CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Day 15
Midterm Exam Discussion

Announcements and Feedback

- Midterm grading is underway
  - Hoping to post grades by the end of the week
- Phase 1 grading will follow after that
- Phase 2 due in two weeks

Question 1 - Algorithms [Lec 4,5,6]

a. How can we help ensure that our supervised learning models are not overfit?

b. In linear regression, what does the error term, ε, capture? What about R^2?

c. Name the four parameters we need to define in order to fully specify and evaluate a K-NN model for a given dataset.

Evaluating Our Model

- How can we be certain our model is good?
- Many solvers will compute a few heuristics to help
  - R^2 captures the amount of the variance explained by our model
    - High R^2 means we've captured most of the variance
  - p-values captures the likelihood that our coefficients are "unimportant"
    - Low p-values means our coefficients are likely significant
- We can also cross validate ourselves!
  - Divide the data into training data and test data.
  - Fit the model on the training data to find β and ϵ
  - Calculate mean squared error on the test data and see if it's consistent

Capturing Variability

- Add in an error term, ϵ: $y = \beta_0 + \beta_1 x + \epsilon$
  - Referred to as *noise*
  - Represents relationships you have not accounted for
  - This term captures the difference between our observations, and the *true* regression line

Remember, our data is just a trace of the real world. It is incomplete. It has uncertainty. We can only estimate the true regression line. Noise attempts to capture this fact.

The Basic Process

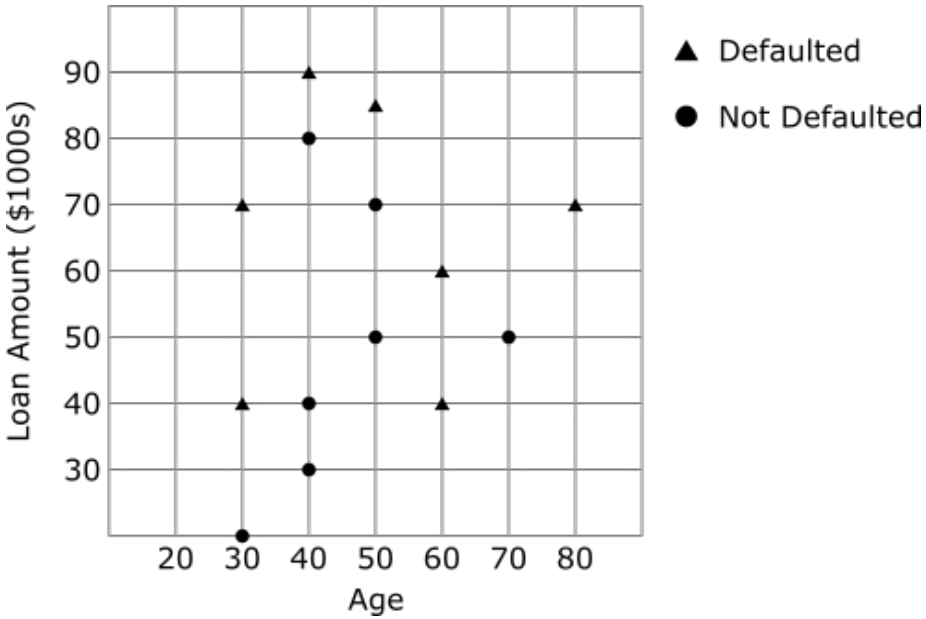
1. Decide on your ***similarity* metric**
2. Split the labeled set into training and test data
3. Pick an **evaluation metric** (similar to R^2 and p-values for linear reg)
4. Run with a few different values of k, check against evaluation metric
5. **Select k** with the best evaluation metric
6. Run on unlabeled data

Numerical Distance and Scale

- If our data is numerical in nature, there are a number of known ways to define "distance" between two things
  - Euclidian, Cosine, Manhattan, Mahalanobis, etc
- **What about scale**?
  - Consider clustering people based on salary and SAT scores:
    - The distance between ($30,000, 1400) and ($100,000, 1450) is dominated by the salary difference
    - Rescaling data, ie (30, 1400) and (100, 1450) balances the effect of each parameter…but is that necessarily the goal?

How you scale your data can have a significant impact on outcome, and therefore is also part of your model!

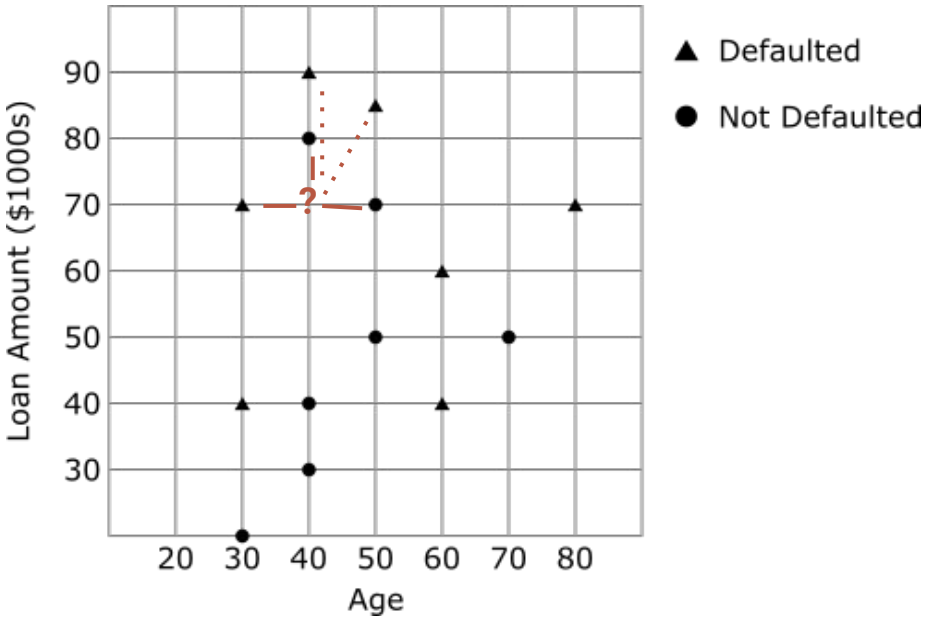
Question 1 - Algorithms [Lec 4,5,6]



d. If we assume euclidean distance determines the similarity of two points, does K-NN predict that a 40 year old client with a loan of $70k will default for k=3? What about for k=5?

e. Explain what would happen if we give the loan amount in terms of dollars instead of thousands of dollars. Based on that explanation, what would our model predict for the same client from part (d) with k=3?

Question 1 - Algorithms [Lec 4,5,6]

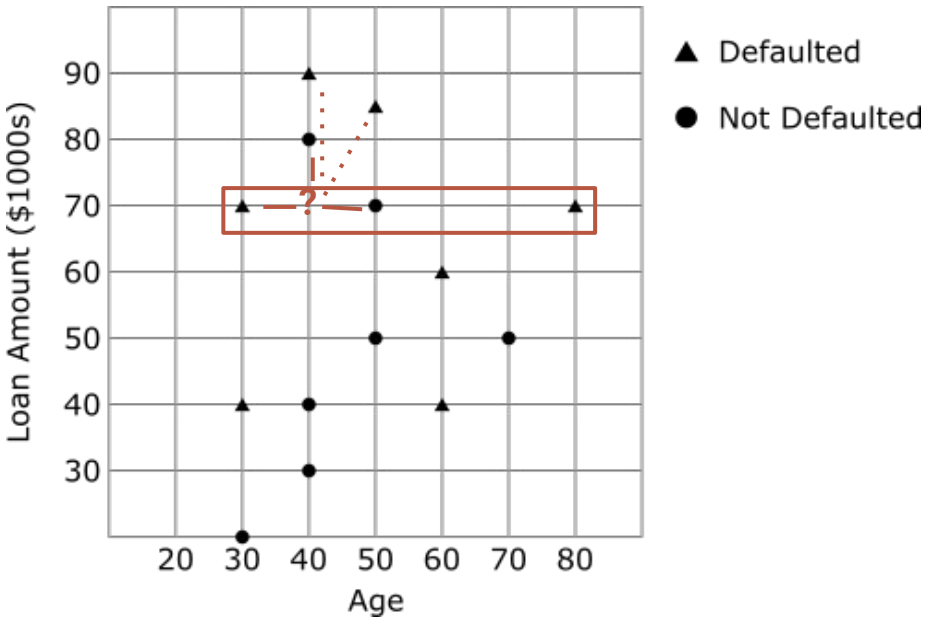


d. If we assume euclidean distance determines the similarity of two points, does K-NN predict that a 40 year old client with a loan of $70k will default for k=3? What about for k=5?

k=3 Not defaulted, k=5 defaulted

e. Explain what would happen if we give the loan amount in terms of dollars instead of thousands of dollars. Based on that explanation, what would our model predict for the same client from part (d) with k=3?

Question 1 - Algorithms [Lec 4,5,6]



d. If we assume euclidean distance determines the similarity of two points, does K-NN predict that a 40 year old client with a loan of $70k will default for k=3? What about for k=5?

k=3 Not defaulted, k=5 defaulted

e. Explain what would happen if we give the loan amount in terms of dollars instead of thousands of dollars. Based on that explanation, what would our model predict for the same client from part (d) with k=3?

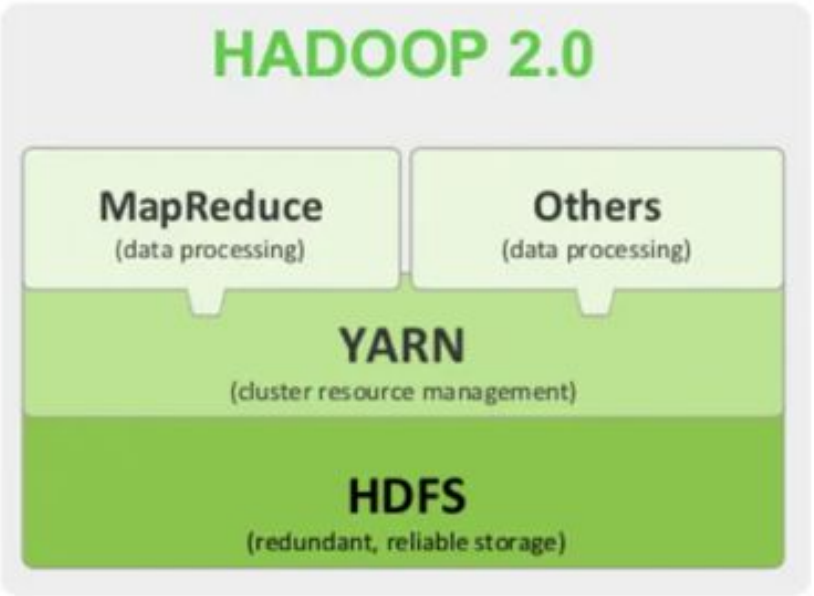
If loan is now in dollars, the loan amount will dominate in the distance calculation. The 3 nearest points to our target are now the three with $70,000 loans, so Defaulted

Question 2 - HDFS [Lec 7,8]

a. List two major differences between Hadoop1.x and Hadoop2.x versions.
b. How is an HDFS block replicated? Where are map and reduce tasks executed?
c. In HDFS, what is a (i) heartbeat (ii) BlockReport? Explain.
d. List two functions of a NameNode. List two functions of a DataNode.
e. What is the primary data type of the MapReduce model? Why are Maps able to run in parallel over the data?

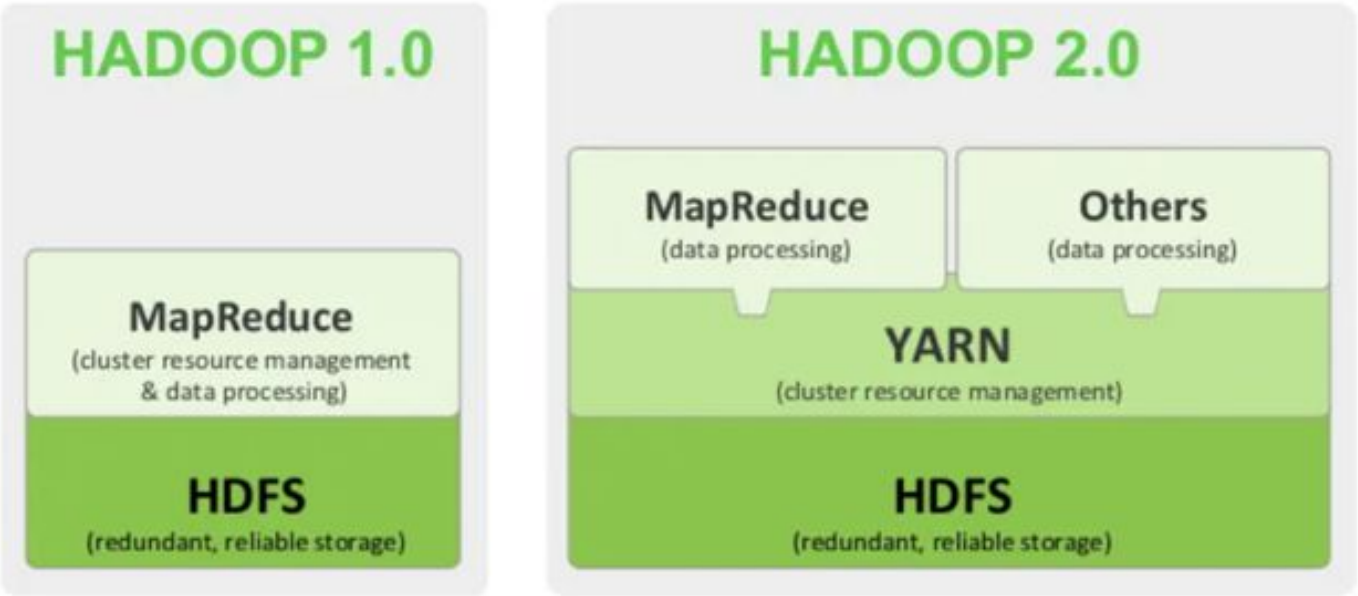
Evolution of Hadoop

Originally, MapReduce was the only supported software system, and also had to handle resource management (via JobTracker and TaskTracker)



HDFS (discussed last lecture) provides the reliable distributed file system as the backbone of Hadoop.

Evolution of Hadoop



In Hadoop 2.0, **JobTracker and TaskTracker were replaced by YARN** (yet another resource negotiator). MapReduce was now only responsible for data processing, **and other data processing software could be supported**.

HDFS (discussed last lecture) provides the reliable distributed file system as the backbone of Hadoop.

Replica Placement

Rack Aware Placement

- Goal: Improve reliability, availability, and network bandwidth utilization
- NameNode determines the rack id for each DataNode
- Replicas are typically placed on unique racks
  - Simple scheme but non-optimal
  - Writes are expensive
  - Typical replication factor is 3
- **Improvement: Place one on a node in the local rack, one on a node in a remote rack, and another on a different node in that same remote rack.**
- For a file this means ⅓ of the replicas on one node, ⅔ on one rack, and the other ⅓ distributed across all other racks. *(not an even distribution)*

Big Ideas

- **Failures are the norm — not an exception**
  - Typical MTBF for commodity components of 1000 days — if you have 1000s in your cluster, probability of at least 1 being down at any time nears 100%
- **Move "Processing" to the Data:** Co-locate processing of the data with the data itself rather than sending data around as in HPC.
- **Process Data Sequentially vs Random Access:** Do mass analytics on large sequential build data as opposed to search for individual items

Data Replication

- HDFS is designed to store very large files across machines in a cluster
- Each file is a sequence of blocks
  - All blocks in a file are the same size (except the last block)
  - Blocks are replicated for fault tolerance
  - Block size and replica are configurable per file
- The NameNode receives a **heartbeat** and **BlockReport** from each DataNode
  - **This report contains information about all the blocks on the DataNode**

DataNode Failure and Heartbeat

- **A crashed DataNode or a network partition can cause a subset of DataNodes to lose connectivity with the NameNode**
- **NameNode detects this by the absence of a heartbeat**
  - NameNode marks these DataNodes, and does not send requests to them
  - Data registered to the failed DataNode is not available to the HDFS
  - Death of a DataNode may cause some blocks to require more replication

Architecture: NameNode and DataNodes

- HDFS clusters consist of a single **NameNode** and multiple **DataNodes**

NameNode

A master server that manages the filesystem namespace, tracks metadata, and regulates client access to files.

DataNodes

Usually one per node in a cluster.

Manages storage attached to their node.

Serves read/write requests, file creation/deletion, and replication.

MapReduce Basics

Fundamental Concept: key-value pairs

- Key-value pairs form the basic structure of MapReduce
- Keys can be anything from simple data types to custom types
- Examples:

<docid, doc>

<yourName, yourLifeHistory>

<graphNode, nodeCharacteristics>

<geneNum, {pathway, geneExp, proteins}>

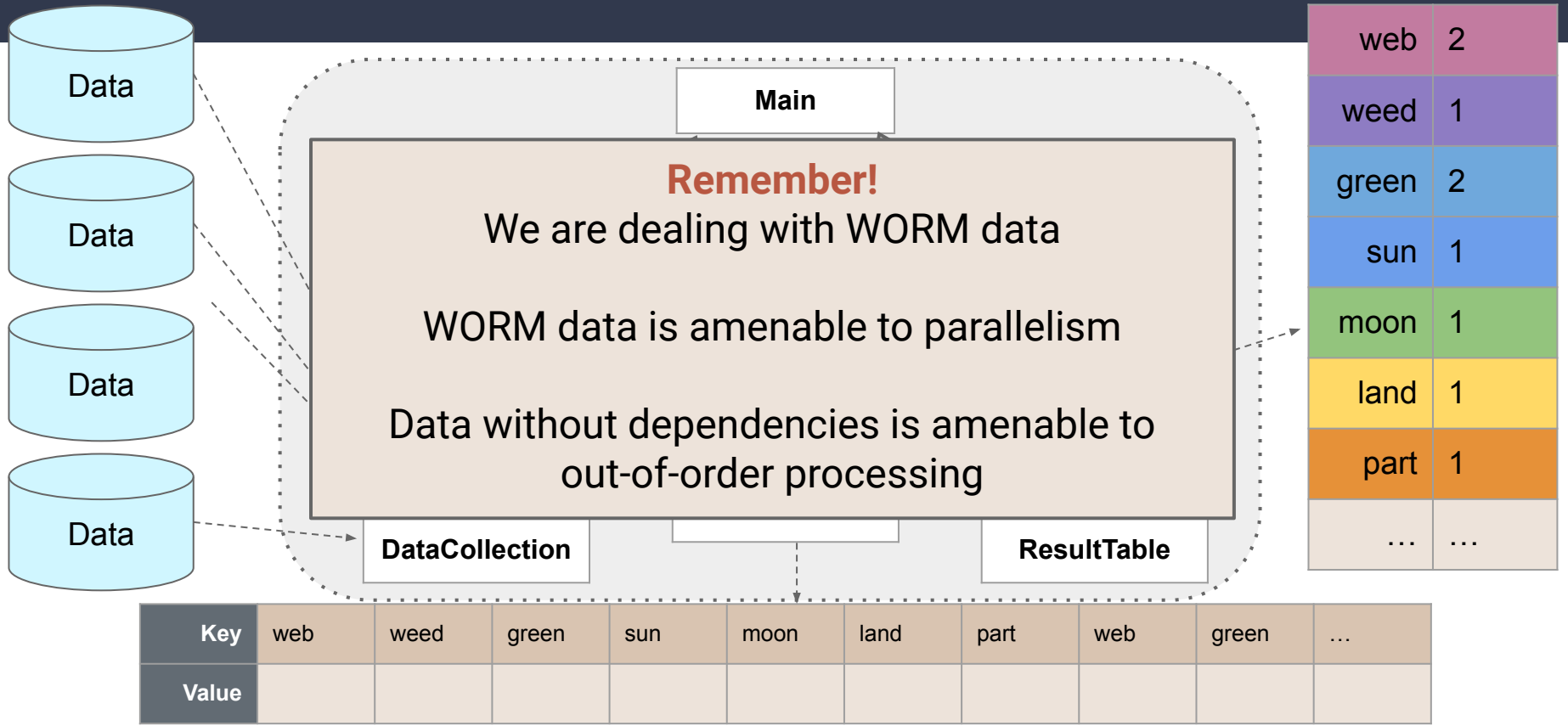
<yourID, yourFollowers>

<studentNum, studentDetails>

<word, numberOfOccurences>

etc…

Distribute the Data



Question 3 - MapReduce [Lec 8,9]

SETI@home is a long-running project searching for extra-terrestrial life by analyzing radio frequency signals recorded by various telescopes. The radio signal intensity was scaled and “printed” (stored) as integers from 0-35 inclusive, with digits from 0-9, and a-z representing 10-35. We want to configure a Hadoop-MapReduce infrastructure to analyze this voluminous repository for any significant contact from extraterrestrials. Consider a Hadoop-MapReduce configuration as given below:

- We use the word count algorithm from class. Here a "word" is a digit or character representing the SETI signal. Our reducer class is also used as our combiner class.
- Assume that the input has a total of **G = 40Tbyte data**. (1T = 1012 bytes, 1M= 106 bytes)
- Input corpus is split equally into **S sites**, each running a MR cluster.
- Assume you plan to configure **M = 200** mappers per site. There are **R** reducers.

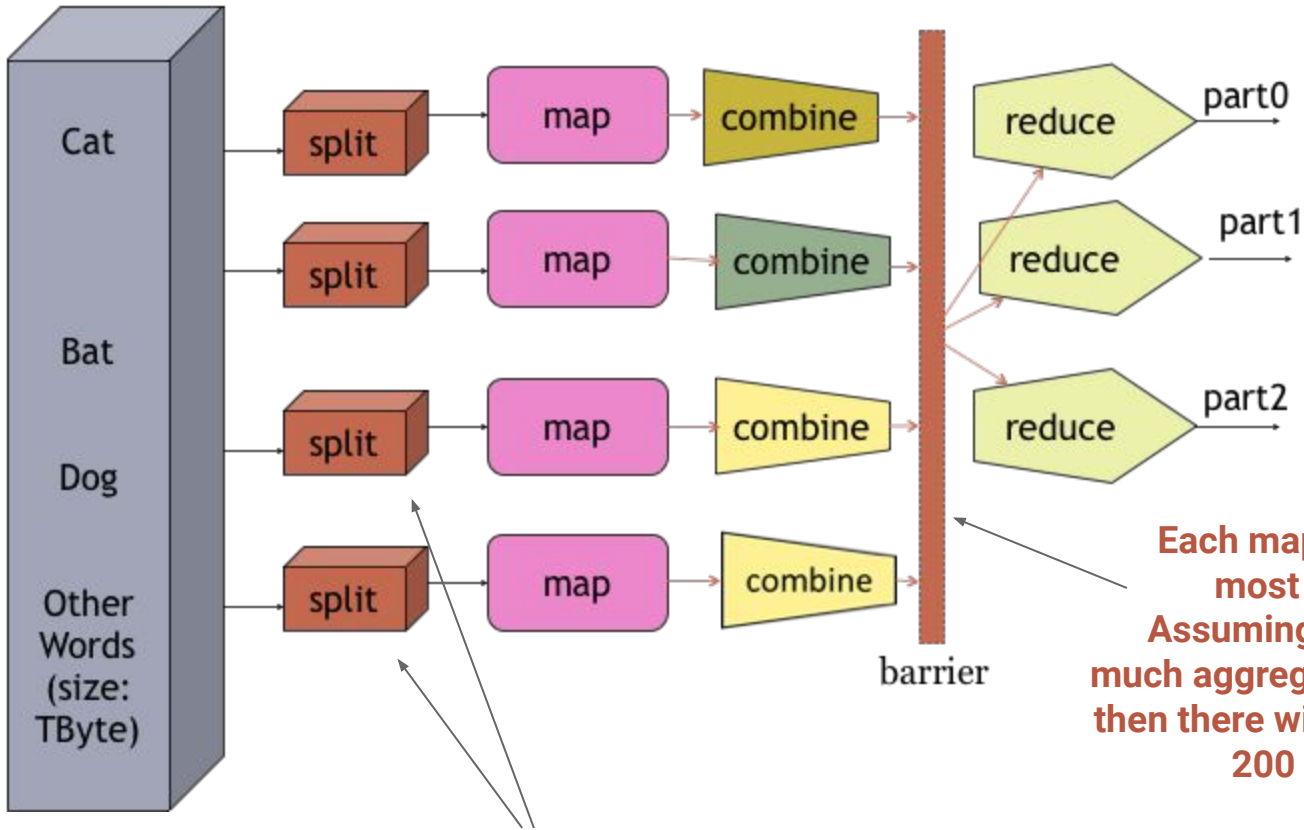
Question 3 - MapReduce [Lec 8,9]

a. What is the:
   i. Input keyspace of the mappers?
   ii. Size of the input processed by each site?
   iii. Workload of each mapper in bytes?

b. Assume that mappers suppress the range of values (0-15) and emit only the radio signals of values (16-35) inclusive. Assume that all combiners will run right before the shuffle and sort step.
   i. What is the maximum number of <key,value> pairs that will be shuffled and sorted?
   ii. How many distinct keys will each reducer have to reduce?
   iii. How many <key,value> pairs will be in the final output?

Input keyspace to mappers: {0-9} ∪ {a-z}

20 different keys / R reducers = 20 / R keys per reducer



20 kv pairs in final output

Each mapper will output at most 20 different keys. Assuming combiners do as much aggregation as possible, then there will be 20 kv pairs x 200 mappers x S sites

(40TB / S) per site

40TB / S / 200 = 2×10^{11} / S bytes per mapper

Question 4 - PageRank [Lec 11,12]

a. Given the above graph, write down the adjacency matrix used to compute the PageRank of the graph. (Use the naive formulation without using teleportation)
b. State the initial condition r0 for power iteration. Perform 3 iterations of power iteration to find r1, r2, and r3.
c. Will the power iteration solution for the above graph converge to what we want? Why or why not? If not, explain how to implement a fix.
d. Describe at least 2 differences in the MapReduce implementation of PageRank.

Page Rank: Matrix Formulation

Stochastic Adjacency matrix ***M***

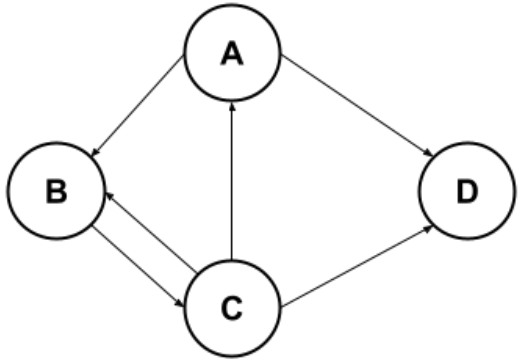
$M_{ji} = 1/(d_i)$ if there is a link from ***i*** to ***j***, else value is 0

If ***r*** is vector with the initial importance of a page and

$$\sum_i r_i = 1$$

Then the flow equation can be written as

$$r = M \cdot r$$



$$\begin{pmatrix} 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

Solving with Power Iteration

Given a web graph with ***n*** nodes, where the vertices are pages and edges are hyperlinks

Power iteration: a simple iterative scheme

Suppose there are ***N*** web pages

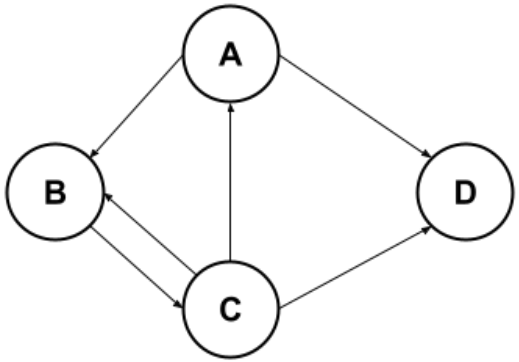
1. **Initialize:** $r(0) = [1/N,....,1/N]^T$
2. **Iterate:** $r(t+1) = M \cdot r(t)$
3. **Stop when:** $||r(t+1) - r(t)||_1 < \varepsilon$

Google Formulation

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

or equivalently

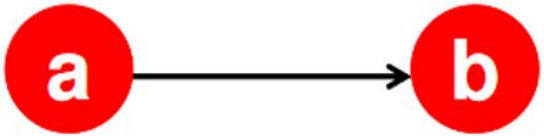
$$r = Mr$$



A = ⅓ C
B = ⅓ C + ½ A
C = B
D = ⅓ C + ⅓ A

¼	1/12	1/12	5/72
¼	5/24	3/24	8/72
¼	1/4	5/24	3/24
¼	5/24	3/24	8/72

Does this converge to what we want?



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

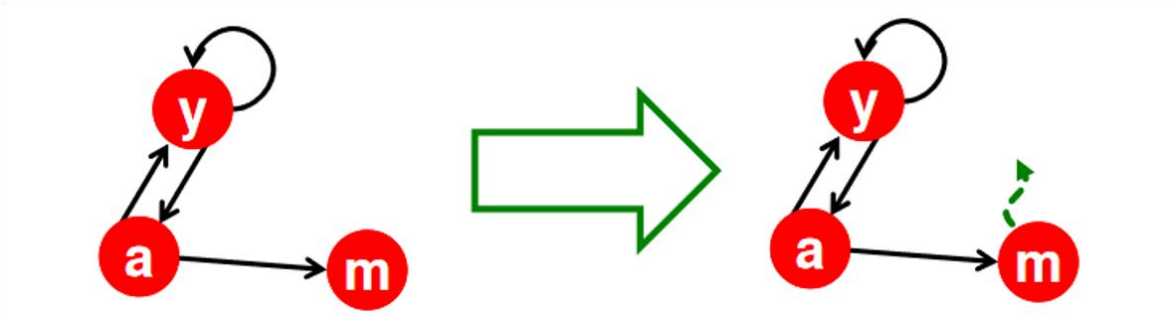
- **Example:**

$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Iteration 0, 1, 2, …

Solution: Teleports

Teleport with probability 1.0 at dead ends



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Graph Representation and Input Format

Our graphs will be represented as a collection of **Node** objects

```
Node:
    nodeId,
    distanceLabel,
    adjancencyList[nodeId, distance],
    ...
```

Input the graph as text and parse it to build our `<key, value>` pairs

So what are our `<key, value>` *pairs?*

<key, value> pairs

We actually need two types of <key, value> pair:

1. `<nodeId n, Node N>` `// nodeId to Node object`
2. `<nodeId n, distance>` `// nodeId to distance so far`

Iteration

- Each *iteration* in the algorithm is a MapReduce job
- Iterations and termination are coordinate by an external *driver* application (more on this in future lectures)
- The first iteration starts at the source node (with distance 0)
  - It updates and emits all distances for nodes in the adjacency list
- The next iteration takes the output from the previous and updates/emits all distances for nodes connected to this set of nodes
- Continue until termination

PageRank with MapReduce

How do we account for dead ends nodes?

- Simply redistribute its PageRank to all other nodes
- One iteration requires PageRank computation + redistribution of “unused” PageRank
  - Track total leaked PageRank during the computation, then redistribute it as a second MapReduce job in the same iteration

What about breaking out of spider traps?

- This can also be taken care of when the leaked PageRank is redistributed from dead ends

Second Phase Redistribution Formula:

Random hop probability

Total leaked PageRank "mass"

$$p' = \alpha\left(\frac{1}{|G|}\right) + (1-\alpha)\left(\frac{m}{|G|} + p\right)$$

PageRank from first phase

Question 5 - Word Co-Occurrence [Lec 13]

a. Write pseudocode for a mapper and a reducer to compute word co-occurrence using the pairs approach. You can assume that the function Neighbors(w) is already defined for you, and returns a list of words in the same context as w.
b. The other approach for computing word co-occurrence is using stripes. How do pairs and stripes relate to our original sequential formulation using matrix M.
c. Describe one advantage and one disadvantage that the stripes approach has compared to the pairs approach.

Word Co-Occurrence - Pairs Method

```
class Mapper
  method Map(docid id, doc d)
    for all w in d do
      for all u in Neighbors(w) do
        emit((w,u), 1)
```

```
class Reducer
  method Reduce(pair p, int[] cnts)
    sum ← 0
    for all c in cnts do
      sum ← s + c
    emit(p, sum)
```

Pairs produce N^2 keys, each key is an entry in the matrix
Stripes produce N keys, each key is a row in the matrix

Analysis of Stripes

+ Stripes generate far fewer <key, value> pairs
+ Stripes are much more compact (the pairs approach duplicates the left word in the pair for every pair)
+ Fewer and shorter keys means less sorting
+ Better for local aggregation
- Values are larger and more complex with more serialization overhead
- Scalability concerns similar to In-Mapper combining (memory overflow)