Let’s begin with a story…

Dora the Data Explorer has a new job!

Let’s explore Yahoo’s data!
Dora’s new job

Explore Yahoo’s data

Yahoo has a ton of data though and they keep getting more

Hard to make sense of all the data

Dora runs into some challenges when working with this data
Screen resolution has its limitations!

40,000 data points from Yahoo data
Tico the Squirrel has an idea!

Increase the number of pixels! Get a Powerwall!

53.7 million pixel Powerwall at the University of Leeds
Boots is laughing at how ineffective this is

- From the beginning of recorded time until 2003, humans had created 5 exabytes (5 billion gigabytes) of data.

- In 2011, the same amount was created every two days

YOU WILL SOON REACH A **NEW LIMIT**!

Besides....
Dora is too adventurous to stay in the office

But look at these tiny screens!
How do you gain insight from your data?

Even if you manage to fit all your data pixels on a screen

Humans don’t think in terms of hundreds of numbers let alone tens of thousands

Can’t draw insight from a large collection of points
We need to find ways to fully utilize/squeeze data points into the resolution we have…

And more importantly make sense out of our data as well!
Boots is thinking...

How about zoom in to the points we are interested in? This is similar to how you navigate online maps.
Boots has an idea

How can Yahoo Maps show continents and also small islands? We need to be able to see the big picture and small outliers.
Boots is thinking...

But we have so many dimensions! We can’t color everything differently! That won’t work for all the dimensions we have.
Hierarchies are a powerful way to explore

Yahoo Maps is so powerful! It can zoom in and out MULTIPLE times and does it FAST! Can we do the same?
Can hierarchies help to better visualize big data?
## Hierarchical Visualization Challenges

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Many data points</td>
</tr>
<tr>
<td>2</td>
<td>Showing outliers</td>
</tr>
<tr>
<td>3</td>
<td>Many dimensions</td>
</tr>
<tr>
<td>4</td>
<td>Interactive visualization</td>
</tr>
<tr>
<td>5</td>
<td>Integrating into working software</td>
</tr>
</tbody>
</table>
### First Challenge

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Many data points</td>
<td>Technique: Tree Maps</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>Showing outliers</td>
<td></td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Many dimensions</td>
<td></td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Interactive visualization</td>
<td></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>Integrating into working software</td>
<td></td>
</tr>
</tbody>
</table>
How Tree Maps Work

TreeMaps visualize **hierarchical structures** onto a rectangular region in a space-filling manner.

A node's weight (bounding box) determines its display size

- Measure of importance or degree of interest

Advantages and Disadvantages of Tree Maps

✓ 100% use of the available display space
✓ Allows users to set display properties (colors, borders, etc.)

✗ Number and variety of domain properties visualized is limited
✗ Cluttered

And Dora is crying

I can’t focus on one set of points only, now I am stuck with the whole dataset! And everything looks the same! Now I have to do this again for the a smaller set!

Moving beyond just displaying data

Data exploration should maximize insight into a data set

Interactively
✓ Retrieve meaningful relations
✓ Extract inferences from the data
✓ Uncover underlying structure
✓ Detect patterns or anomalies
✓ Test underlying assumptions

Active process of discovery

NOT passive display
## Second Challenge

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Many data points</td>
</tr>
<tr>
<td>Technique</td>
<td>Tree Maps</td>
</tr>
<tr>
<td>2</td>
<td>Showing outliers</td>
</tr>
<tr>
<td>Technique</td>
<td>Zoom clustering</td>
</tr>
<tr>
<td>3</td>
<td>Many dimensions</td>
</tr>
<tr>
<td>4</td>
<td>Interactive visualization</td>
</tr>
<tr>
<td>5</td>
<td>Integrating into working software</td>
</tr>
</tbody>
</table>
How Zoom Clustering Works

- Cluster data points
- Store clusters in tree structure
- Allow branching and zooming into different areas of tree
- Support back tracking in zoom tree

User zoom tree operation

Zoom Tree Examples

Seems easy to use!

Demo video: Zooming Plot Charts

http://youtu.be/8dfkke95xCM?t=3m52s

Advantages and Disadvantages of Zoom Clustering

**Advantages**
- Not Overwhelmed with Data Points
- Can Process Data in Parallel
- Ability to Focus on Interesting Areas

**Disadvantages**
- Many-Dimensional Data

## Third Challenge

<table>
<thead>
<tr>
<th></th>
<th>Many data points</th>
<th>Tree Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Showing outliers</td>
<td>Zoom clustering</td>
</tr>
<tr>
<td>3</td>
<td>Many dimensions</td>
<td>Parallel Coordinates</td>
</tr>
<tr>
<td>4</td>
<td>Interactive visualization</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Integrating into working software</td>
<td></td>
</tr>
</tbody>
</table>
How Parallel Coordinates Work

- Large number of dimensions
- Highlights relation between dimensions

Difficult to distinguish the overall structure when the number of tuples becomes very large

Self Organizing Map & Parallel Coordinates

SOM algorithm + Parallel Coordinates

Drill-down on the clusters reveals the original data elements and the weight vector.

**Self Organizing Map:**

Nonlinear projection from m-dimensional space onto the two-dimensional display space.

Relies on distance, similarity and average.

Self Organizing Maps Explained

http://www.tubechop.com/watch/2698630

Interactive data analysis:

- Visual User Interface with drill down, filtering and zooming
- Zoom in by simply drawing a rectangle across the selected cluster bands.

Using the rectangle zoom for a more detailed view.

Advantages and Disadvantages of Parallel Coordinates

Cool machine learning method!

BUT.....

X Parallel Coordinates does not provide a good overview as it becomes hard to see the structure in the data when the dataset gets large
X Runs out of encoding possibilities as the number of dimensions increases.
X Preprocessing or filtering the data is required
X Not efficient for visualizing datasets with non-numerical data
X Number of clusters/neurons predefined

## Fourth Challenge

<table>
<thead>
<tr>
<th></th>
<th>Many data points</th>
<th>Tree Maps</th>
<th>Showing outliers</th>
<th>Many dimensions</th>
<th>Parallel Coordinates</th>
<th>Interactive visualization</th>
<th>Parallel sampling</th>
<th>Integrating into working software</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dora is frustrated!

I can’t wait this long!
I want to access the remote server faster.

How do I quickly visualize hierarchical data?
Which visualization approach is cost-effective?
1. Subsampling and clustering
   a) Can be very fast
   b) Need to account for errors
   c) Grid and multiresolution

2. Parallel Servers
   a) Distributed computation (e.g. MapReduce, Spark, etc.)
   b) Don’t have to account for errors
   c) Aren’t restricted to certain data types
   d) Technically challenging to implement
1. **Uniform Grid**
   - Distributes points from original dataset into equal sized grids
   - Single level representation
   - Can be constructed quickly

2. **Hierarchical Multiresolution**
   - Multi level representation
   - Fewer approximation errors by showing more points where the data is changing rapidly
   - Quadratic complexity makes large problems difficult

---

[4] Lori A. Freitag and Raymond M. Loy, Comparison of Remote Visualization Strategies for Interactive Exploration of Large Data Sets, 2001
Isa comes to rescue Dora and Boots with another idea.

Why don’t we combine hierarchical clustering with parallel servers?

[4] Lori A. Freitag and Raymond M. Loy, Comparison of Remote Visualization Strategies for Interactive Exploration of Large Data Sets, 2001
Data Explorers are excited about PINK

Why PINK (Parallel Single Linkage)?

- Scalable parallel algorithm for single-linkage hierarchical clustering
- Structure of single linkage problem can be exploited for parallelism
- Single linkage hierarchical clustering dendrogram for a dataset and the MST of the corresponding complete graph produce identical clusters

[4] Lori A. Freitag and Raymond M. Loy, Comparison of Remote Visualization Strategies for Interactive Exploration of Large Data Sets, 2001
Connection to Minimum Spanning Tree

[4] Lori A. Freitag and Raymond M. Loy, Comparison of Remote Visualization Strategies for Interactive Exploration of Large Data Sets, 2001
How does PINK work?

Split Data Evenly

(a) Problem domain decomposition with $k$ partitions

(b) Two processes are each assigned two complete subgraphs

(c) Six processes are assigned one bipartite subgraph for the six pairs of partitions

(d) $K^2/2$ processors

Generating the Minimum Spanning Tree

- Solve subproblems using prim’s algorithm
- Combine partial solutions
- Subproblems may have edges not in MST
- Treat partial solutions as candidate edges
- Apply Kruskal’s algorithm to candidate edges

Binary Merging of Partial Solutions

- Combine two MSTs at a time from consecutive processors
- Add an edge that does not join vertices that are already in the same component
Explorers are happy with PINK’s performance

Why does PINK combine the dendrograms from consecutive processes?
- Overlapping data partitions
- Detect and eliminate edges sooner
- Cuts down memory and communication cost

What are the limitations of PINK algorithm?
- Minimum processor requirement \((K^2/2)\)
- Binary Merge – the entire dendrogram must fit in one processor
## Fifth Challenge

<table>
<thead>
<tr>
<th></th>
<th>Many data points</th>
<th>Tree Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Showing outliers</td>
<td>Zoom clustering</td>
</tr>
<tr>
<td>3</td>
<td>Many dimensions</td>
<td>Parallel Coordinates</td>
</tr>
<tr>
<td>4</td>
<td>Interactive visualization</td>
<td>Parallel sampling</td>
</tr>
<tr>
<td>5</td>
<td>Integrating into working software</td>
<td>?????</td>
</tr>
</tbody>
</table>
Open Challenges

1. We have the back end of big data but no front end

2. Most big data tools are focused on batch processing which isn’t good for visualization

This is changing
Open Challenges

3. Most front end tools don’t integrate with the back end tools

This is improving

4. Most front end tools don’t handle this type of data well (high dimensionality and many data points)
Comparison of Hierarchical Techniques

<table>
<thead>
<tr>
<th></th>
<th>Many data points</th>
<th>Showing outliers</th>
<th>Many dimensions</th>
<th>Interactive visualization</th>
<th>Working software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree Maps</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Parallel Coordinates</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Zoom Clustering</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Parallel Sampling</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>🟢</td>
<td>N/A</td>
</tr>
</tbody>
</table>

How can we combine these?
Can Dora use hierarchies to visualize big data?
Yes! With the right combination of techniques
Our Hybrid Opinion

Reviewed several techniques with different advantages and disadvantages

Use hierarchical clustering to tackle many data points

Use parallelization for performance
Use sampling because simple parallelization isn’t enough

This also happens to be approach we picked for our project!