Geography on the web

Lars Backstrom
Facebook

Introduction

- Information is becoming increasingly geographic as it becomes easier to geotag all forms of data, and many devices have embedded GPS.
- What sorts of questions can we answer with this geographic data?
  - Query logs
  - Friendship
- Data is noisy. Is there enough signal? How can we extract it.
- Simple methods aren't quite good enough, we need a model of the data.

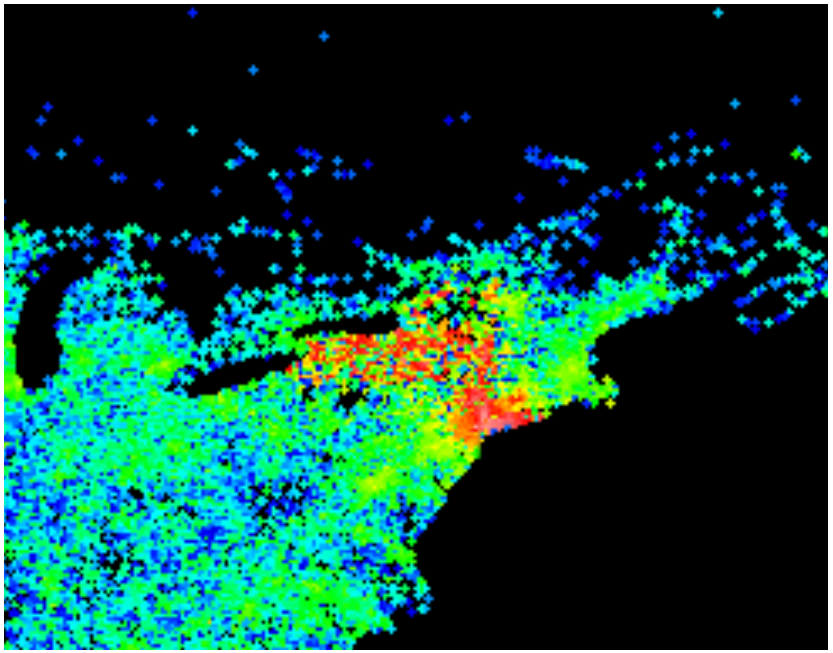
Outline

- Query Logs
  - Probabilistic, generative model of queries
  - Results and evaluation
  - Adding temporal information to the model
  - Modeling more complex geographic query patterns
  - Extracting the most distinctive queries from a location
- Facebook GEO-Data
  - Understanding data
    - Where do people live?
    - How far away do their friends live?
    - How are the two related?
  - Location Algorithm
    - Can we predict locations from friends' locations?
    - Using the network to improve beyond prediction based on located friends' locations.
- Conclusions

Outline

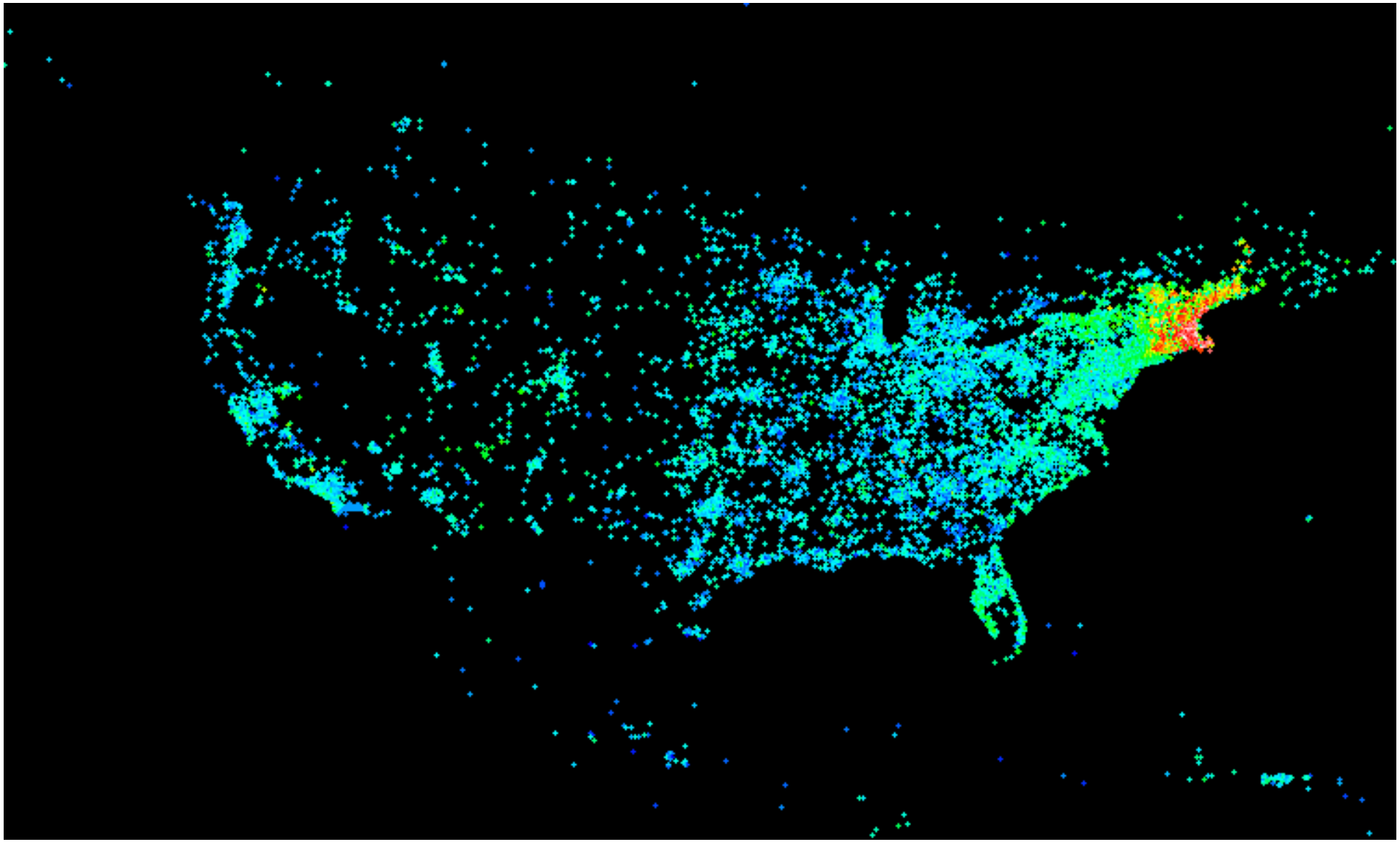
- Query Logs
  - Probabilistic, generative model of queries
  - Results and evaluation
  - Adding temporal information to the model
  - Modeling more complex geographic query patterns
  - Extracting the most distinctive queries from a location
- Facebook GEO-Data
  - Understanding data
    - Where do people live?
    - How far away do their friends live?
    - How are the two related?
  - Location Algorithm
    - Can we predict locations from friends' locations?
    - Using the network to improve beyond prediction based on located friends' locations.
- Conclusions

Query Logs

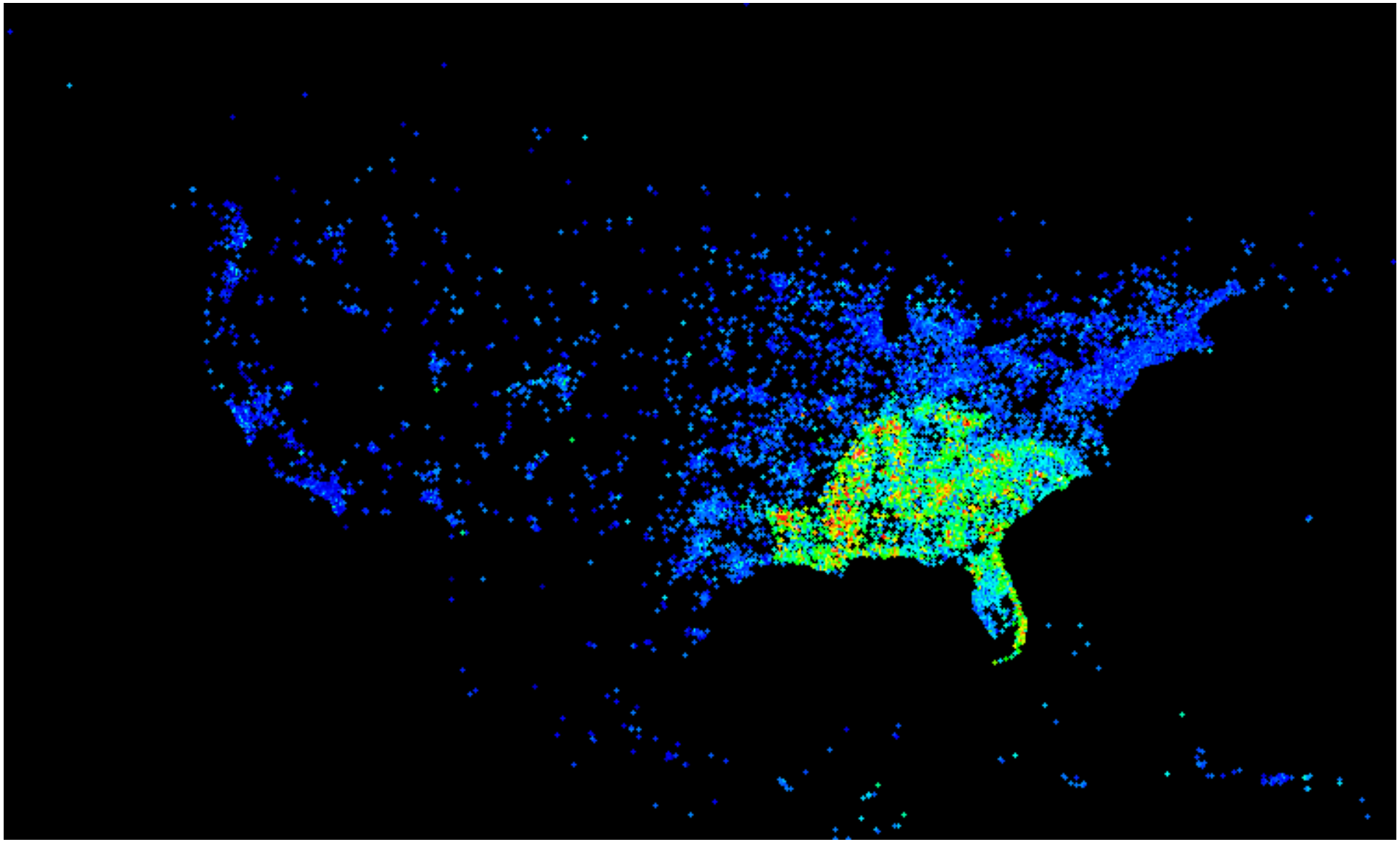


- Many topics have geographic focus
  - Sports, airlines, utility companies, attractions
- Our goal is to identify and characterize these topics
  - Find the center of geographic focus for a topic
  - Determine if a topic is tightly concentrated or spread diffusely geographically
- Use Yahoo! query logs to do this
  - Geolocation of queries based on IP address

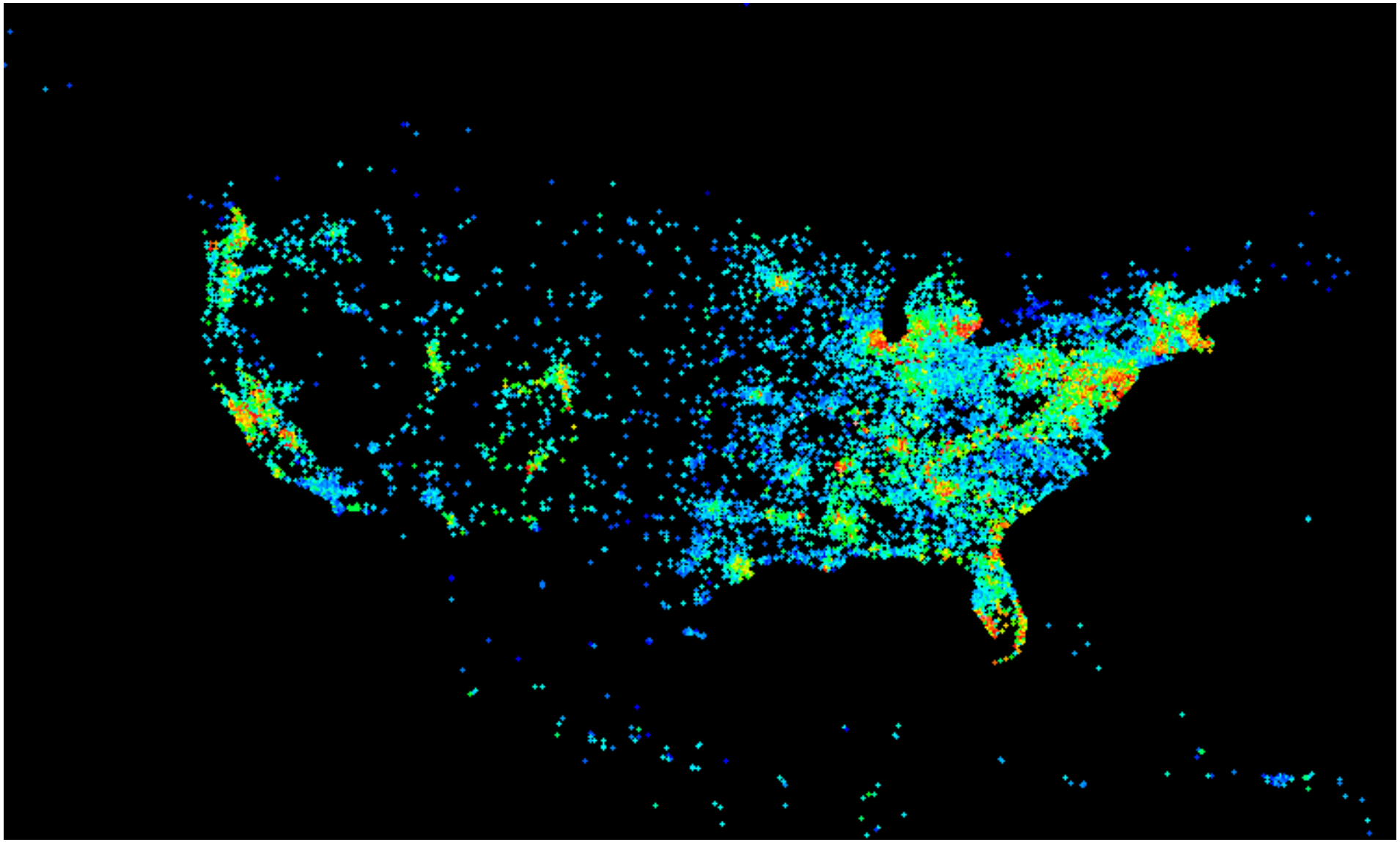
Red Sox



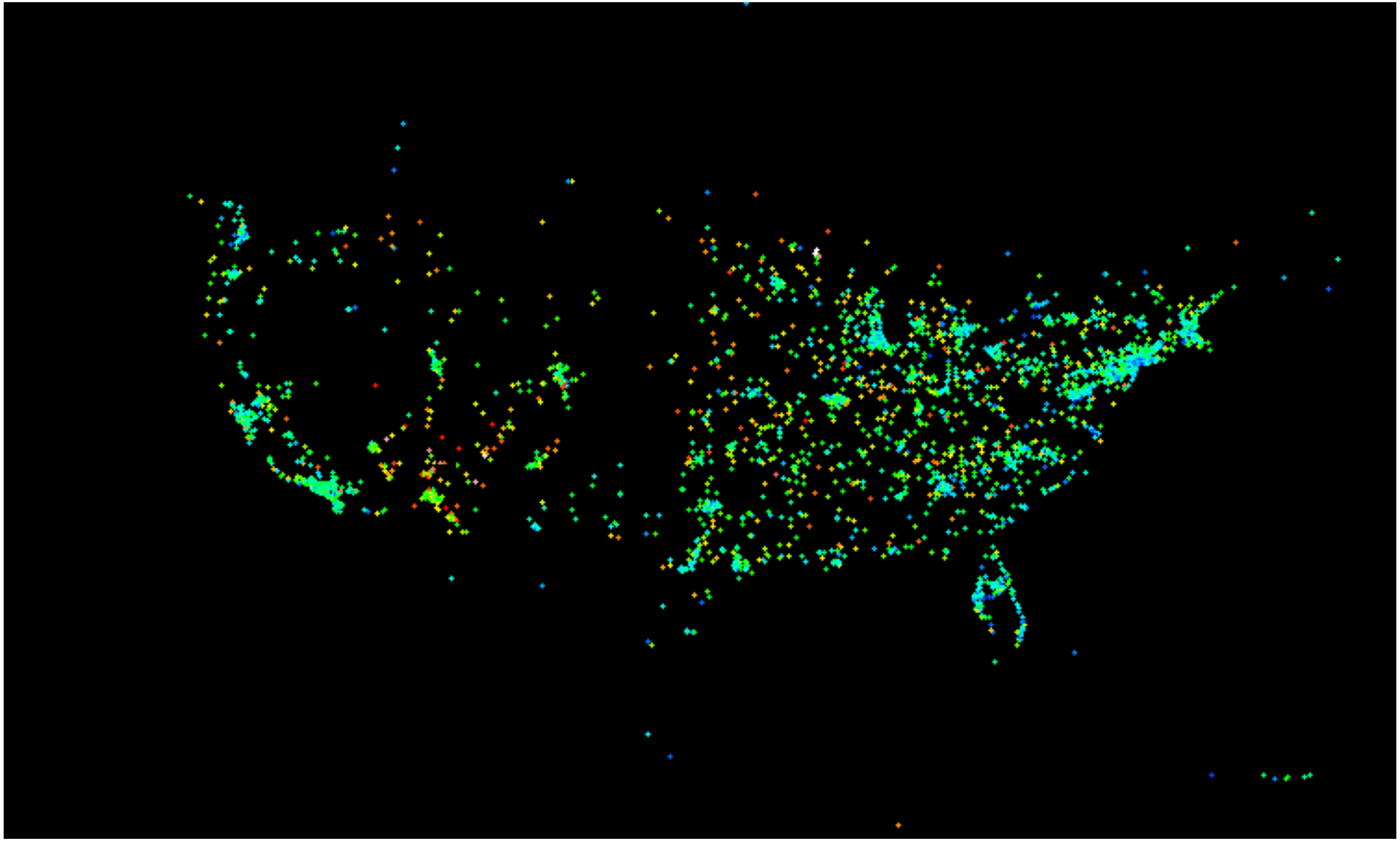
Bell South



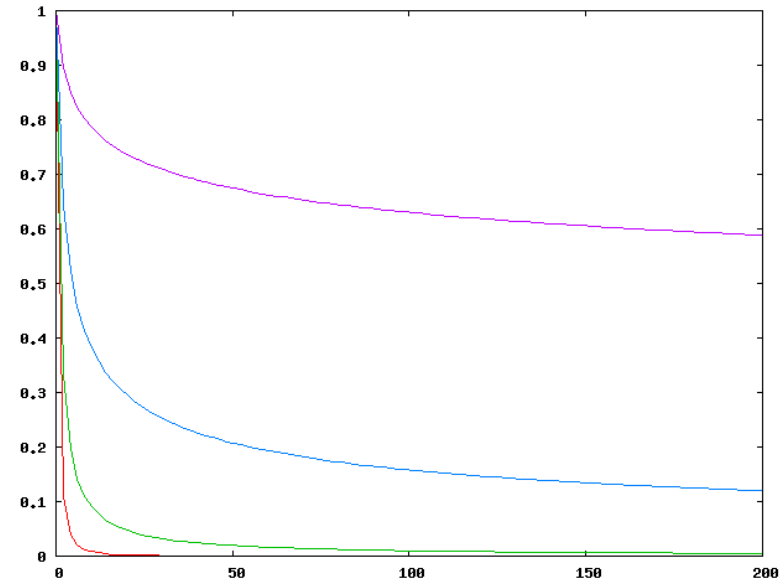
Comcast.com



Grand Canyon National Park



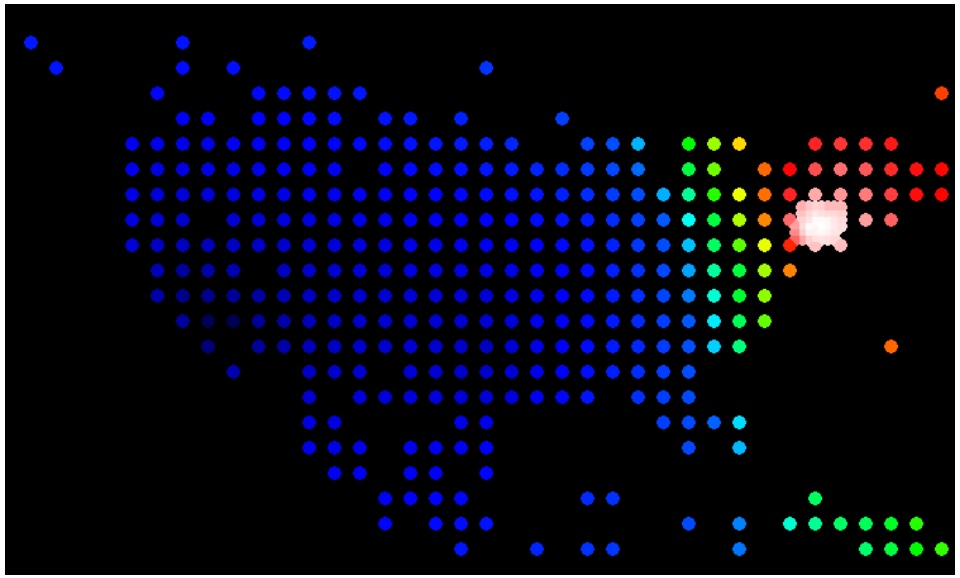
Probabilistic Model

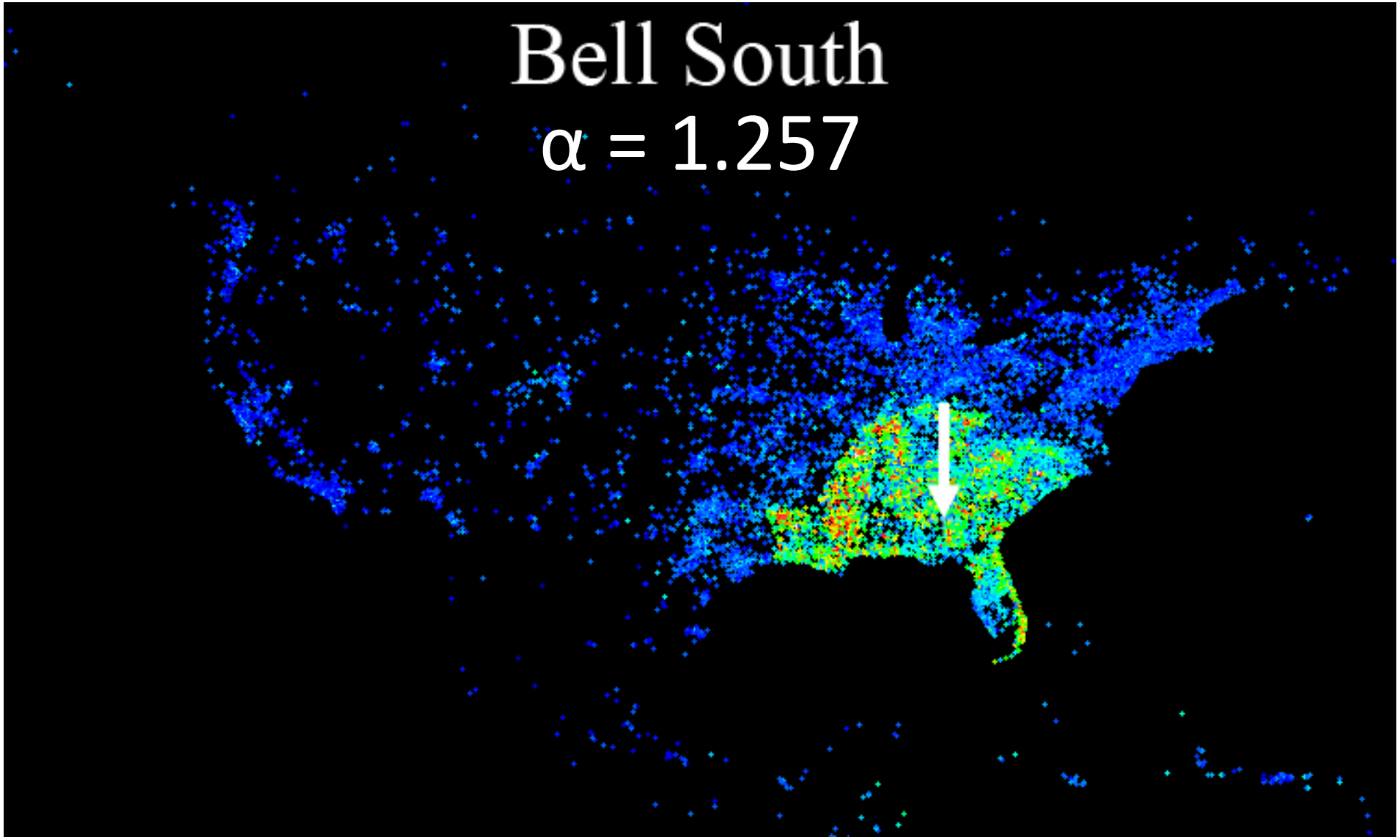


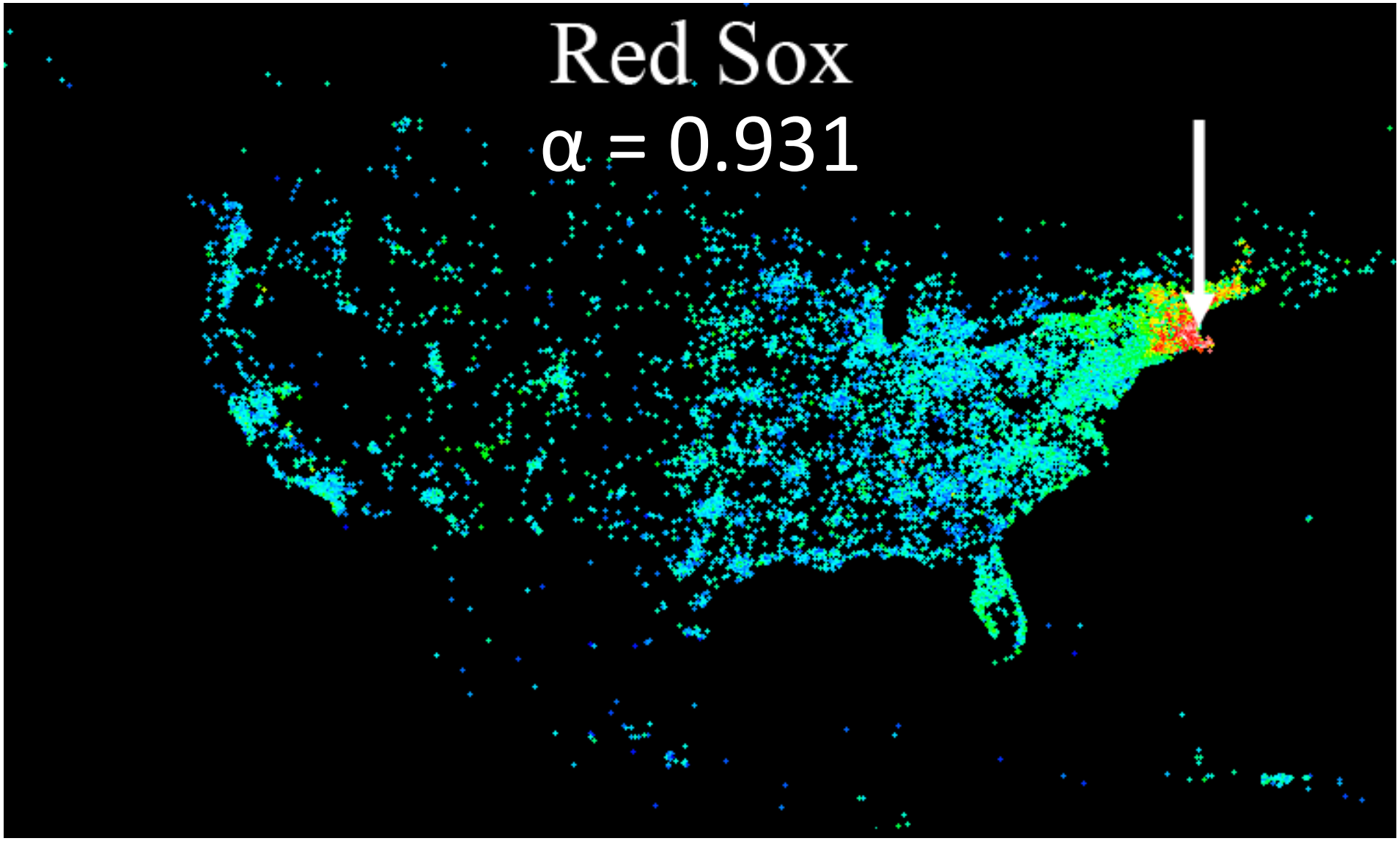
- Consider some query term *t*
  - e.g. 'red sox'
- For each location *x*, a query coming from *x* has probability p_x of containing *t*
- Our basic model focuses on term with a center "hot-spot" cell *z*.
  - Probability highest at z
  - p_x is a decreasing function of ||x-z||
- We pick a simple family of functions:
  - A query coming from x at a distance d from the term's center has probability $p_x = C\, d^{-\alpha}$
  - Ranges from non-local ($\alpha = 0$) to extremely local (large α)

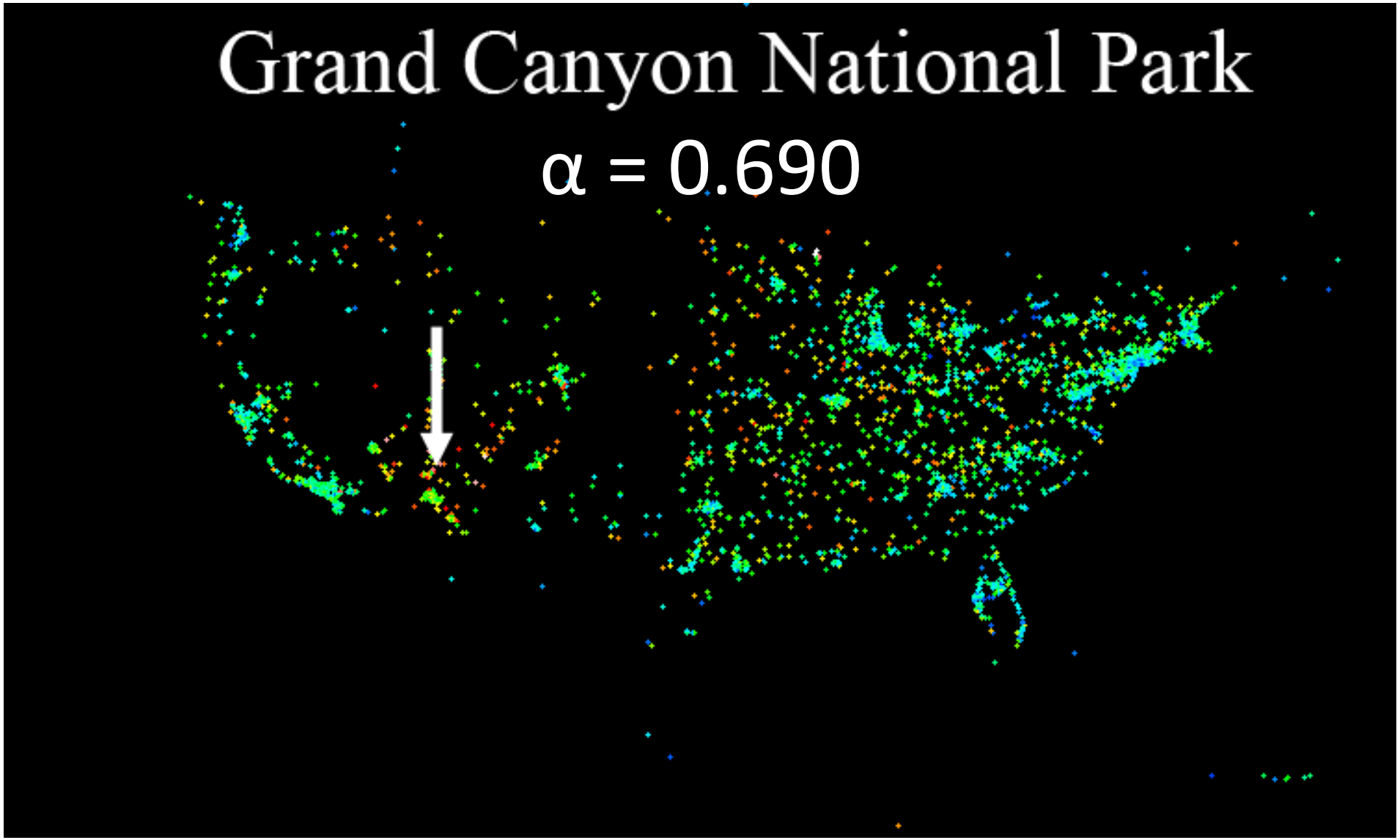
Algorithm

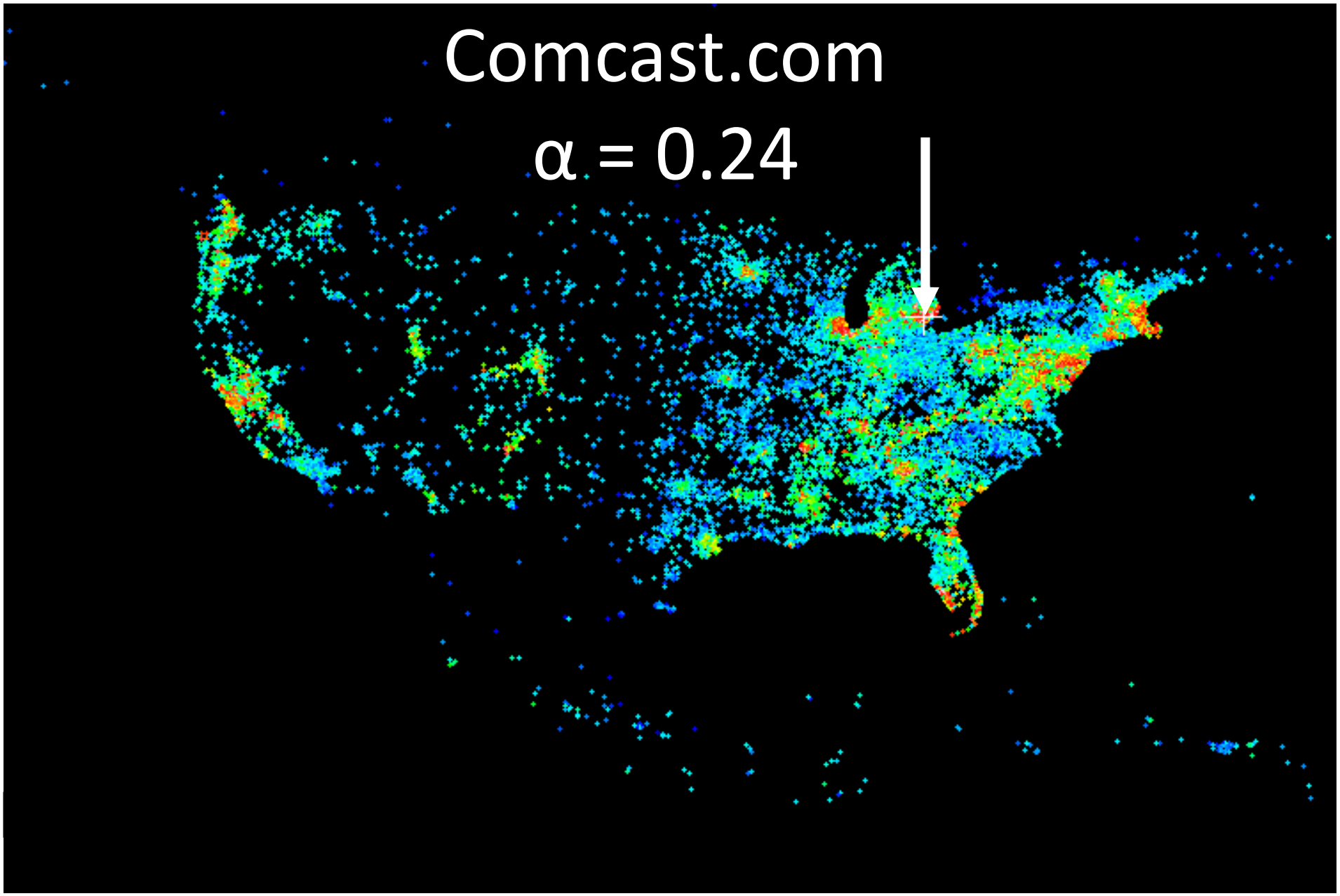
- Maximum likelihood approach allows us to evaluate a choice of center, C and α
- Algorithm finds parameters which maximize likelihood
  - For a given center, likelihood is unimodal and search algorithms find optimal C and α
  - Consider all centers on a course mesh, optimize C and α for each center
  - Find best center, consider finer mesh




Bell South
α = 1.257


Red Sox
α = 0.931


Grand Canyon National Park
α = 0.690


Comcast.com
α = 0.24

More Results (newspapers)

Newspaper	α
The Wall Street Journal	0.11327
USA Today	0.263173
The New York Times	0.304889
New York Post	0.459145
The Daily News	0.601810
Washington Post	0.719161
…	
Chicago Sun Times	1.165482
The Boston Globe	1.171179
The Arizona Republic	1.284957
Dallas Morning News	1.286526
Houston Chronicle	1.289576
Star Tribune (Minneapolis)	1.337356

- Term centers land correctly
- Small α indicates nationwide appeal
- Large α indicates local paper

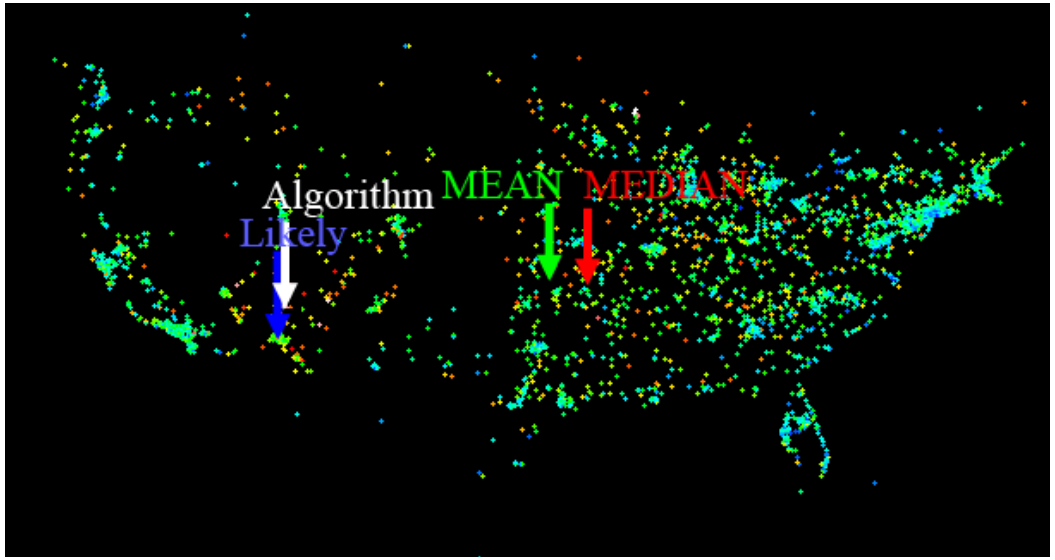
More Results

School	α
Harvard	0.386832
Caltech	0.423631
Columbia	0.441880
MIT	0.457628
Princeton	0.497590
Yale	0.514267
Cornell	0.558996
Stanford	0.627069
U. Penn	0.729556
Duke	0.741114
U. Chicago	1.097012

City	α
New York	0.396527
Chicago	0.528589
Phoenix	0.551841
Dallas	0.588299
Houston	0.608562
Los Angeles	0.615746
San Antonio	0.763223
Philadelphia	0.783850
Detroit	0.786158
San Jose	0.850962

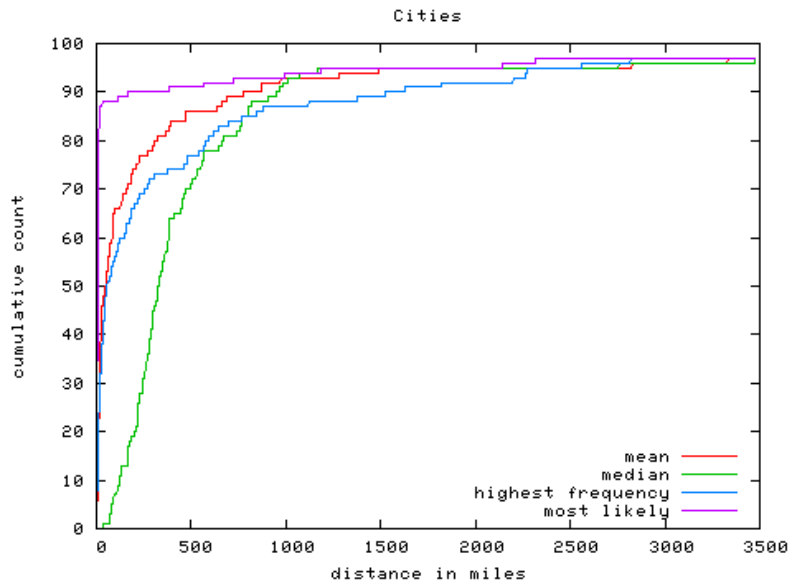
Evaluation

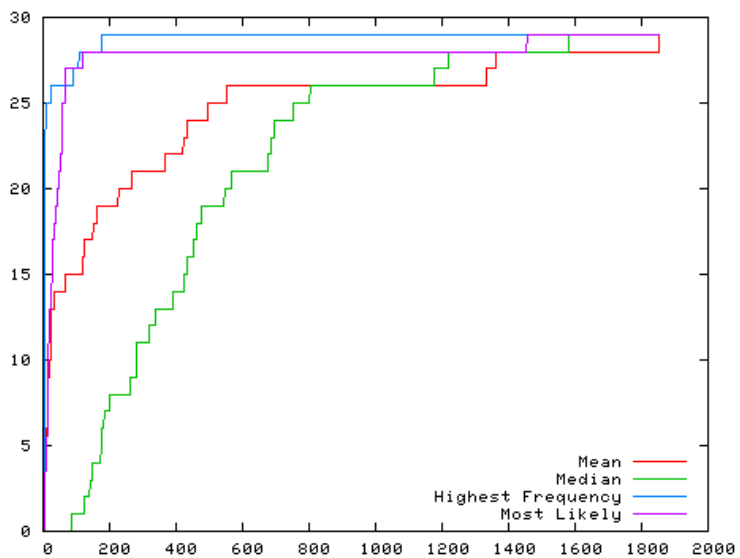
- Consider terms with natural 'correct' centers
  - Baseball teams
  - Large US Cities
- We compare with three other ways to find center
  - Center of gravity
  - Median
  - Most likely grid cell
    - Compute baseline rate for all queries
    - Compute likelihood of observations at each 0.1x0.1 grid cell



    - Pick cell with lowest likelihood of being from baseline model

Baseball Teams and Cities





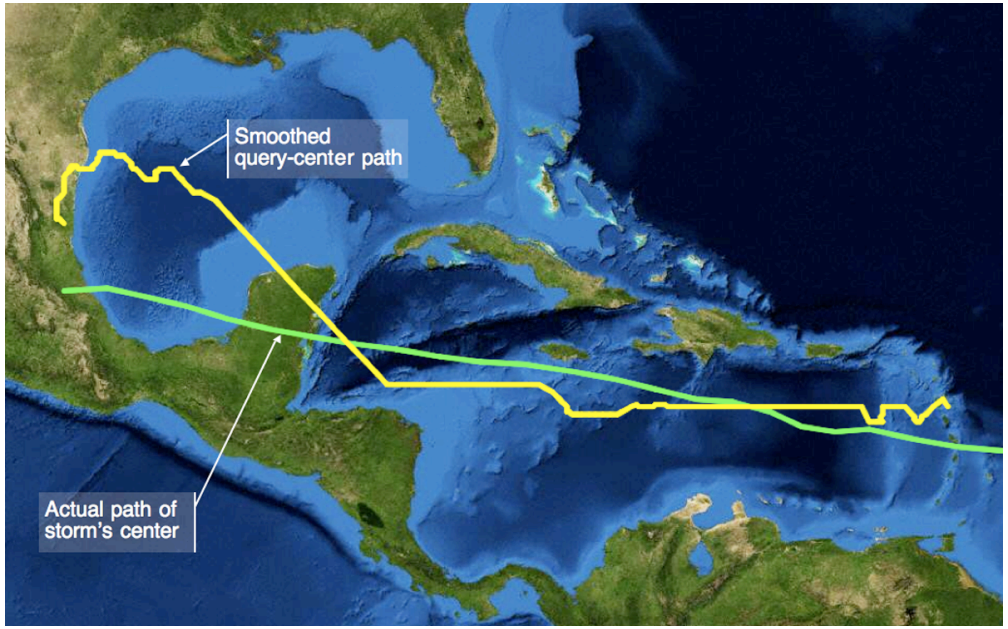
- Our algorithm outperforms mean and median
- Simpler likelihood method does better on baseball teams
  - Our model must fit all nationwide data
  - Makes it less exact for short distances

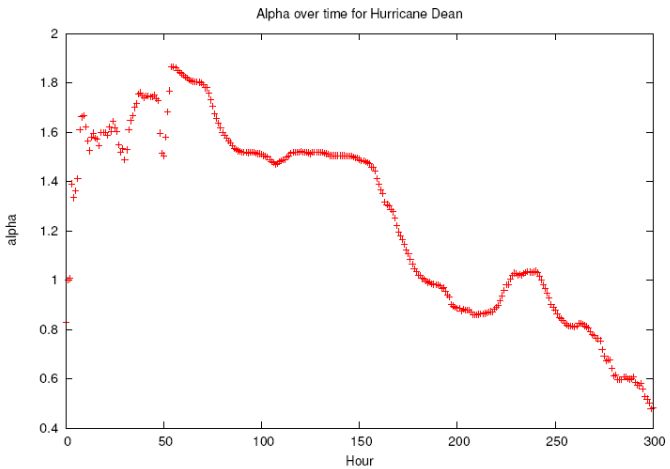
Temporal Extension

- We observe that the locality of some queries changes over time
  - Query centers may move
  - Query dispersion may change (usually becoming less local)
- We examine a sequence of 24 hour time slices, offset at one hour from each other
  - 24 hours gives us enough data
  - Mitigates diurnal variation, as each slice contains all 24 hours

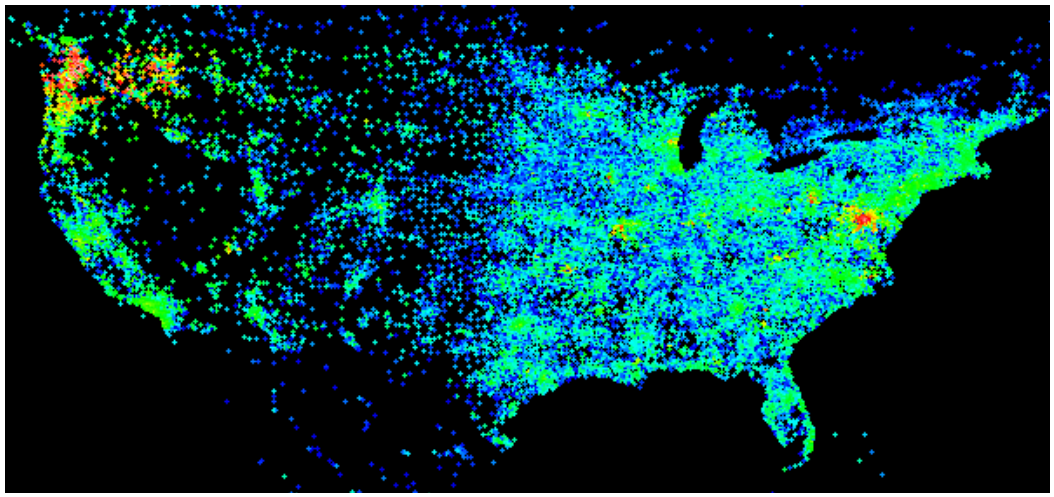
Hurricane Dean

- Biggest hurricane of 2007
- Computed optimal parameters for each time slice
- Added smoothing term
  - Cost of moving from A to B in consecutive time slices $\gamma|A-B|^2$
- Center tracks hurricane, alpha decreases as storm hits nationwide news





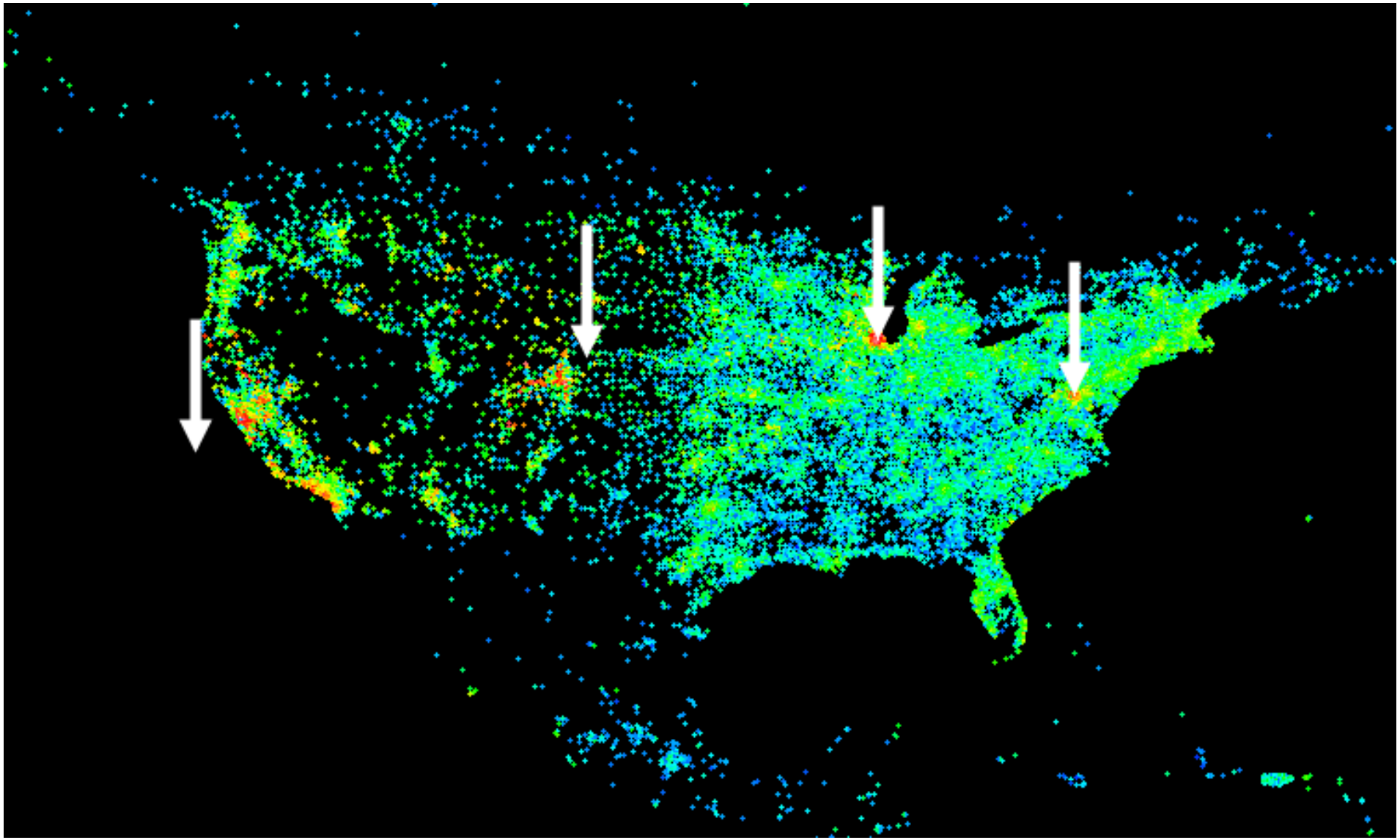
Multiple Centers



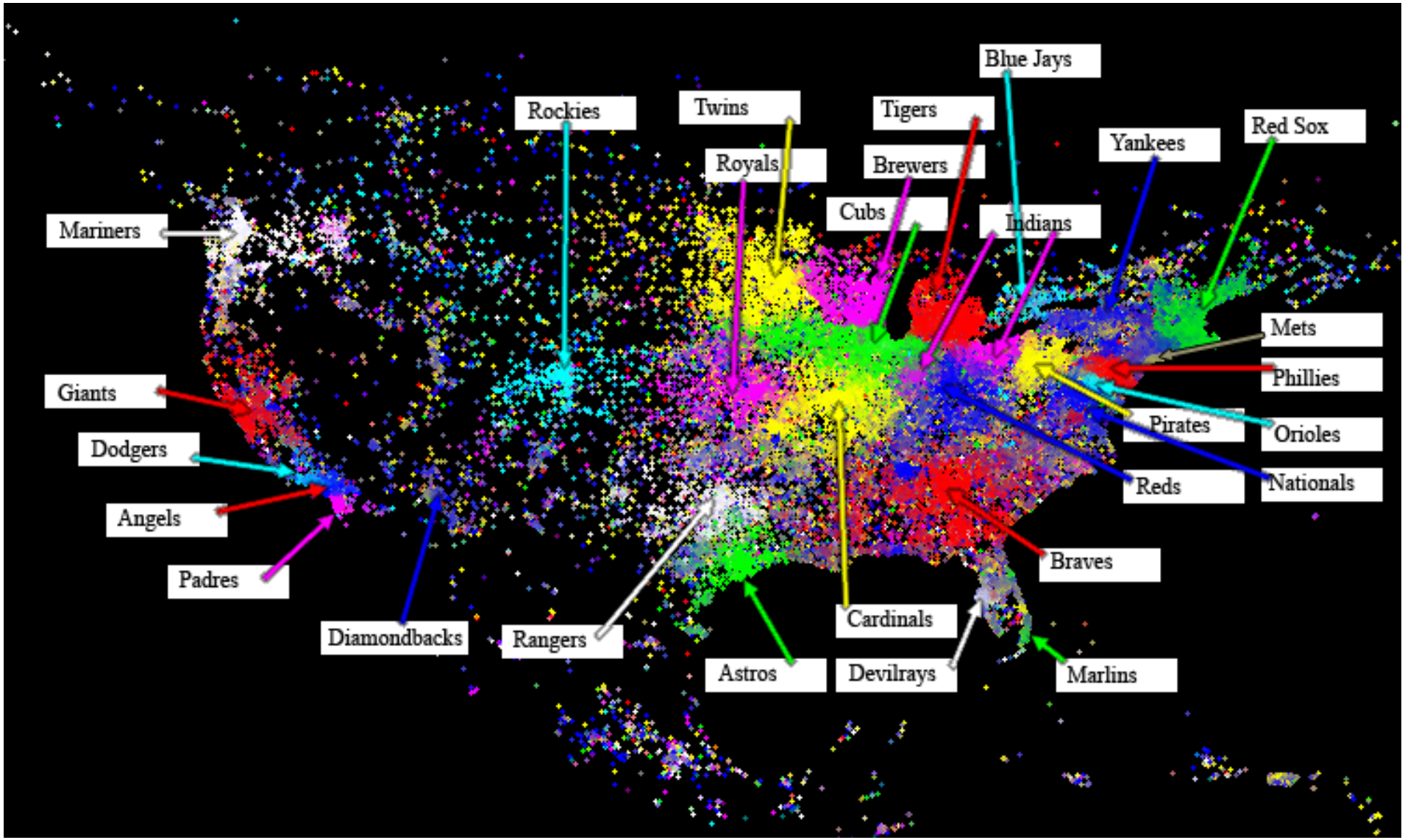
- Not all queries fit the one-center model
  - 'Washington' may mean the city of the state
  - 'Cardinals' might mean the football team, the baseball team, or the bird
  - Airlines have multiple hubs

- We extend our algorithm to locate multiple centers, each with its own C and α
  - Locations use the highest probability from any center
  - To optimize:
    - Start with K random centers, optimize with 1-center algorithm
    - Assign each point to the center giving it highest probability
    - Re-optimize each center for only the points assigned to it

United Airlines

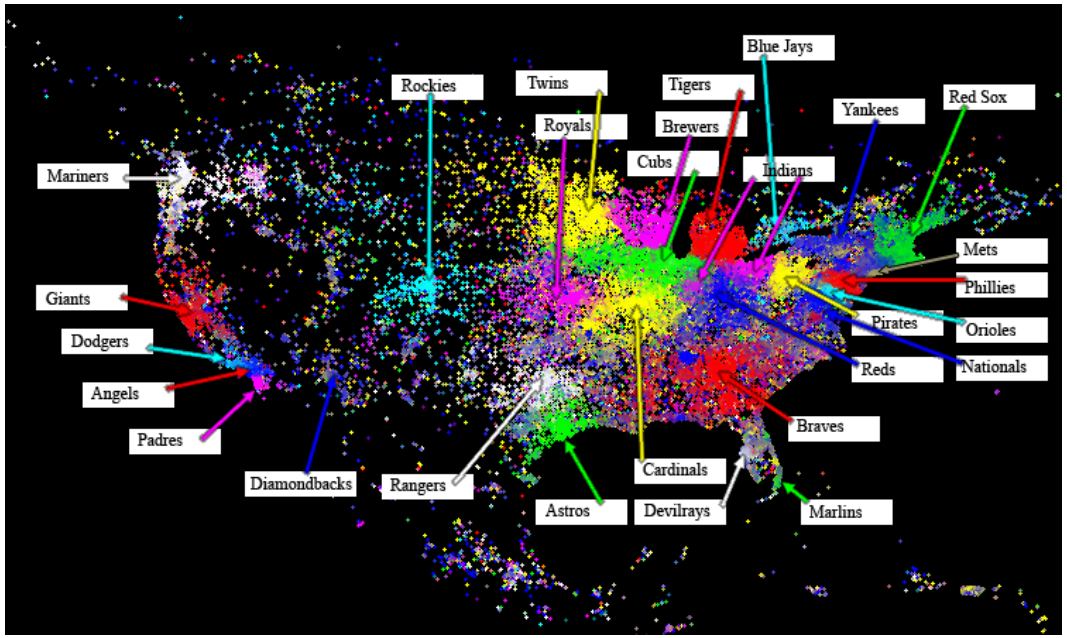


Spheres of influence



Spheres of Influence

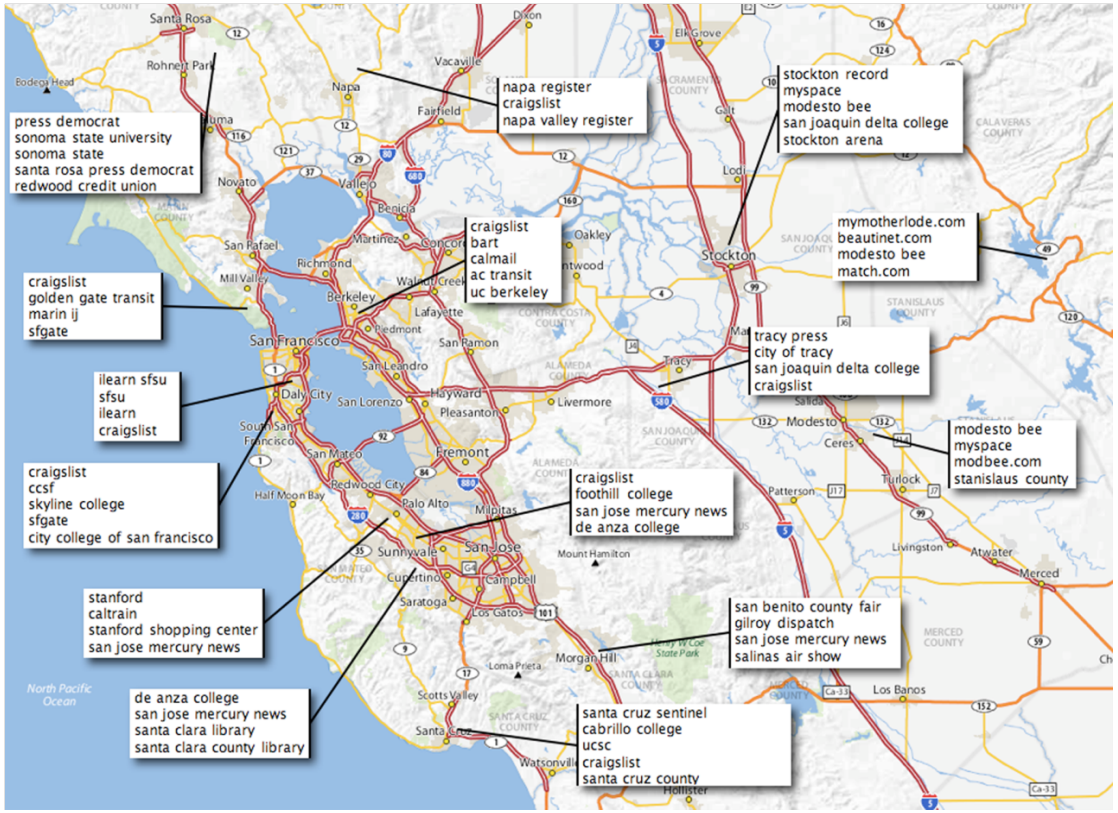
- Each baseball team assigned a color
- A team with N queries in a cell gets N^C votes for its color



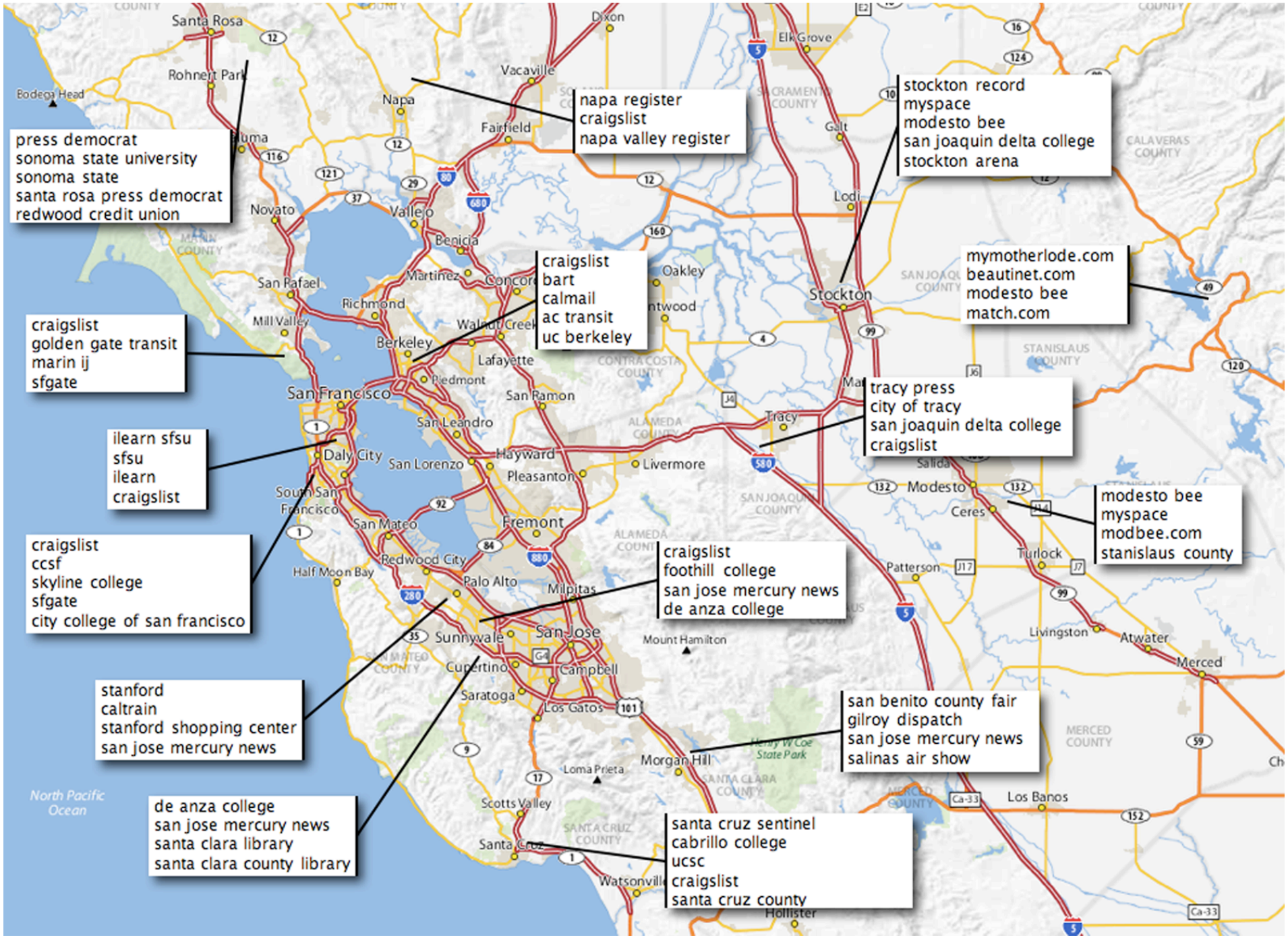
- Map generated be taking weighted average of colors

Distinctive Queries

- For each term and location
  - Find baseline rate *p* of term over entire map
  - Location has t total queries, s of them with term
  - Probability given baseline rate is: $p^s(1-p)^{t-s}$



- For each location, we find the highest deviation from the baseline rate, as measured by the baseline probability


Santa Rosa
Rohnert Park
Bodega Head
Petaluma
Novato
Vallejo
Napa
Vacaville
Fairfield
Dixon
Elk Grove
Galt
Lodi
Benicia
Martinez
Concord
Oakley
Antioch
Stockton
San Rafael
Richmond
Mill Valley
Walnut Creek
Berkeley
Lafayette
Piedmont
San Francisco
San Ramon
San Leandro
Tracy
Daly City
San Lorenzo
Hayward
Livermore
Pleasanton
Salida
Modesto
Ceres
South San Francisco
San Mateo
Fremont
Redwood City
Half Moon Bay
Palo Alto
Milpitas
Patterson
Turlock
Livingston
Atwater
Merced
Sunnyvale
San Jose
Mount Hamilton
Cupertino
Campbell
Saratoga
Los Gatos
Morgan Hill
Loma Prieta
Scotts Valley
Santa Cruz
Watsonville
Hollister
Los Banos
North Pacific Ocean
napa register
craigslist
napa valley register
stockton record
myspace
modesto bee
san joaquin delta college
stockton arena
press democrat
sonoma state university
sonoma state
santa rosa press democrat
redwood credit union
craigslist
bart
calmail
ac transit
uc berkeley
mymotherlode.com
beautinet.com
modesto bee
match.com
craigslist
golden gate transit
marin ij
sfgate
tracy press
city of tracy
san joaquin delta college
craigslist
ilearn sfsu
sfsu
ilearn
craigslist
modesto bee
myspace
modbee.com
stanislaus county
craigslist
ccsf
skyline college
sfgate
city college of san francisco
craigslist
foothill college
san jose mercury news
de anza college
stanford
caltrain
stanford shopping center
san jose mercury news
san benito county fair
gilroy dispatch
san jose mercury news
salinas air show
de anza college
san jose mercury news
santa clara library
santa clara county library
santa cruz sentinel
cabrillo college
ucsc
craigslist
santa cruz county

Outline

- Query Logs
  - Probabilistic, generative model of queries
  - Results and evaluation
  - Adding temporal information to the model
  - Modeling more complex geographic query patterns
  - Extracting the most distinctive queries from a location
- Facebook GEO-Data
  - Understanding data
    - Where do people live?
    - How far away do their friends live?
    - How are the two related?
  - Location Algorithm
    - Can we predict locations from friends' locations?
    - Using the network to improve beyond prediction based on located friends' locations.
- Conclusions

Geo-properties of Friendship

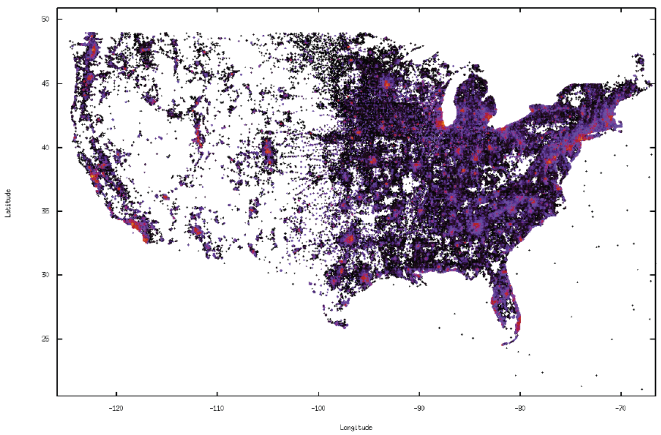
- Using Facebook data, we explore the geographic properties of social networks
  - How well do link distance distributions match existing theories?
  - How does it depend on where people live (urban vs. rural)?
- Expanding beyond user-provided addresses
  - Can we use what we've learned about geography to predict unknown location from sparse geo-data
  - How can network data improve location data

Related Work - Routing

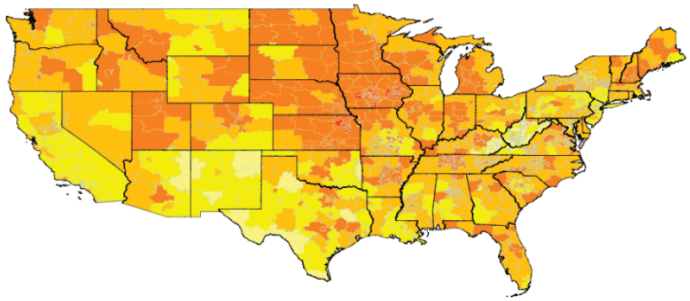
- Long history of link between friendship and geography
  - Milgram experiment (1967) showed short paths between people
  - Watts and Strogatz (1998) showed theoretical models of how this might work
  - Kleinberg (1999) showed intuitive geographic properties could lead to small world
    - Lattice network with random links distributed by distance
  - Liben-Nowell et al. (2005) showed some of these properties on LiveJournal network

US FB Population

- We have two ways to determine location
  - From IP Address
    - Lots of inaccurate, bad data
  - From user provided address
    - Also lots of inaccurate, bad data
    - But, more precise when correct
- Using user provided data, we can map FB usage
  - 3 Million US users provided this data
  - More interesting when compared to US Population
    - Surprisingly midwest is highest



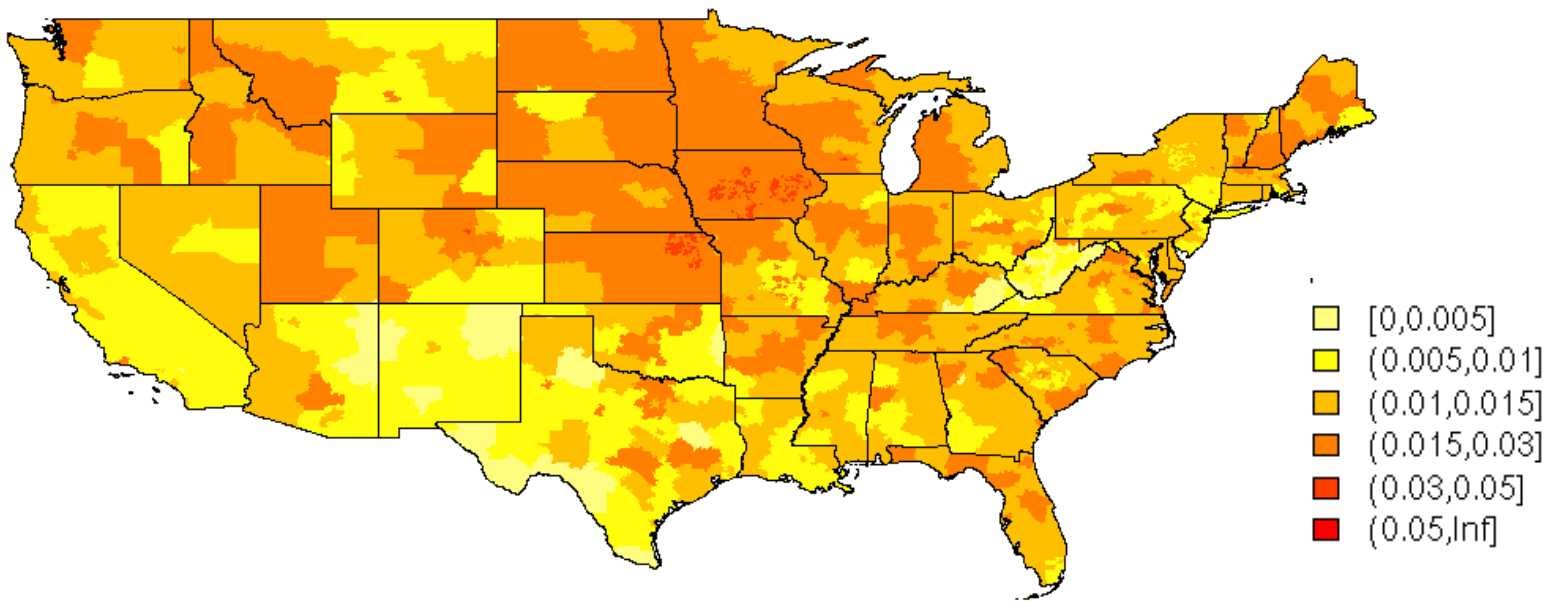
FB Density



FB Density, normalized by Popluation

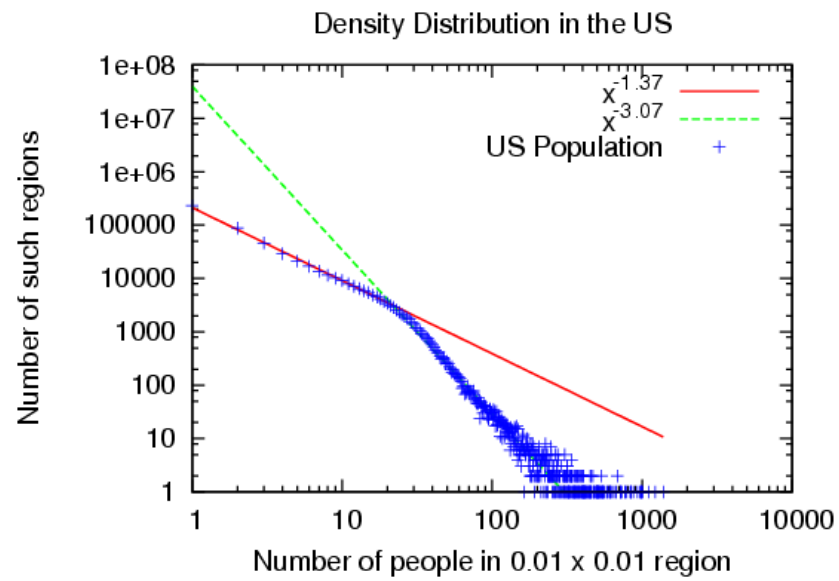
User Data

- User provided addresses mapped to lat,long using TIGER/Line data from US Census bureau
  - Only addresses that can be mapped to real street addresses used



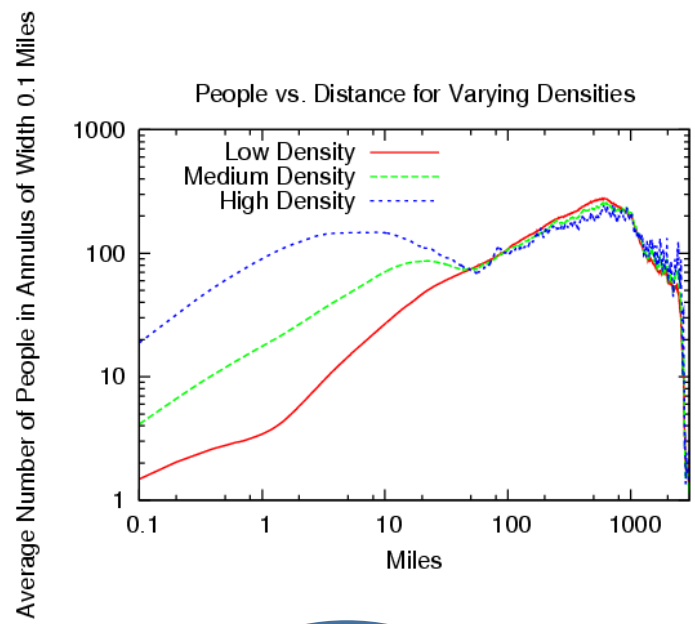
Population Density

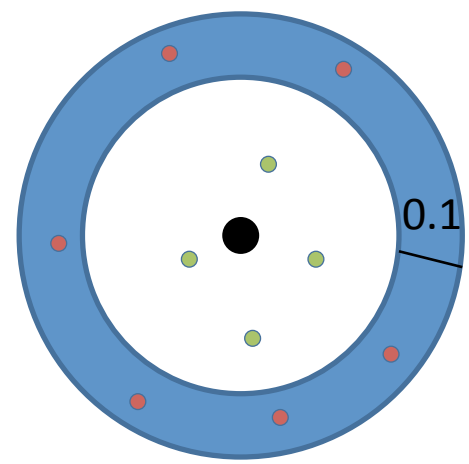
- Consider each 0.01x0.01 degree (~0.36 sq. miles) region in the US
  - How many FB users are there in each region?
  - Distribution shows two distinct behaviors on log-log plot
    - Rural Region, slow falloff
    - Urban Region, fast falloff
    - Transition at 560K^{2}' / FB user or about 5,600^{2}' / person



Population Density

- We divide the US into three regions
  - Low, medium and high density – 1/3 of users in each
- For each person, we count people in annulus of width 0.1 mile and radius r
  - High density regions have more people nearby
  - Curves converge at 50 miles
  - High density curve goes down as we transition urban to rural

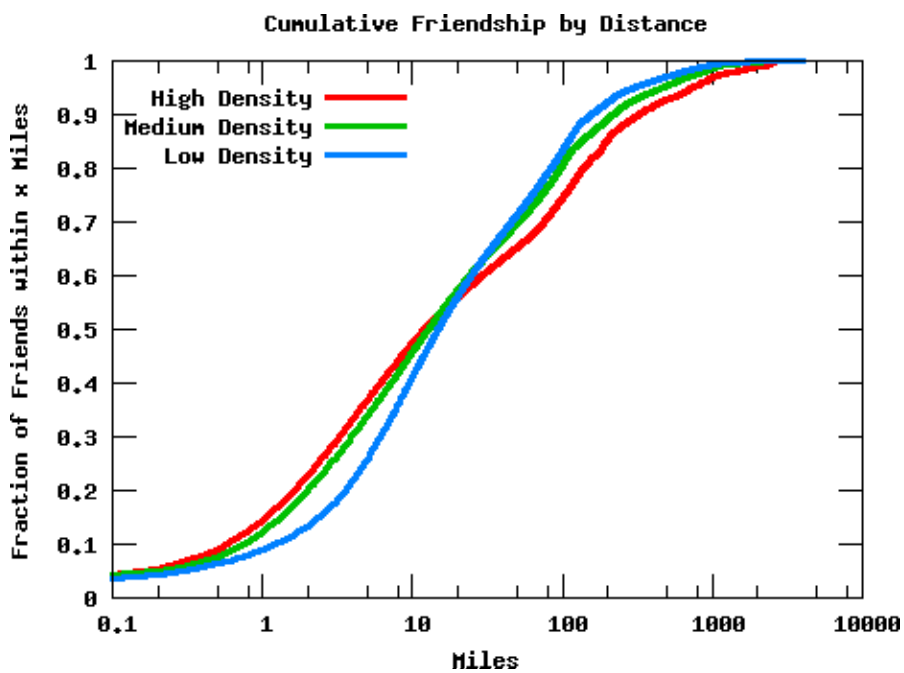




Where our friends live

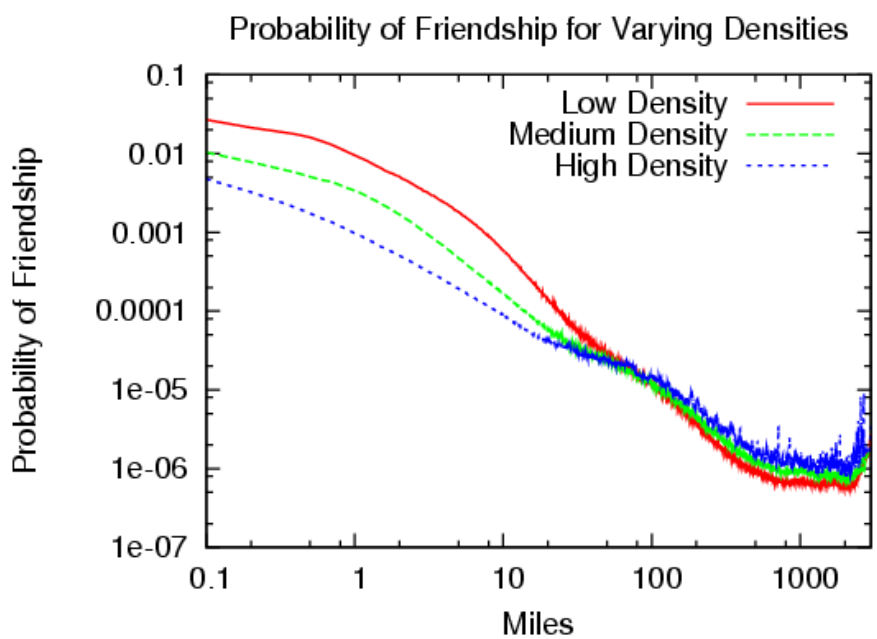
- Mostly nearby, but a significant number far away too
  - 20% within 2 miles
  - 50% within 12 miles
  - 20% over 100 miles
- Population density plays a role

	20%	50%	80%
Low Density	3.5	14.8	83.6
Medium Density	2.1	12.7	97.1
High Density	1.6	11.9	137.1



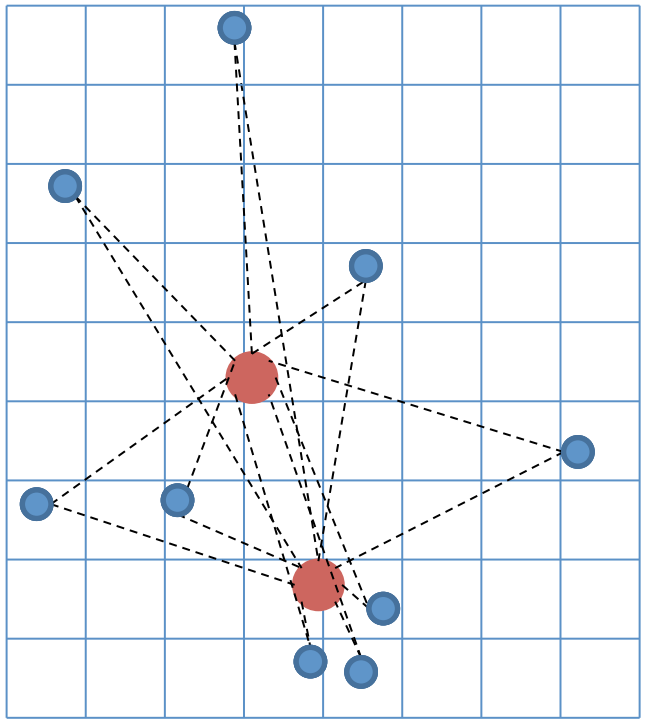
Probability of Friendship

- What is the probability that you know someone x miles away?
  - Count the number you know x miles away
  - Count the total number of people x miles away
  - Divide and aggregate over all users
    - Tail of curve is roughly x^{-1}
- Breaking down by density
  - Higher density means lower probabilities at lower distances
    - Larger denominator
  - At ~100 miles all curves converge
  - Beyond 100 miles, urban users have higher probability
    - More cosmopolitan?



Using Friends to Geolocate

- For some people (about 3%) we have precise user-provided locations
- How well can we infer the locations of the rest based on friendship?
  - We observed the probability of friendship as a function of distance (about x^{-1})

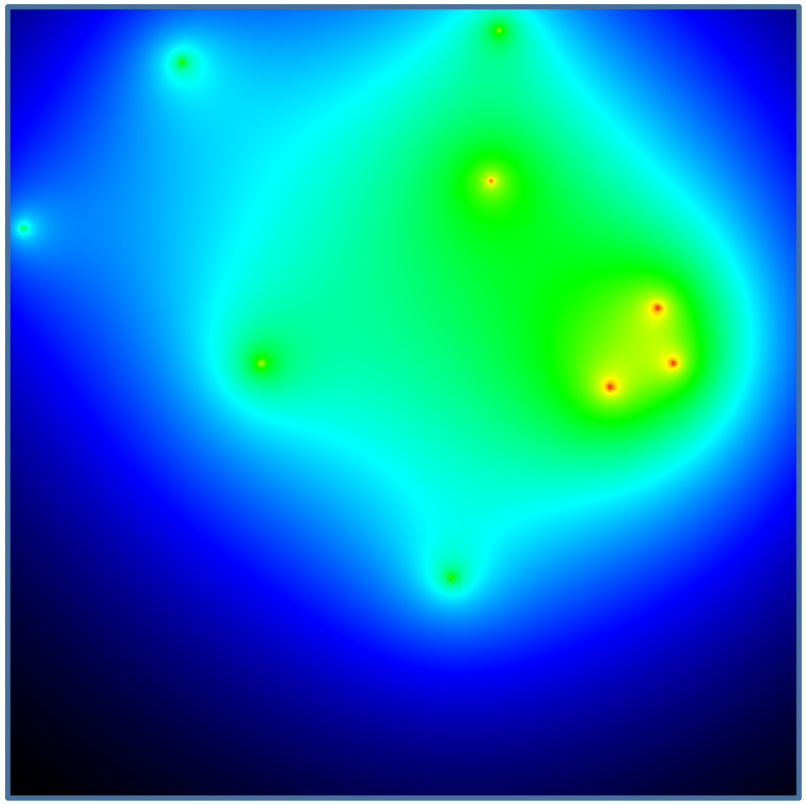


Location Inference Problem

- Problem Statement:
  - Input: user u, with friends $\{v_1, v_2, \ldots, v_k\}$ where v_i is located at location p_i
  - Output: an estimate for the location of u
- Friendship probabilities suggest generative model
  - Assume that users u and v are friends with probability f(distance(u,v))
  - f(.) learned from observations of users with known geography – about x^{-1}
- Pick location to maximize likelihood

Pinpointing Location

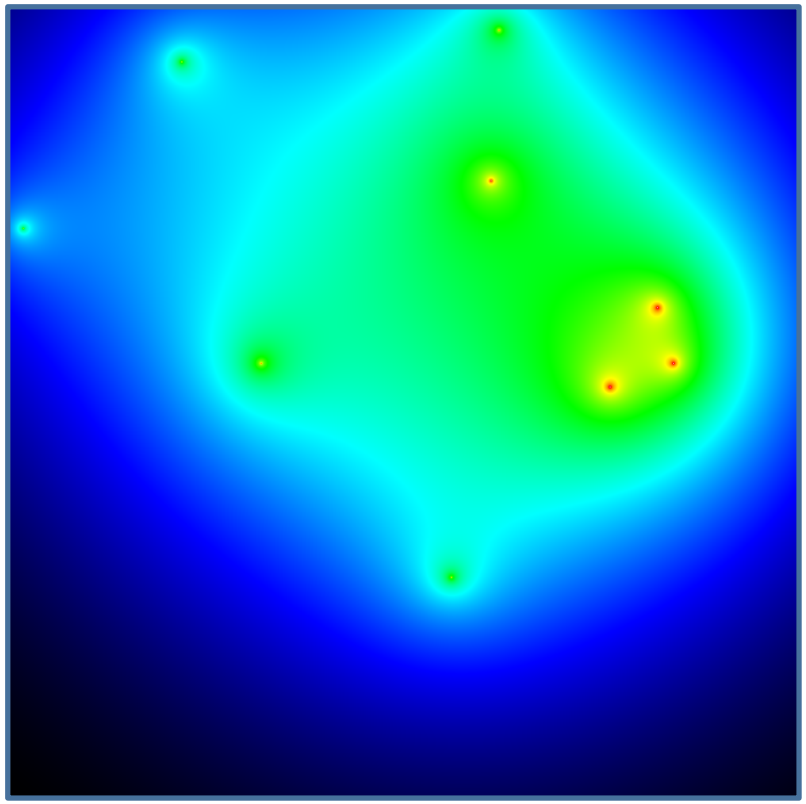
- Given f(.) can easily compute likelihood of location L
  - Π_i f(distance(L,p_i)) * $\Pi_{\sim i}$ 1-f(distance(L,p_i))
    - First term pulls towards friends
    - Second term pulls towards sparsely populated regions
  - {i} is the set of friends, {~i} is the set of non-friends
- Naively this is expensive
  - Many choices for L
  - {~i} is huge



Likelihood Surface

Pinpointing Location

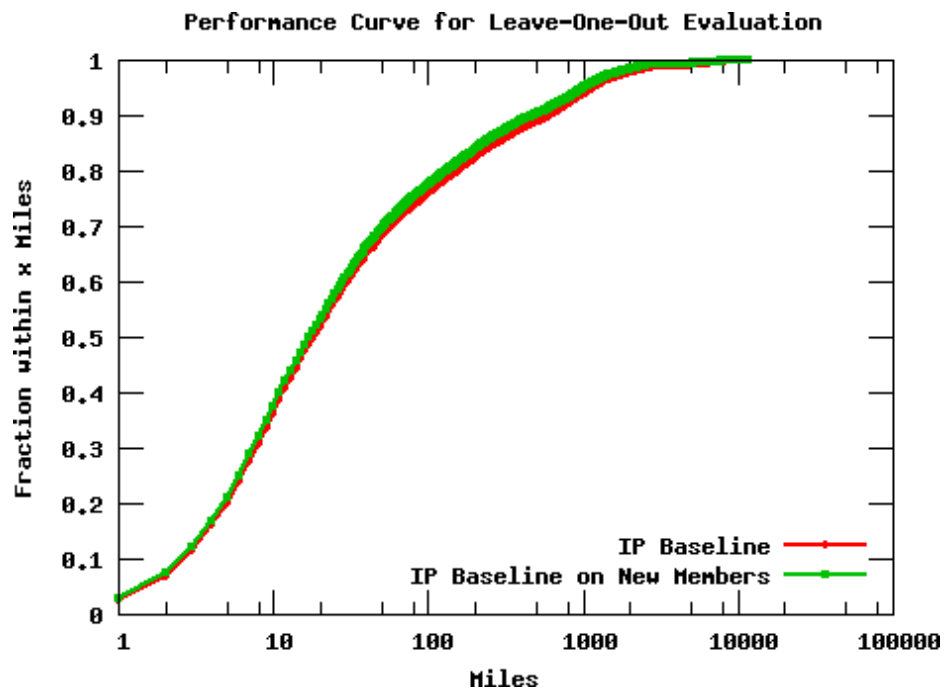
- Simple optimizations make this practical
  - Product over all non-friends can be pre-computed one time
  - Optima is (almost) always colocated with a friend
- Algorithm
  - For all L, precompute Π_{users} 1-f(distance(L,p_i))
  - For each friend, compute likelihood at friend's location
  - Pick maximum



Likelihood Surface

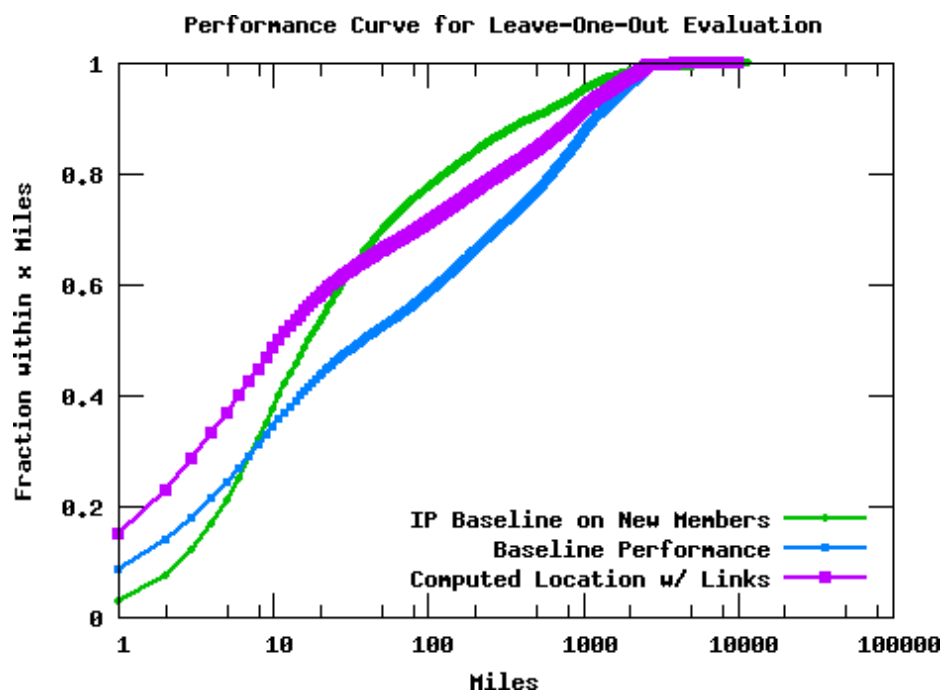
Prediction Performance

- Performance evaluated at x miles
  - What fraction of users are placed within x miles of their true location
  - True location comes from user-provided addresses
- Performance of IP derived locations
  - 57% within 25 miles
  - 76% within 100 miles
  - Performance on new users/recently updated addresses not much better
    - Measure for people with new addresses in last 90 days
    - Suggests errors come from IP-Location, not people moving



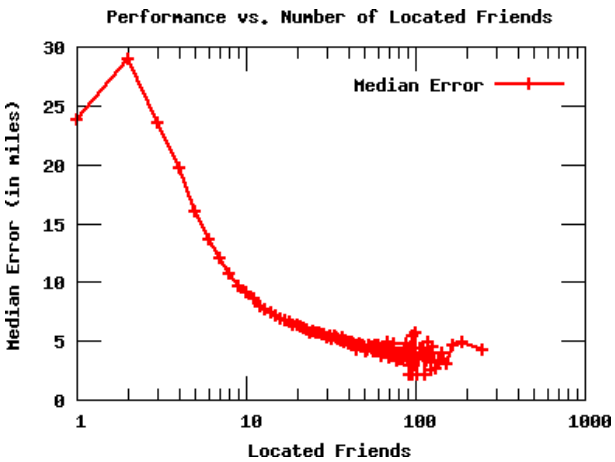
Prediction Performance

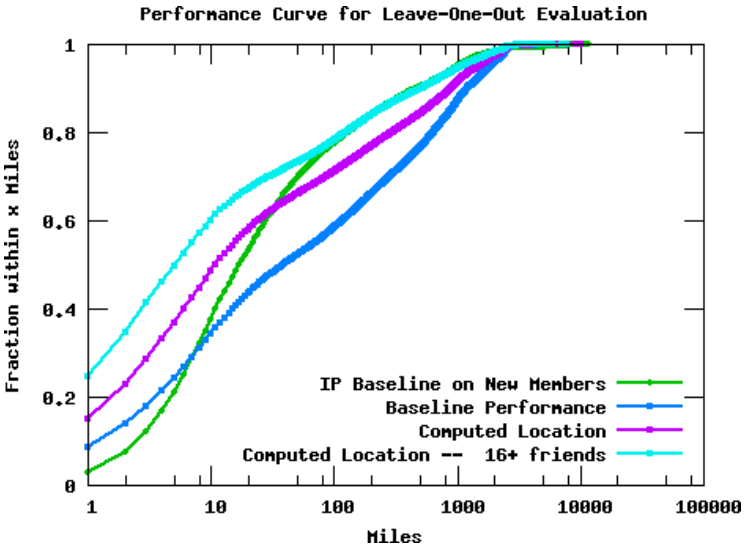
- A simple network-based baseline comes from placing a person at the location of a random friend
  - Does better than IP at short distances
    (IP resolution is rarely better than city-level)
  - Much worse overall
- Our algorithm is better at shorter distances
  - 61% (vs. 57%) at 25 mi.
- It is worse at greater distances
  - 71% (vs 76%) at 100 mi.



Prediction Performance

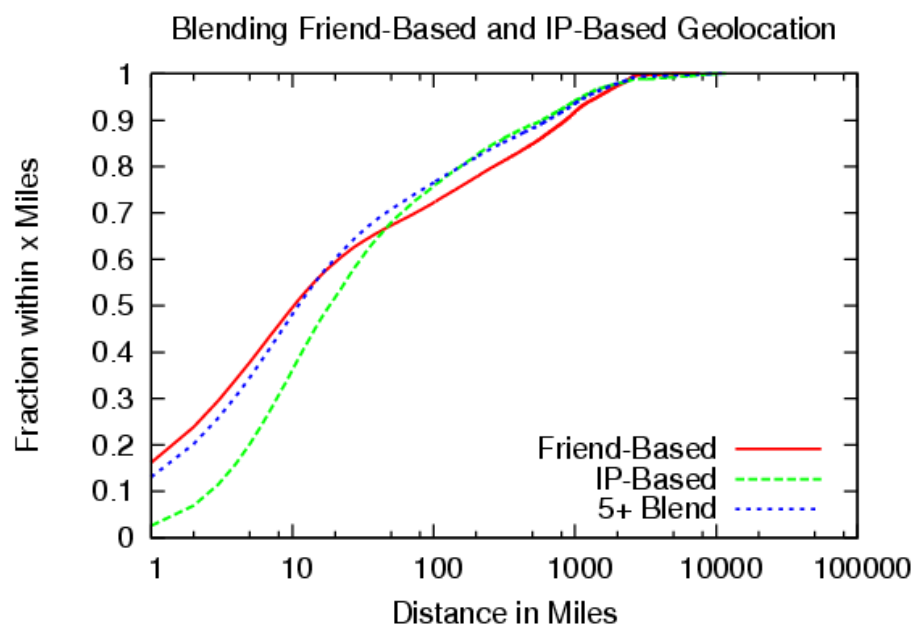
- We predict more accurately for people with more friends
  - Median error with 5 located friends is 16 miles
  - With 16 friends, it is 6.8 miles
- Error on users with 16 or more located friends is better at all distances
  - 69% (vs. 57%) at 25 miles
  - 78% (vs 76%) at 100 miles





Hybrid Algorithm

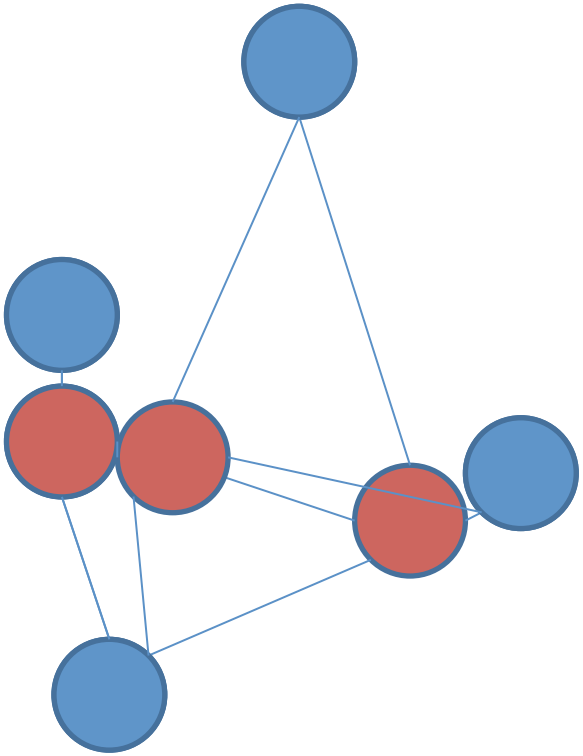
- Results suggest hybrid algorithm



  - Less than 5 friends with user-provided location, use IP-Location
  - Otherwise use friend-based location
  - Strict improvement over IP on this metric

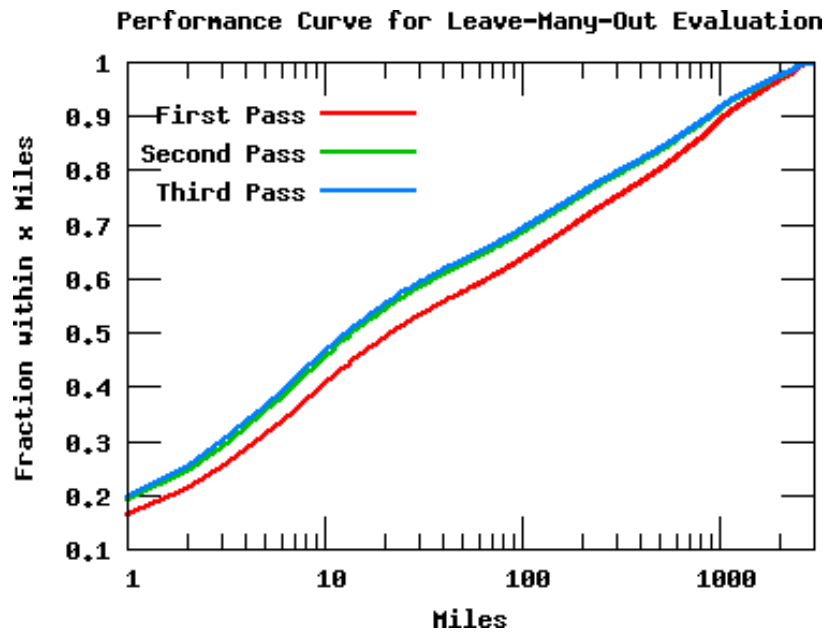
Using the Network Further

- Use prediction for u to improve prediction for u's friends
  - Ideally, maximize global likelihood
    - Hard problem, can't be easily solved
  - Repeat our algorithm multiple times
    - Run once to get first approximations
    - Subsequently, optimize based on known and approximated locations



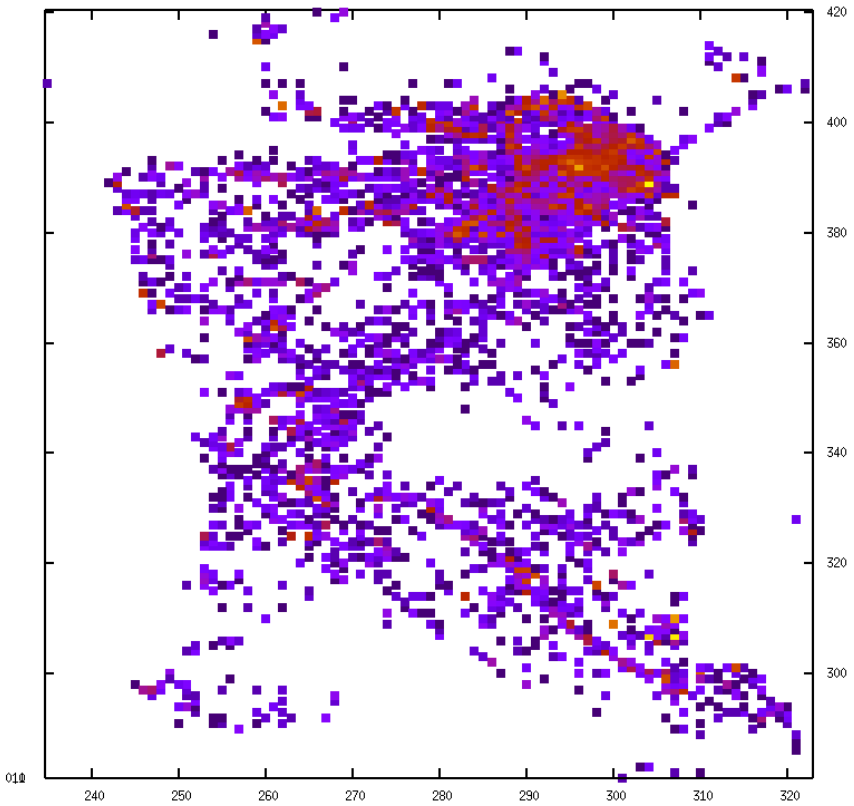
Testing Iterative Method

- To test, remove locations from ½ of users
- Predict locations for those users based on other ½
- Performance curve shows improvement
  - Second pass significantly better
  - Third pass shows little improvement



Facebook Checkins

- Most popular places to checkin
  - Stadium
  - Airport
  - Fisherman's wharf

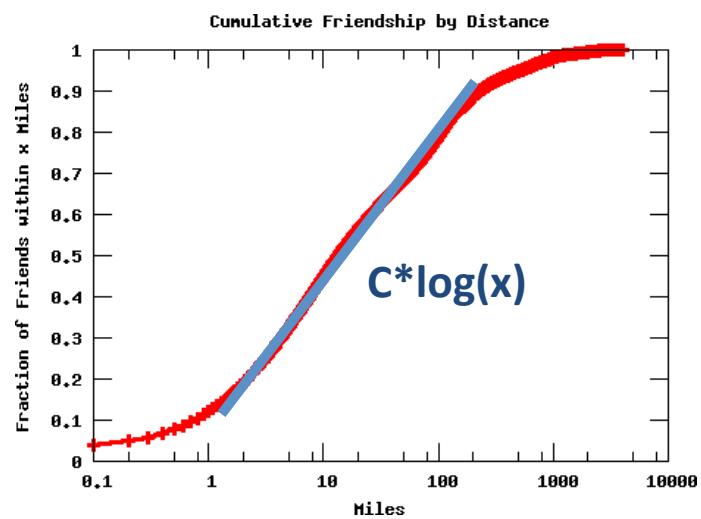


Outline

- Query Logs
  - Probabilistic, generative model of queries
  - Results and evaluation
  - Adding temporal information to the model
  - Modeling more complex geographic query patterns
  - Extracting the most distinctive queries from a location
- Facebook GEO-Data
  - Understanding data
    - Where do people live?
    - How far away do their friends live?
    - How are the two related?
  - Location Algorithm
    - Can we predict locations from friends' locations?
    - Using the network to improve beyond prediction based on located friends' locations.
- Conclusions

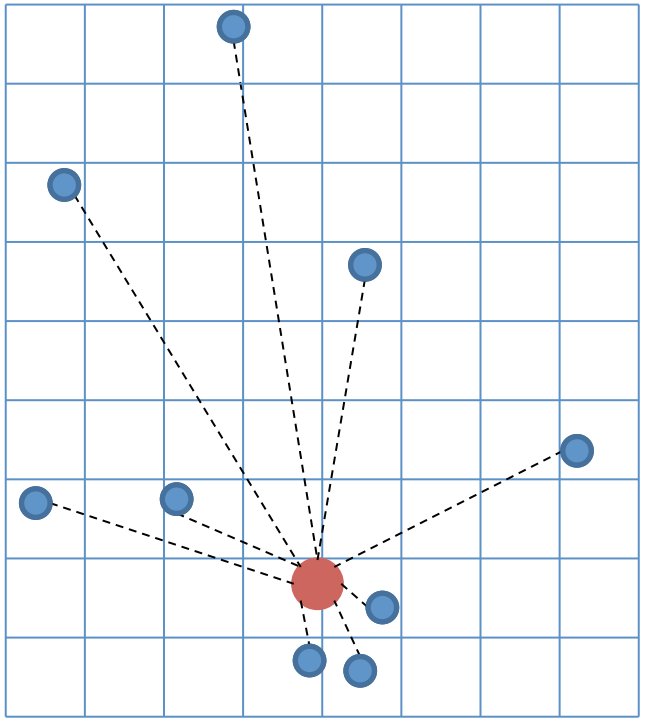
Conclusions

- Large-scale study of social geography confirms some observations made one small scales
  - Most friends live within about 10 miles
  - People in urban areas have more long-range ties
- Also, lends further credence to geographic explanations of social routing
  - Kleinberg model links to O(x) nodes at distance x with prob. $1/x^2$ each
    - Cumulative distribution of friends is C*log(x)



Conclusions

- Using known locations from just a few people, we can bootstrap approximations for the rest
  - Allows us to serve better local content to users
- Future work
  - Use edge creation times (more weight on new edges)
  - Explore different meanings of distance
    - Is the social distance of NY-LA more than NY-Rapid City, SD?



Questions