Building Blocks for Exploratory Data Analysis Tools

Sara Alspaugh  
Archana Ganapath  
Marti Hearst  
Randy Katz  

UC Berkeley  
Splunk Inc.  
UC Berkeley  
UC Berkeley
Motivation
Motivation
Key Idea

- exploit similarity between data sets to make recommendations
  - but we don’t have analyzed data sets, so use queries (small programs) instead
Approach

1. collect user queries from Splunk
   - but not data sets

2. apply latent semantic analysis
   - test key idea
   - use extension of this for recommending
Splunk

- View a demo here: http://www.eecs.berkeley.edu/~alspaugh/misc/splunk_demo_screencast.mov
  - you might need QuickTime for your browser
  - be patient, it can take a while to load

- Splunk collects and indexes large amounts of semi-structured time series data

- Data is often log data
  - each time-stamped entry is a row

- Users visualize data via GUI and query language

- Data is processed in stages expressed in queries:
  - command arguments | command arguments | ...

- No schema; key-value pairs are extracted at run time
  - think bags of key-value pairs instead of tables
Splunk Query Example

```
search source=eqs7day-M1.csv
| eval Description=
  case(Depth<=70, "Shallow",
       Depth>70 AND Depth<=300, "Mid",
       Depth>300, "Deep")
| table Datetime, Region, Depth, Description
```

- commands and operators
- field (i.e., key or column)
- value (i.e., column values)
- pipe to next command
1. Romeo and Juliet.
2. Juliet: O happy dagger!
3. Romeo died by dagger.
4. “Live free or die”, that’s the New-Hampshire’s motto.
5. Did you know, New-Hampshire is in New-England.

**DOCUMENTS**

d1 : romeo, juliet.
d2 : juliet, happy, dagger
d3 : romeo, die, dagger.
d4 : live, free, die, New-Hampshire
d5 : New-Hampshire

**QUERIES**

1. search sourcetype=access_combined
   | where isnull(task_queue)
   | timechart count span=1min
   | eval count=count/60
2. search host="appspot.com" change_time=*  
   | eventstats count as Total
   | bucket change_time span=log10
   | stats count as Count, max(Total) as Total by change_time
   | eval percentage=Count/Total*100

bucket : change_time
search : sourcetype, host, change_time
eval : count, total, percentage, 60, 100
eventstats : count
stats : count, total, change_time
timechart : count, 1min
where : task_queue

die dagger
Conclusion

• Evidence for making recommendations based on similarity of data:
  inconclusive but promising

• Possible approaches:
  – recommendation algorithms: LSI, nearest neighbor
  – dimensionality reduction: NMF, t-SNE (hat tip reviewer #3)
  – Bayesian: naïve, hierarchical
  – natural language processing
  – others?
Thank you.

Questions?