# Scalable Search and Ranking for Scientific Data

Mirek Riedewald (Northeastern University)
Alper Okcan (Northeastern University)
Daniel Fink (Cornell Lab of Ornithology)

# Acknowledgments


- Wes Hochachka (Lab of O)
- Giles Hooker (Cornell Stats)
- Steve Kelling (Lab of O)
- Art Munson (Cornell CS)
- Biswanath Panda (Google)
- Kevin Webb (Lab of O)

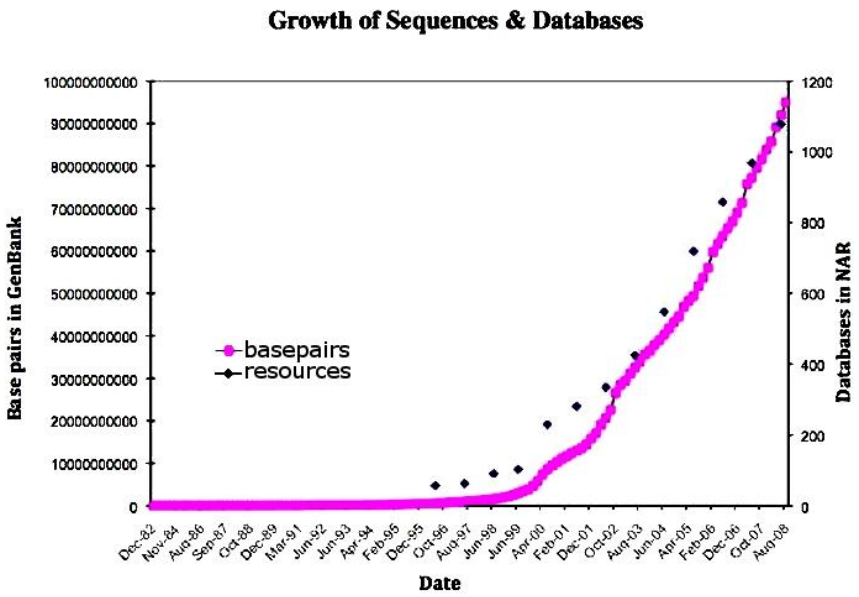

Cornell University
Computer Science

Cornell University
Department of Statistical Science


This material is based upon work supported by the National Science Foundation under Grant Nos. 0612031, and 0920869.

# Data-Driven Science

- Genome data
- Large Hadron Collider
  - Petabytes of raw data
- SkyServer
  - 818 GB, 3.4 billion rows
- Cornell Lab of Ornithology
  - 69M observations, 1000s of attributes
- DataONE
  - “Universal access to data about life on earth and the environment”

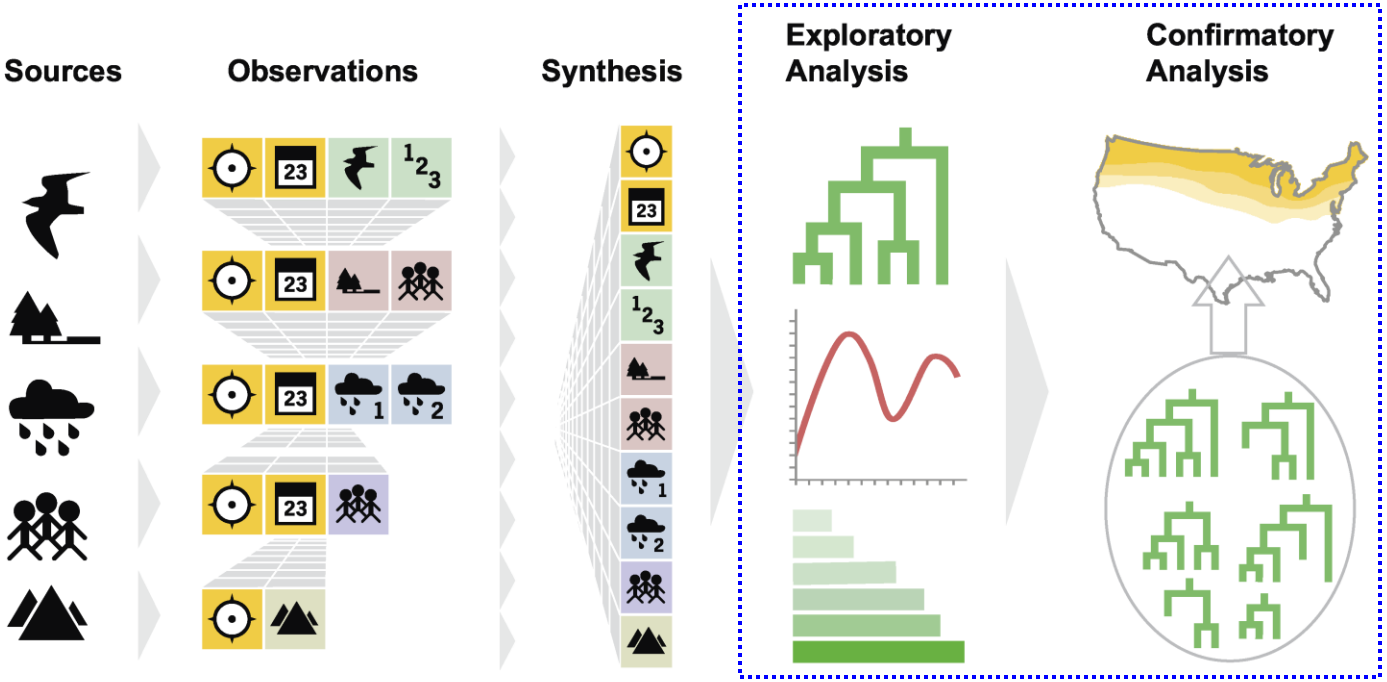



Source: Nature

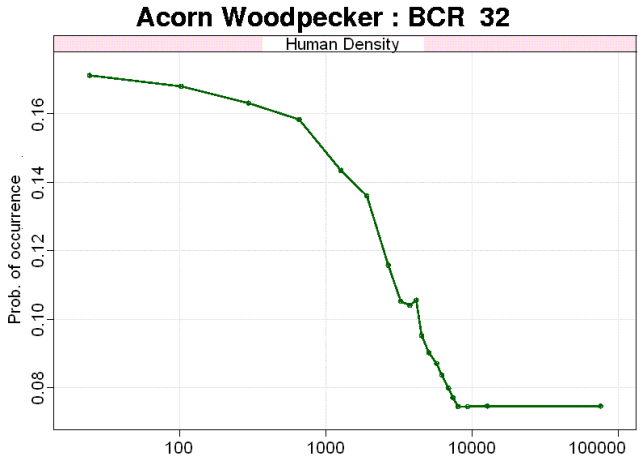
…science and engineering data are constantly being collected, created, deposited, accessed, analyzed and expanded in the pursuit of new knowledge. In the future, U.S. international leadership in science and engineering will increasingly depend upon our ability to leverage this reservoir of scientific data captured in digital form, and to transform these data into information and knowledge aided by sophisticated data mining, integration, analysis and visualization tools. (National Science Foundation Cyberinfrastructure Council, 2007)

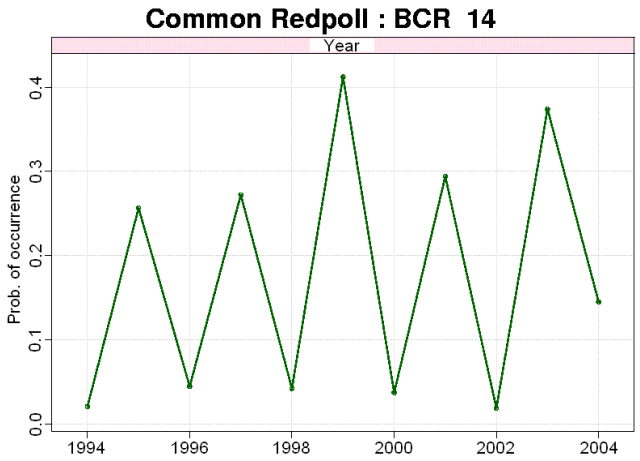
# New Tools For Discovery

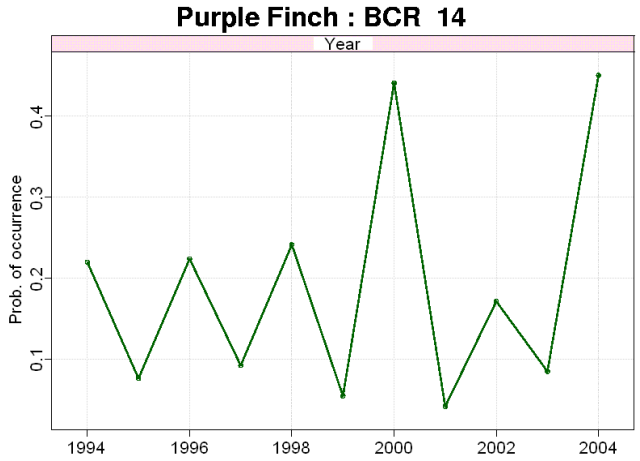
- Abundance of observational data
  - Discovery in observational data
    - Science + data mining + distributed data management = trouble?
  - Surprising pattern => inspiration for new hypothesis
    - Distinguish real versus spurious patterns

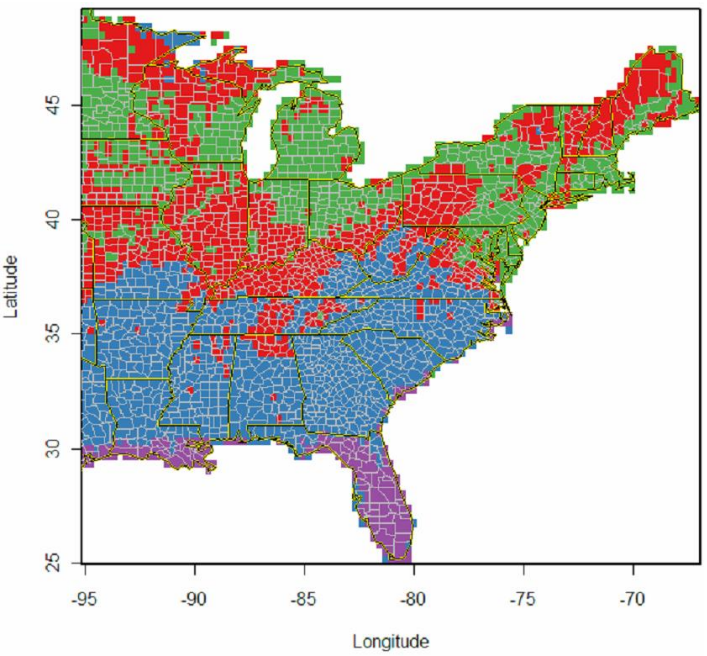


# Finding Patterns

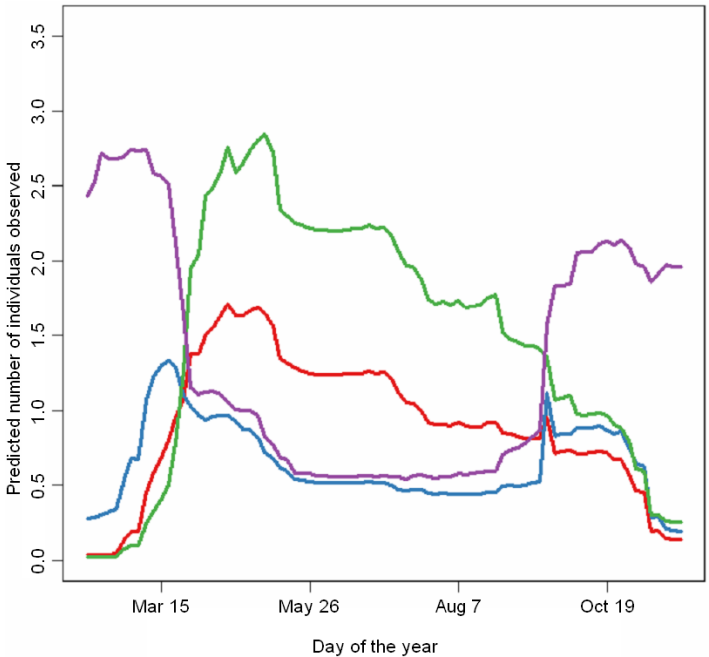



Tree Swallow migration


# The Scolopax System

- Search for patterns in prediction models based on user preferences
  - Model addresses many problems of observational data

- **Make this as easy and fast as Web search**

- User-friendly query language
  - Support broad class of patterns
- Formal foundation for query optimization
  - Inspired by relational algebra
- Query optimizer for execution in a distributed system
  - Fast ranking of patterns
  - On-the-fly pattern creation
- New data mining techniques
  - Good predictions for observational data
    - Noise, outliers, skew, missing values
    - Exploit known structure, e.g., spatio-temporal correlation
  - Amenable to fast distributed training, evaluation, pattern confidence computation

# Query Algebra

- Operators manipulate functions
  - Model, model summary are functions
  - Challenge: best representation
    - Decision tree, SVM, ANN, set of tuples
    - More structure enables better query optimization
    - Which structural properties of models are most important?
- Selection operator
  - Limits input domain of function, e.g., by geographical region
- Function join
  - E.g., summary pairs with opposite trends
- General summary operator
  - Maps a set of functions to a new set of functions, possibly with different input and output domains
  - Annotated with property attributes, e.g., approximation error, confidence
  - Too general for optimization

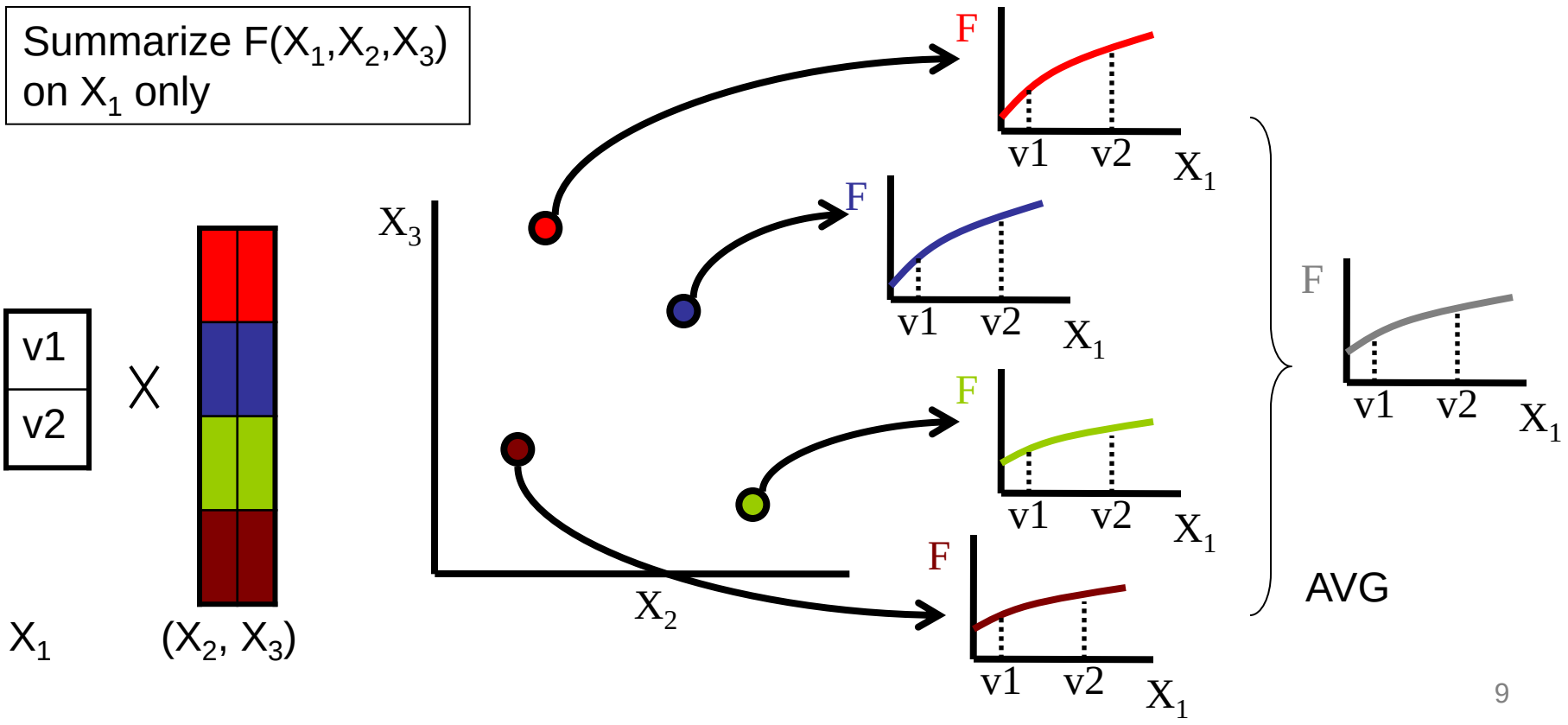
# Important Summary Operators

- Important classes of summary operators
  - Commonly used
  - Amenable to efficient implementation

- Simple transformations
  - Shift to mean=0, scale output to [0,1]
  - Relatively easy to find rewrite rules
- Popular transformations
  - Principal component analysis, regression model coefficients
- Single-function score operator
  - Max-min difference, variance, slope of regression line
- Multiple-function score operator
  - Quality of clustering of set of functions (#clusters, cluster cohesion and separation, edge density)
  - Function similarity

# The Projection-Summary Operator

- Maps input space to subset of attributes
- New function value for tuple **x**' is an aggregate of the function values of all original tuples **x** that map to **x**'
  - Cross product followed by GROUP-BY

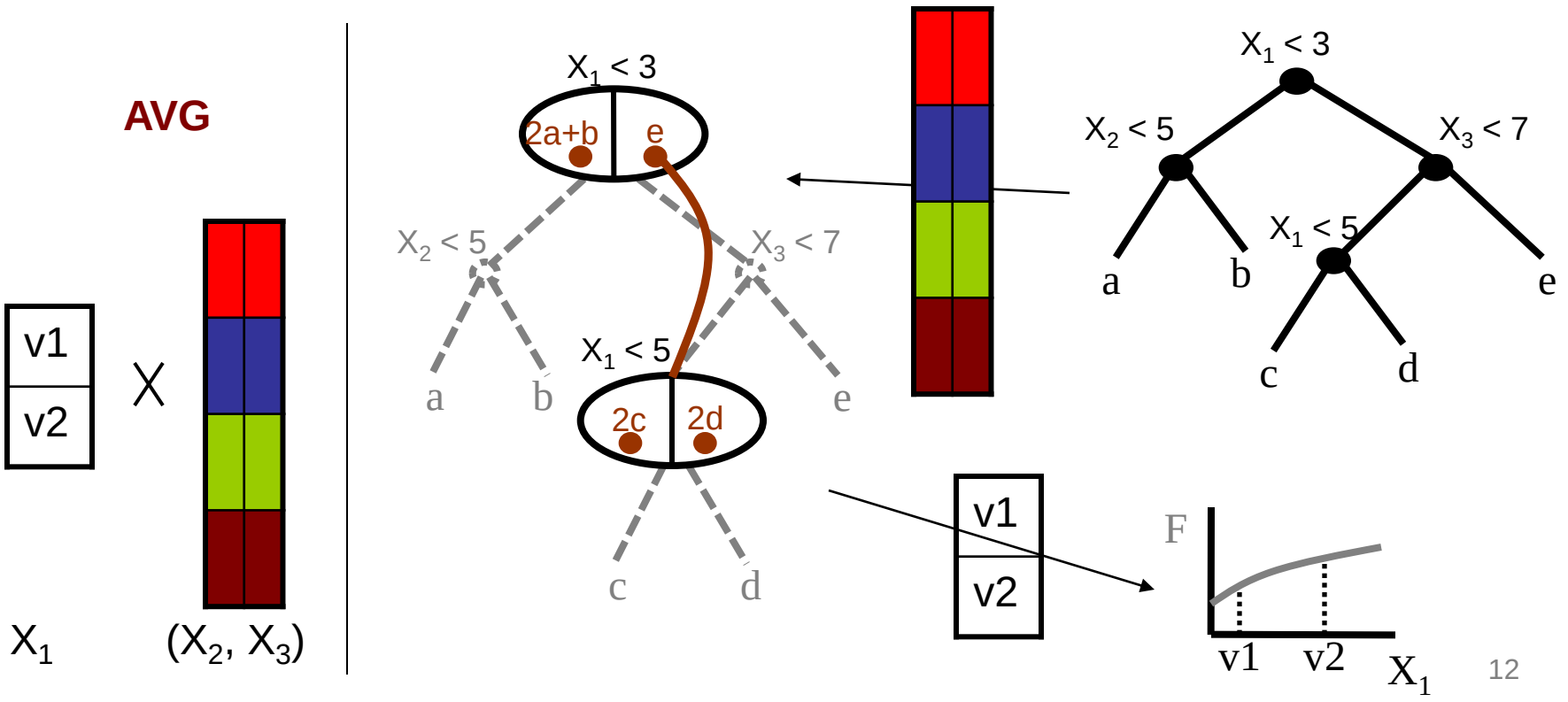


# Projection-Summary Properties

- Expresses all commonly used model summaries
  - $(X_2,X_3)$ from data sample: partial dependence function
- High computational cost
  - ~$10^{10}$ evaluations for medium-scale analysis
    - $5*10^5$ one- and two-dim summaries for 1000 attributes
    - Each for 10 visualization and $10^3$ data points
    - Many slices and dices, multiple models
- Workload structure
  - Repeated values (not entire tuples!) due to cross product: share computation
  - Aggregation of model predictions: push into model
  - Inter-summary commonality due to shared non-summary attributes: share computation
  - Exploiting this has to be model-specific

# Single-Summary Computation For Tree Models

- Exploit workload properties
  - Original cost: O(|V|*|D|*|T|) time
  - New cost: O(|D|*|T|+|V|*|t|) time
- Generalizes to tree ensembles



# Naïve vs. ShortCircuiting

| \|V\| | Naïve (secs) | Multi-point Shckt (secs) |
|---|---|---|
| 100 | 85.0 | 3.02 (=2.96+0.06) |
| 400 | 311.5 | 3.17(=2.97+0.20) |
| 625 | 469.8 | 3.29(=2.96+0.33) |

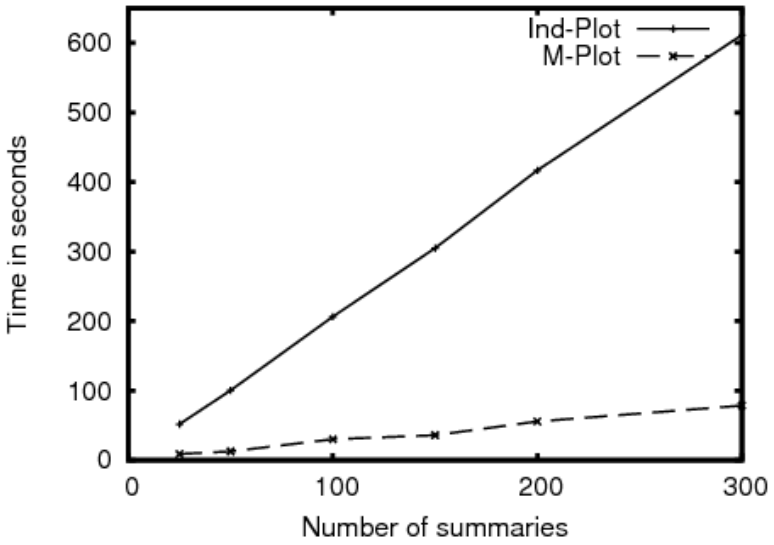
2-d Summary On Frequent Attributes (12%,11%)

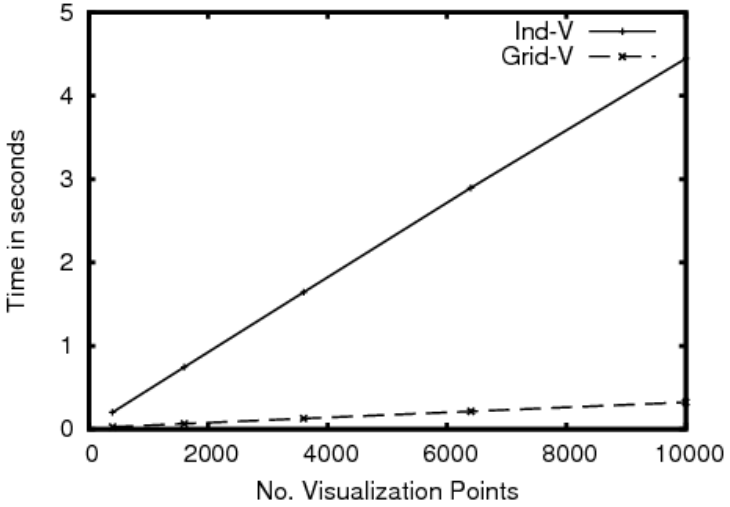
| \|V\| | Naïve (secs) | Multi-point Shckt (secs) |
|---|---|---|
| 100 | 84.8 | 2.1 (=2.1+0.001) |
| 400 | 324.5 | 2.1(=2.1+0.001) |
| 625 | 462.7 | 2.1(=2.1+0.002) |

2-d Summary On Infrequent Attributes (1%)

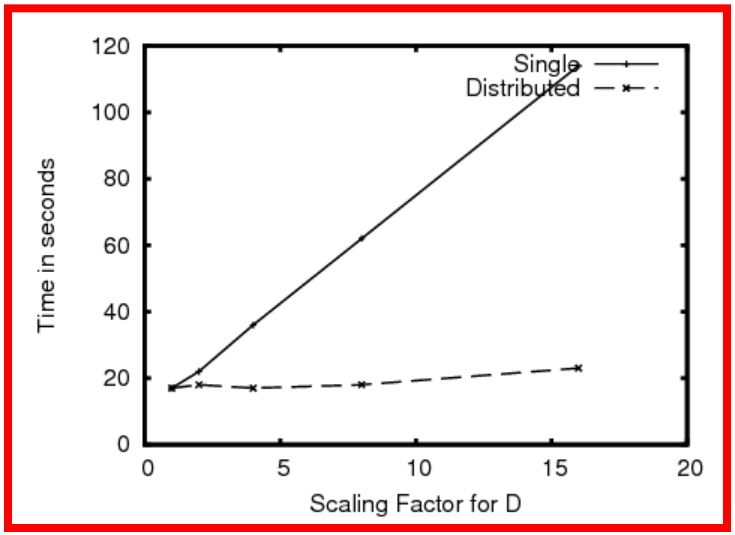
# More Improvements

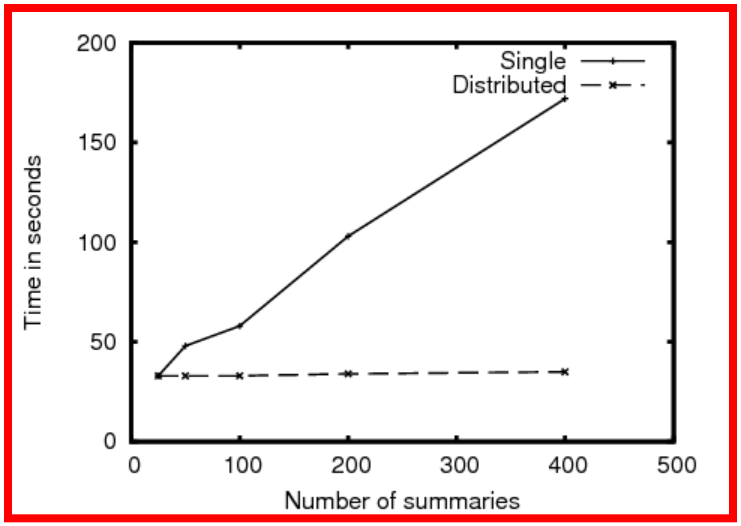
- Many summaries
  - Shared attributes
    - $\{X_1, X_2, \ldots, X_{100}\}$, summaries on $X_1$ and $X_2$
    - Common non-summary attributes: $\{X_3, \ldots, X_{100}\}$
  - Avoid repeated tree traversal
- Grid of visualization points
  - Decompose cross-product during traversal

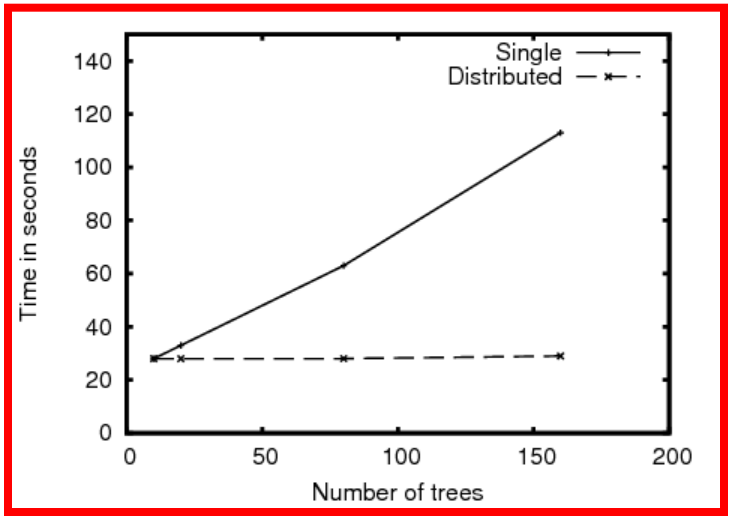


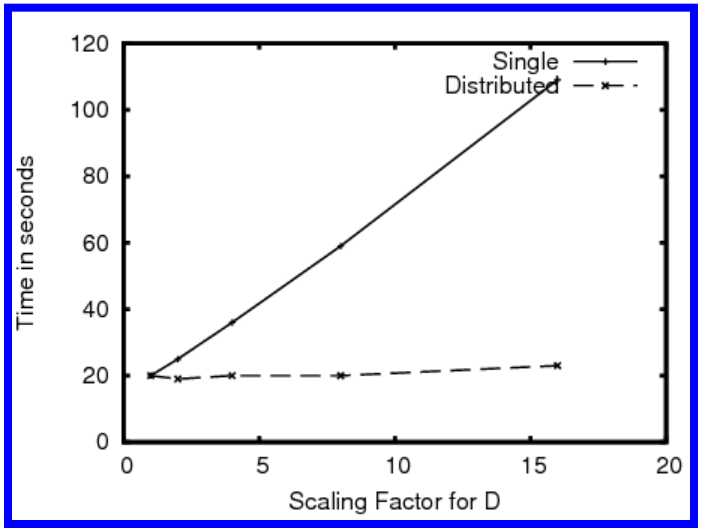



# Parallel Implementation on Hadoop




Red: visualization points given, blue: only short-circuit tree computation

# Next Steps

- Refine design of formal pattern preference language
- Projection-summary operator improvements
  - Current result for trees: months of processing time reduced to hours with exact same results
  - Extend to other model types
  - Approximation for further speedup
- Optimization for entire ranking query
  - Nested summary operators
- User-friendly query language
  - Refine ranking function with minimal user involvement
- New data mining techniques
  - Faster interaction detection
  - Parallelizable semi-parametric models