Lean Data Analysis

Achieving Awesome Data Dexterity with MATLAB Desktop

Joan Carroll
Liquidnet, Inc.
Post Trade Analytics
IMAGINE
What to do...?

ASSESS

EXPLORE
What to do...?

VISUALIZE RESULTS

DEPLOY
IMAGINE

It’s 2004.
You are in charge of Yahoo’s web crawler and search engine ...
An Industry In Trouble
Does this still work?

- Assess
- Explore
- Deploy
- Visualize Results
Threat => Opportunity
Or more precisely ...

VISUALIZE RESULTS...

DEPLOY

and ... iterate
How Do You Perceive Data?
Looming Threat?
Exploitable Resource?
A Hidden Opportunity
A Deep Dive
“Organizations need to invest in a lean approach to data and analytics, which aligns all business capabilities, including strategy, people, process, and technology.”

Source: Deloitte: Big Data, Time for a lean approach in financial services
Scenario

- Historical Analysis
- Granularity to Tick Level
- Back-test along with Web-based data
- Predictive Model
- Desktop Hardware:
  - 24Gb, 16-Cores, Terabyte RAID1 and Network Stg
- Use Internal MATLAB Library
Issues

- Time
- Capacity
- Efficiency
- Robustness
Does this still work?

- Assess
- Explore
- Deploy
- Visualize Results
# Assess Step –

## DATA: NoSQL?

<table>
<thead>
<tr>
<th>Database</th>
<th>Type</th>
<th>MATLAB Interface?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mongo</td>
<td>Document</td>
<td>Yes</td>
</tr>
<tr>
<td>Couch</td>
<td>Document</td>
<td>Yes</td>
</tr>
<tr>
<td>Redis</td>
<td>Key-Value</td>
<td>Yes</td>
</tr>
<tr>
<td>Cassandra</td>
<td>Wide Column</td>
<td>?</td>
</tr>
<tr>
<td>HDFS</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Now for the ETL … starting with ‘E’

- Extract
- Transform
- Load

OPTIONS
- mongoimport – CSV -> Mongo Document
- MATLAB Mongo Driver - CRUD Operations
- Mongo Command Interface
- Mongo HTTP Interface (read-only)
- RESTful interfaces (3rd Party)
What about Parallelization?

mongo map-reduce (javascript!?)
Map-Reduce

Source: Map-Reduce for Machine Learning on Multicore, Chu, Kim, Lin, Yu, Bradski, Ng, Olukotun
Machine Learning

Map

COMPUTE
- weights
- distances
- probabilities

Reduce

UPDATE
- new weights
- centroids
- matrix
Multicore Map-Reduce Speedup

Source: Map-Reduce for Machine Learning on Multicore, Chu, Kim, Lin, Yu, Bradski, Ng, Olukotun
```javascript
var mapfun = function() {
  var Xy = {
    count: 1,
    x: this.explanatory_var,
    y: this.response_var,
    x^2: (this.explanatory_var)^2,
    y^2: (this.response_var)^2,
    xy: (this.explanatory_var * this.response_var)
  };
  emit(this.key, Xy);
};

var reduce = function(key, Xys) {
  var mat = {
    count: 0, sum_X: 0.0, sum_y: 0.0,
    sum_Xy: 0.0, sum_XX: 0.0, sum_yy: 0.0,
    lm: {} };
  for (var iX = 0; iX < Xys.length; iX++) {
    mat.count += 1;
    mat.sum_X += Xys[iX].x;
    mat.sum_y += Xys[iX].y;
    mat.sum_Xy += (Xys[iX].x * Xys[iX].y);
    mat.sum_XX += (Xys[iX].x * Xys[iX].x);
    mat.sum_yy += (Xys[iX].y * Xys[iX].y);
  }
  mat.lm['b'] = (mat.count * mat.sum_Xy - mat.sum_X * mat.sum_y) / (mat.count * mat.sum_XX - mat.sum_X * mat.sum_X);
  mat.lm['intercept'] = (mat.sum_y - mat.lm.b * mat.sum_X) / mat.count;
  mat.lm['r2'] = Math.pow((mat.count * mat.sum_Xy - mat.sum_X * mat.sum_y) / Math.sqrt((mat.count * mat.sum_XX - mat.sum_X * mat.sum_X)), mat.count * mat.sum_yy - mat.
  return mat;
}

db.lean.mapReduce(
  mapfun,
  reduce,
  { out: "lean_mapred_test" }
);```
equivalent MATLAB code for linear regression
... accessing
% Replicate Fitted Linear Regression Implemented in MongoDB MapReduce command

MongoStart
host = 'localhost';
port = '27017';
serverUrl = sprintf('%s:%s',host,port);
db = 'mydb';
ns = sprintf('%s.lean',db);
leanDb = Mongo(serverUrl);

% get number of days in timeseries, for prealloc
vDates = leanDb.distinct(ns,'ADATE');
numDays = length(vDates);

% foreach symbol ...
vSymbols = leanDb.distinct(ns,'RIC');
umSyms = length(vSymbols);

for idx = 1:numSyms
    % obtain a cursor of all observations over time
    bb = BSONBuffer;
    bb.append('KEY', char(vSymbols(idx)) ); % per symbol loop
    query = bb.finish();
    cursor = MongoCursor(query);
    if leanDb.find(ns, cursor)
        mat = zeros(numDays,2); % preallocate regression matrix
        k = 1;
        while cursor.next()
            b = cursor.value();
            sprintf('Signal %10.5f = f(Factor %10.5f)',b.value('y'),b.value('X'))
            X = double(b.value('X'));
            y = double(b.value('y'));
            mat(k,:) = [X y];
            k = k +1;
        end
    end
    % remove unused rows
    mat(k:end,:) = [];
    % linear model
    mdl = fitlm(mat(:,1),mat(:,2));
end
So Far, So Good
Visualize Results –

Scheduling ... ?

<crickets>
**EXPLORE:** Cloudera “QuickStart” Hadoop VM

- MATLAB MCR -> MCRInstaller (Unix)
- ETL
  - **Extract**
    - Share Local Drive -> VM (HGFS)
    - --> copy to HDFS in next step
  - **Transform & Load**
    - hadoop dfs –copyFromLocal ...

**DEPLOY:** MATLAB Hadoop job (Streaming Hadoop)
- Schedulers: Oozie, YARN etc
Map-Reduce

Figure 1: Multicore map-reduce framework

Source: Map-Reduce for Machine Learning on Multicore, Chu, Kim, Lin, Yu, Bradski, Ng, Olukotun
Demo
Inundated with Data?
Awesome!
Thank You!