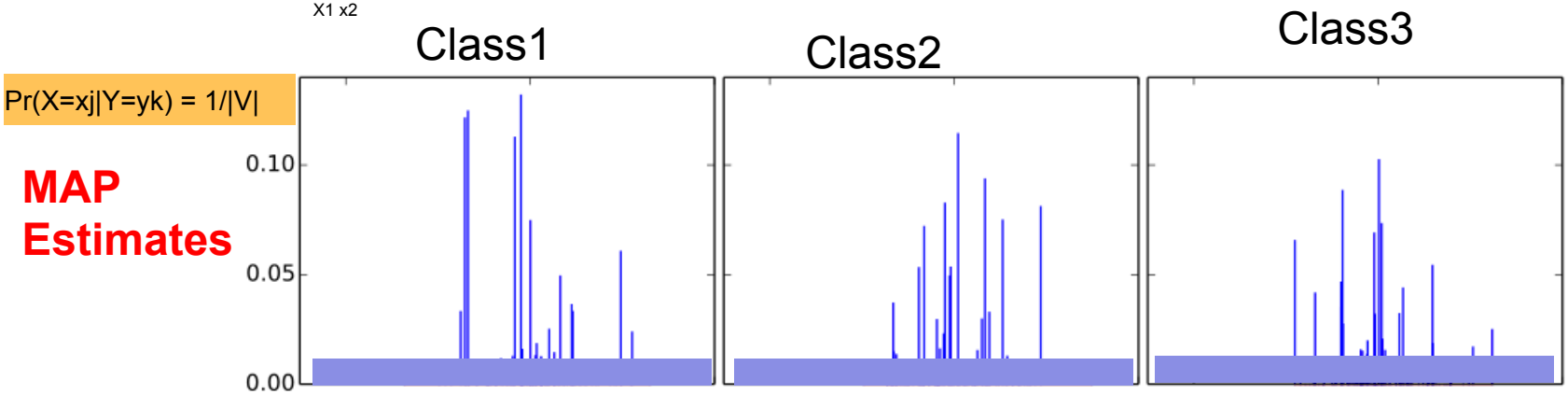
ML Estimates
Learning a NB Via Maximum Likelihood

MAP Estimates
Learning a NB Via Bayesian hierarchical model

MAP Estimates: Smoothed Probabilities

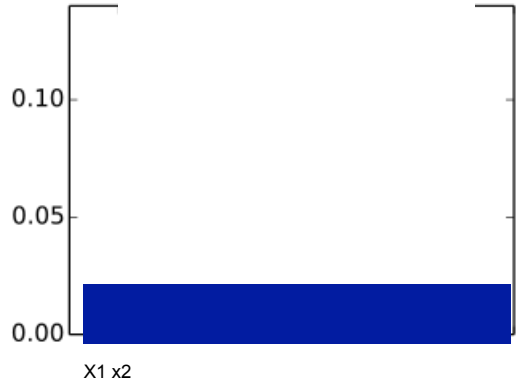


Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|
+
Training data



Huge Probability mass assigned to background model:
Bias versus variance?

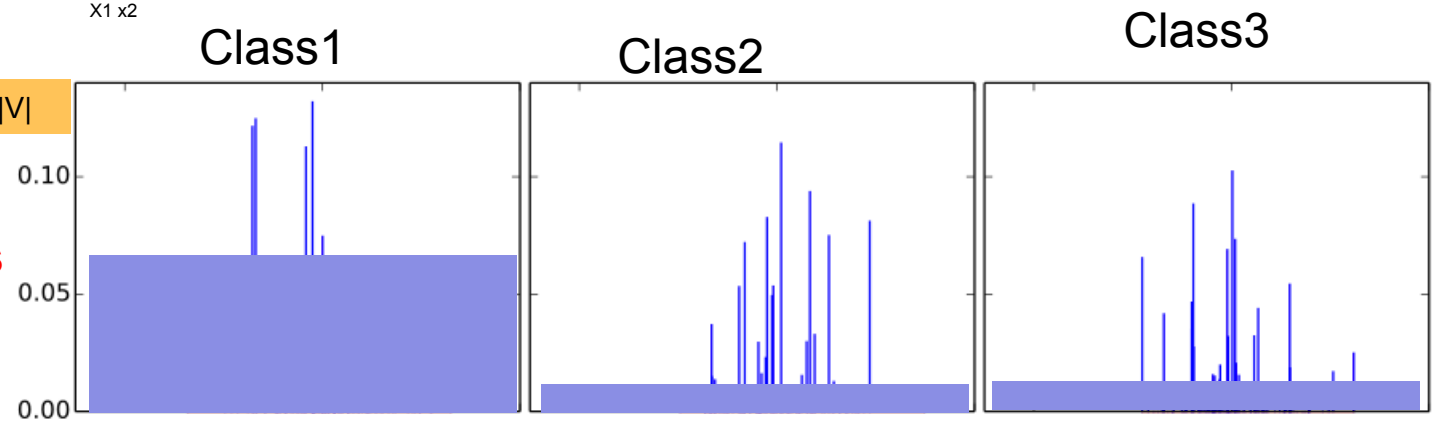
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

+

Training data

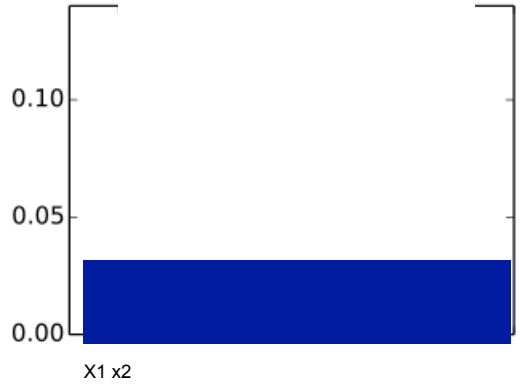


Pr(X=xj|Y=yk) = 1/|V|

MAP Estimates

Huge Probability mass assigned to background model:
Bias versus variance?

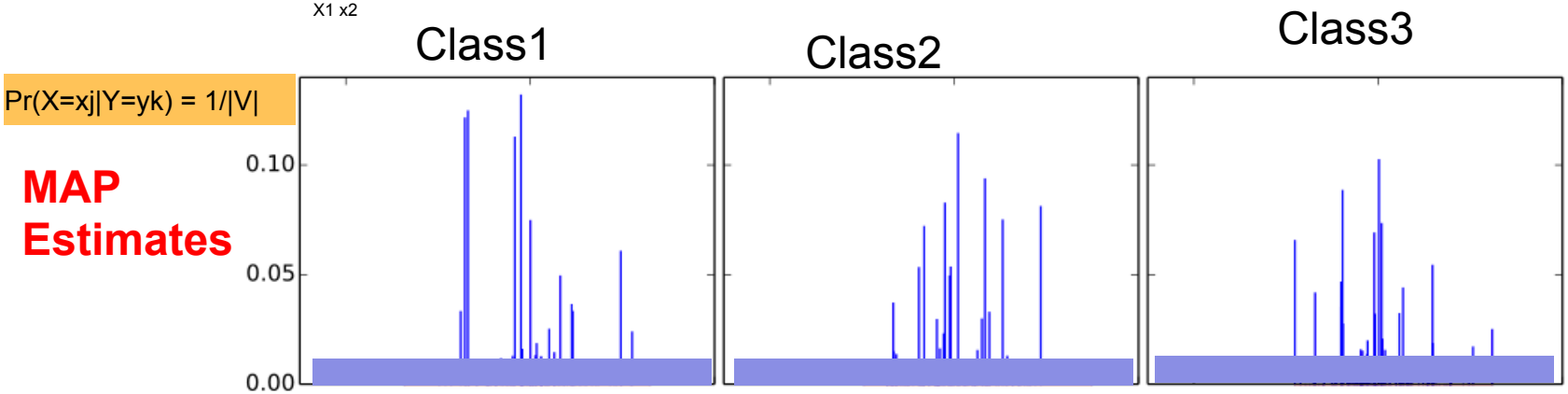
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

+

Training data



Huge Probability mass assigned to background model:
Bias versus variance? High Bias!

Class Prior Estimates

Maximum likelihood estimates for π_k are

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|} \tag{8}$$

where $|D|$ denotes the number of elements in the training set D.

Alternatively, we can obtain a smoothed estimate, or equivalently a MAP estimate based on a Dirichlet prior over the π_k parameters assuming equal priors on each π_k, by using the following expression

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + l}{|D| + lK} \tag{9}$$

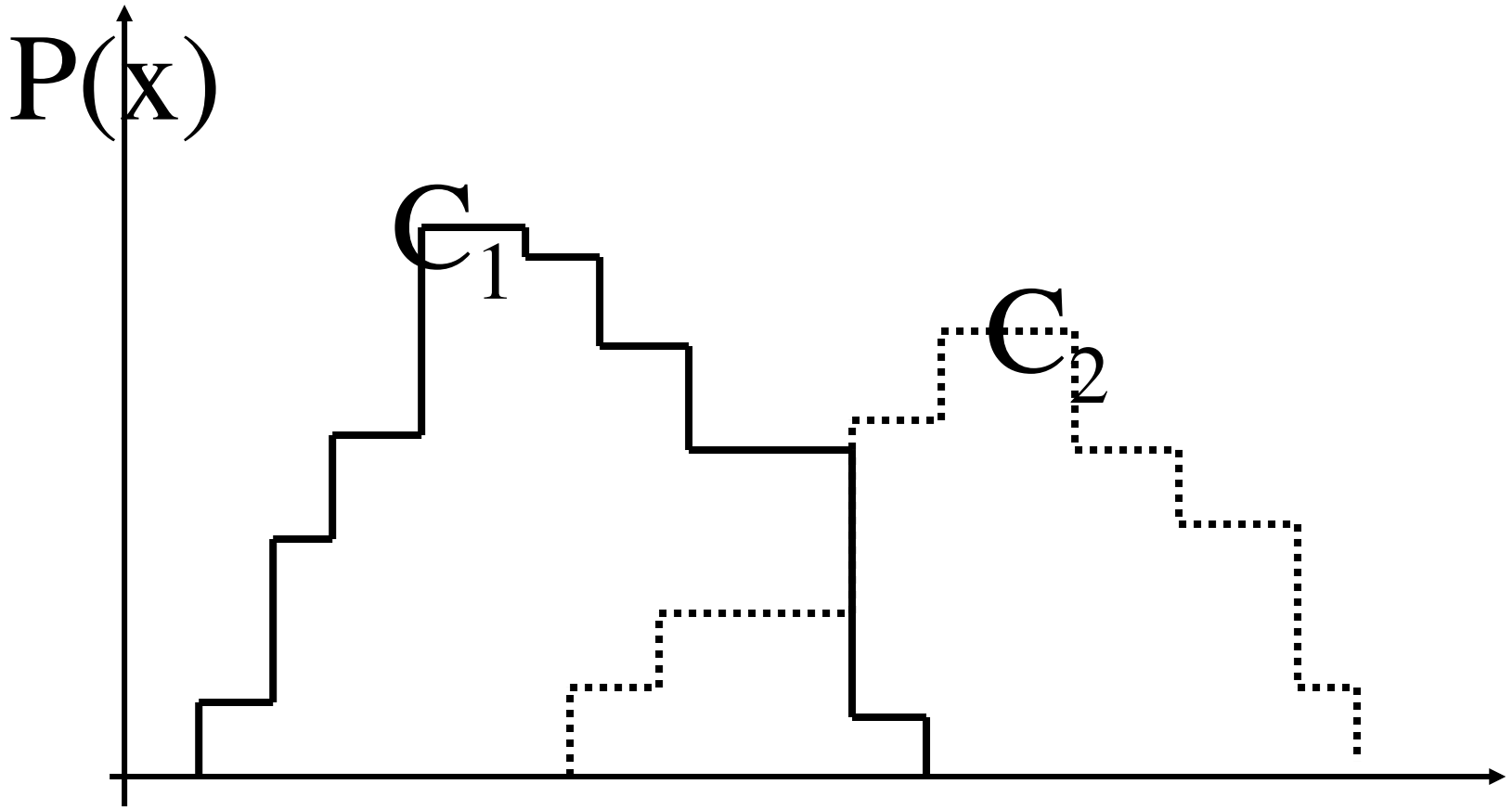
where K is the number of distinct values Y can take on, and l again determines the strength of the prior assumptions relative to the observed data D.

Naive Bayes for Continuous Inputs

- When the Xi are continuous we must choose some other way to represent the distributions
- P(Xi|Y)=Gaussian(Mu, Sigma).
- One common approach is to assume that for each possible discrete value yk of Y, the distribution of each continuous Xi is Gaussian, and is defined by a mean and standard deviation specific to Xi and yk.
- In order to train such a Naïve Bayes classifier we must therefore estimate the mean and standard deviation ians:

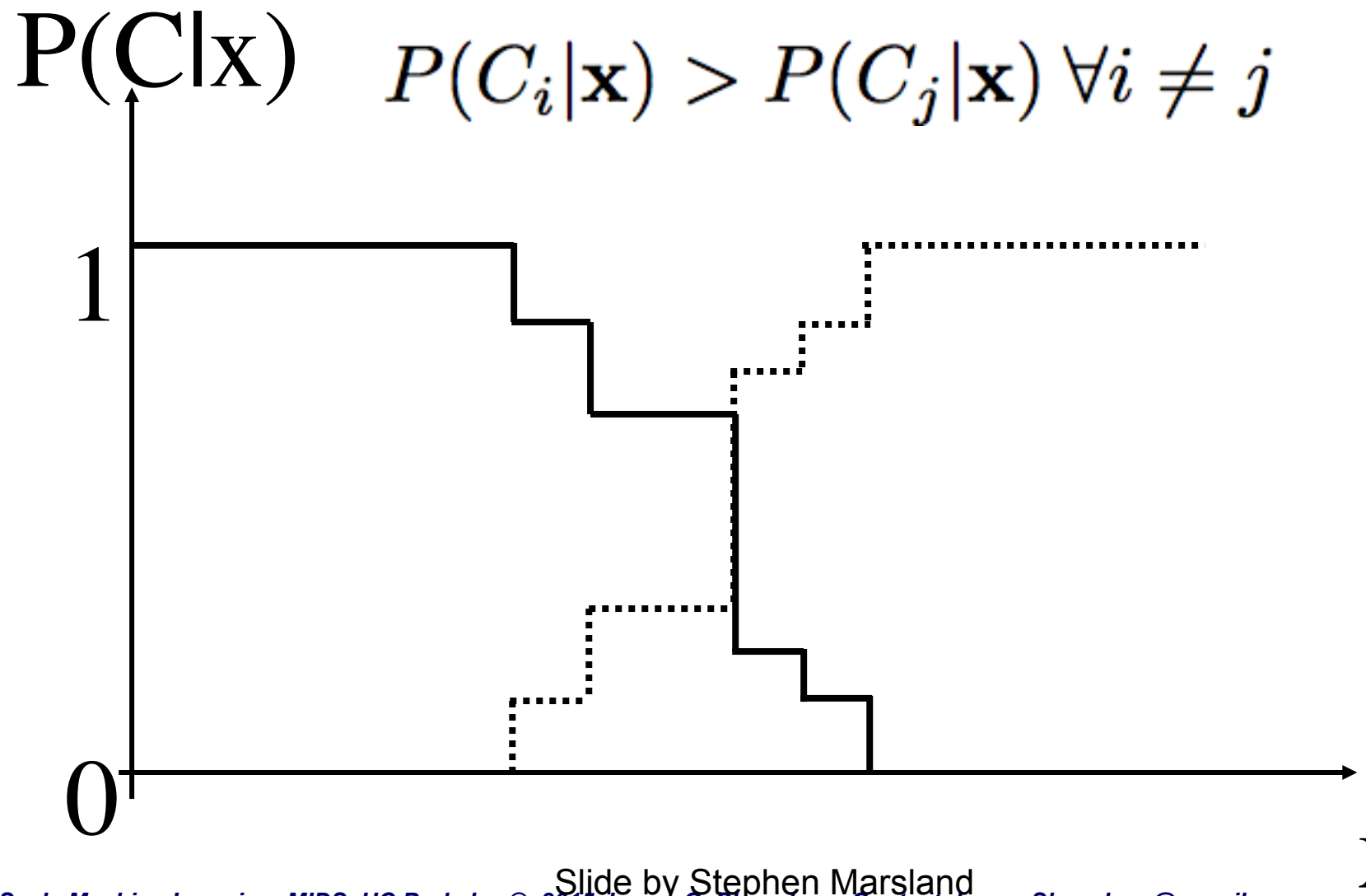
$$\mu_{ik} = E[X_i | Y = y_k]$$
$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 | Y = y_k]$$

Feature Histograms



Slide by Stephen Marsland

Posterior Probability



Slide by Stephen Marsland

MLE-based Estimates

Again, we can use either maximum likelihood estimates (MLE) or maximum a posteriori (MAP) estimates for these parameters. The maximum likelihood estimator for μ_{ik} is

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k) \tag{13}$$

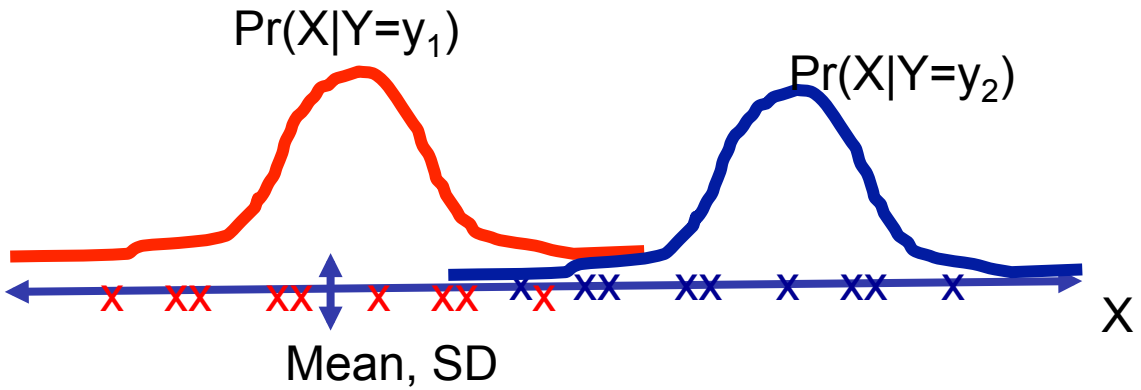
where the superscript j refers to the jth training example, and where $\delta(Y = y_k)$ is 1 if $Y = y_k$ and 0 otherwise. Note the role of δ here is to select only those training examples for which $Y = y_k$.

The maximum likelihood estimator for σ_{ik}^2 is

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k) \tag{14}$$

Estimate μ, σ from data

$$\mu_{ik} = E[X_i | Y = y_k]$$
$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 | Y = y_k]$$



Continuous Inputs: Pr(Y|X) ~ N(μ, σ)

If X is Normally distributed with mean μ and standard deviation σ, we write

$$\boxed{X \sim N(\mu, \sigma^2)}$$

μ and σ are the **parameters** of the distribution.

The probability density of the Normal distribution is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-(x-\mu)^2/2\sigma^2}$$

For the purposes of this course we do not need to use this expression. It is included here for future reference.

Naïve Bayes

- **Combine discrete input variables with continuous input variables?**
  - Naïve Bayes, Decision trees
- **Whereas these requires feature transformations**
  - Logistic regression: one hot-encoding

- **YES**

Continuous NB: Complexity

- **For each attribute Xi and each possible value yk of Y. Note there are $2nK$ of these parameters, all of which must be estimated independently.**
  - K classes and n continuous input variables

- **End of live session**