


The logo for Identified, featuring the word "Identified" in a bold, dark blue sans-serif font. The letter "I" is stylized with an orange square at its top and a white circle at its base.

Learn how Serial Metrics created the  
geo-inferencing system for  
[Identified.com](https://www.identified.com)





Identified is a data and analytics company, based in San Francisco that uses Facebook data to provide professional insights. The company is recognized for developing SYMAN (Systematic Mass Normalization), an artificial intelligence technology to identify professionals that would otherwise not be available to employment recruiters. The technology analyzes data from sources like Facebook and filters out relevant professional information that can assist in the recruitment process.

The company was in search of a reliable way to detect the location of profiled users for the purpose of recommending jobs to prospective employees that are within a commutable distance.

## LOCATION-BASED RECOMMENDATIONS

Imagine if you knew the location of your users and could offer them the right products or services at the right time. Without an address or other positioning information, you cannot target the right demographic with location-specific offers.

Identified.com asked Serial Metrics to build a geo-inferencing system, which could detect the residence of 'scraped' Facebook accounts and then surface job recommendations only within a commutable distance.

A geo-inferencing system can detect customer location by predicting whether a user lives near the largest cluster of their friends. It solves the problem of identifying user location, and related demographics, when users fail to provide an address or place of residence.

This system was able to take 700,000 Facebook accounts with known location data, and infer the location of 50 million Facebook accounts that had no corresponding location data.



## CLUSTERING LOCATIONS

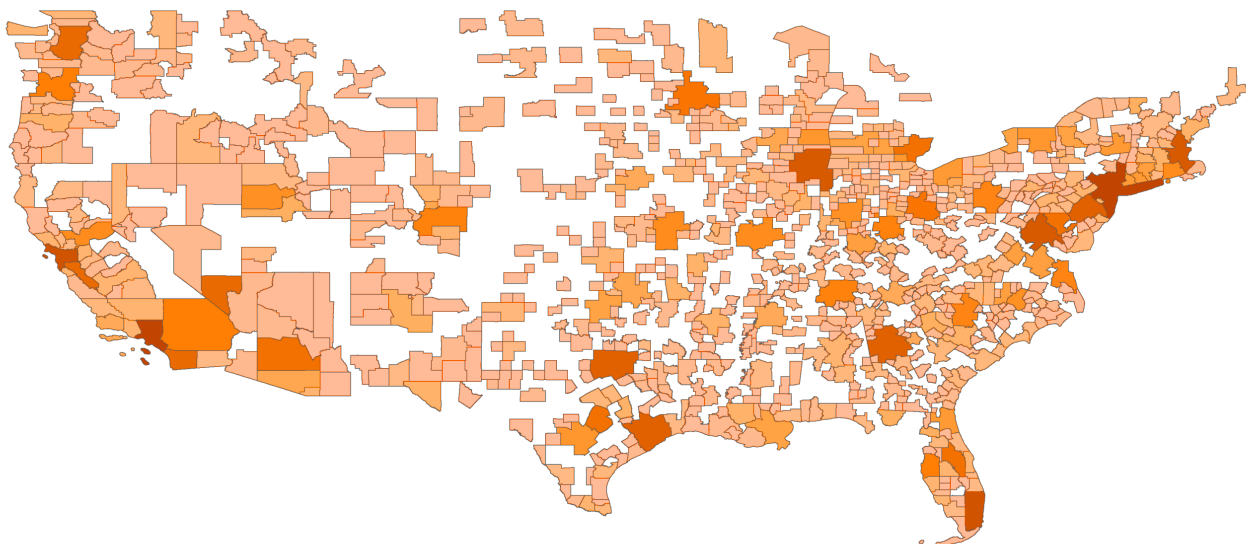
We start with 1MM user IDs and their corresponding, unique, verified, lat/long coordinates.

We split the 1MM user IDs into two groups: 70% for training and 30% for model verification.

We then group the location coordinates into meaningful geographic regions reducing the space of predicted outcomes from 50 MM to about 929 possible outcomes.

These geographic regions or Metropolitan Statistical Areas, correspond to sectors with multiple cities/towns near a single large city that wields substantial influence over the region.

The next step is to map the location coordinates, both verified and unverified, to their corresponding MSA, which allows us to equip our model with demographic data that we have collected for each user.



0 2 4 8 Decimal Degrees

MSA

## START WITH A LIST OF USERS, INCLUDING A PARTIAL SET WITH LOCATION DATA

Serial Metrics processed 50MM user IDs. 4% with verified location, and 96% without location data. The goal was to use a subset of the 4% known locations to predict the location of the remaining 96%.



## CLUSTER LOCATIONS

We first group the lat/long data in some intelligent way to limit the number of model outcomes, since lat/long data were all unique.



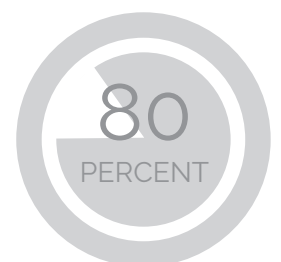
## CREATE A DYADIC DATASET

Join the list of user IDs with the list of corresponding Friend IDs to establish a dyadic dataset. The dyad will likely live, work, or attend school near each other.



## BUILD A GEO-INFERENCING SYSTEM

Group dyads into clusters of friends that map to specific MSAs. Rank clusters by size. Then evaluate if a user for whom we have an unverified lat/long pair, lives in the same MSA as the largest cluster of friends for whom we have a verified lat/long pair.



## PRODUCTION CODE

A server-side application that ingest a sets of user IDs, and features describing the user-friend dyads, then returns a location prediction (at the MSA-level) for any user that is missing a verified location co-ordinate pair. To ensure speed and scalability, the system we built was written in C++





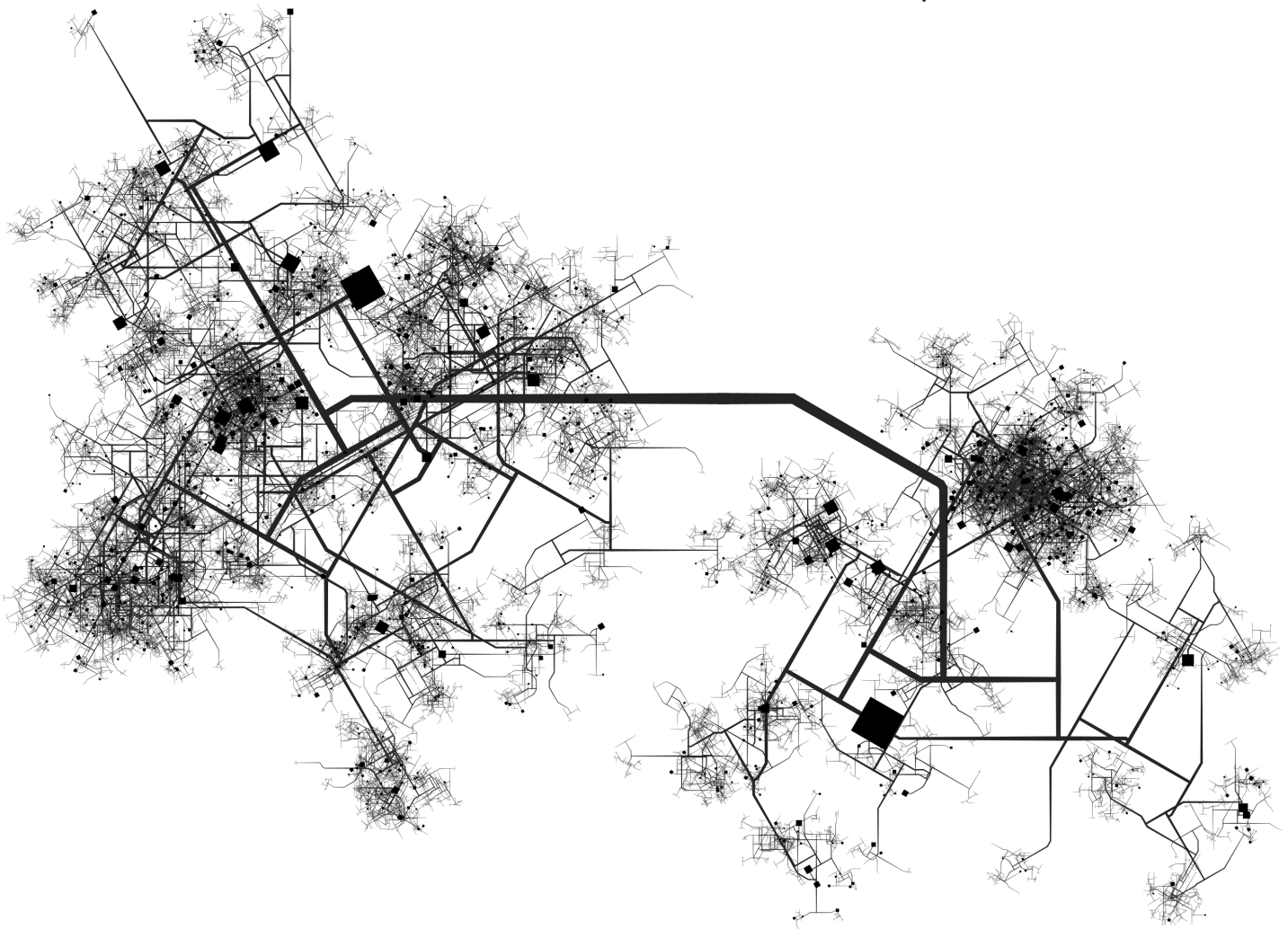
## ESTABLISH A HYPOTHESIS

To predict the location of any unplaced user we rely on the total network users, and the interactions between repeatedly sampled pairs of users (or dyads), from which we infer “relatedness.”

A handful of known locations belonging to a large, tightly clustered and closed network is exponentially more informative than a set of users without corresponding network information.

Even if a user’s location is unknown, if we know the location of the user’s largest cluster of friends, we can infer the location of the user with an unknown location.

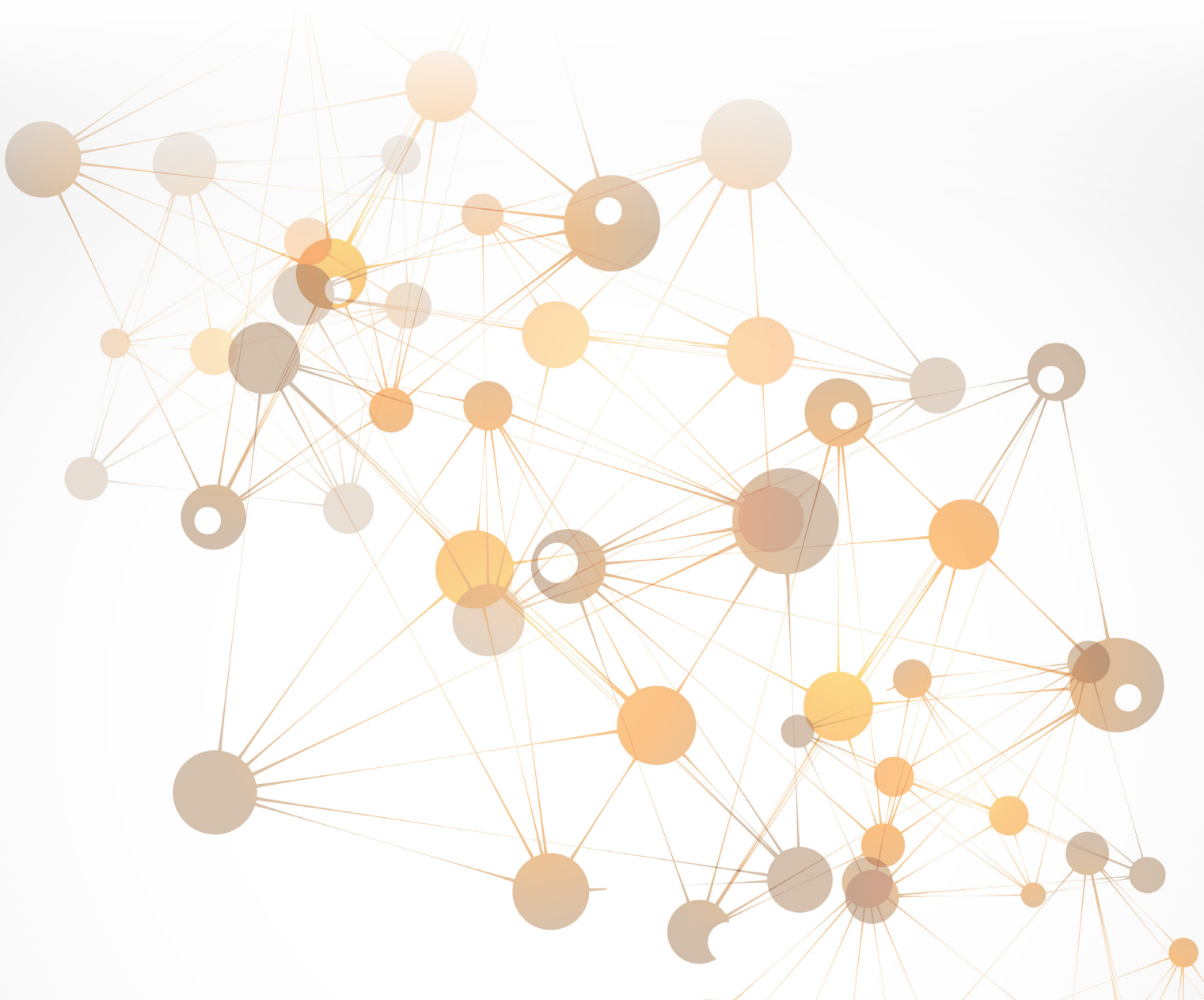
Modeling interactions among users using location, education, industry, employer, and position data, we were able to detect which friends among a list of friends could be used to predict the location of the user in question.



## CREATE A DYADIC MODEL

The geo-inferencing model requires an underlying dyadic dataset to summarize the ties of a closed network. The dyadic dataset allows modeling interdependence among relationship pairs, where one actor influences another actor. In this way, we can model user similarity, and therefore infer the user location.

Since friendship is intrinsically dyadic and depends on both person for formation and maintenance, we form a dyadic dataset as the basis for a model that says if we know where one person in a dyad lives, we can estimate with great precision and accuracy the location of the second member of the dyad.



## MODEL INPUTS (JUST A FEW OF THEM)

**SHARED METRO AREA:** This is a binary variable and the 'outcome of measure'. A user that lives in the same city as his largest group of friends scores a 1. If the user lives in any other city, he or she scores a 0.

**CLUSTER SIZE:** The size of the user's largest group of friends that reside in one metropolitan area. Eg., A user that has one friend in Milwaukee and three friends in New York, scores a 3 because the largest single-city group of friends is 3. The larger this value, the denser the cluster of friends, and the more likely a user is lives near them.

**MEAN DISTANCE:** The average distances among all the user's friends. If all the friends live in the same city, this distance is 0. If all the friends live 1000km apart, this distance would be 1000km. Smaller values indicate tight clusters of friends.

**STANDARD DEVIATION OF DISTANCE:** The square root of the variance of the distance between all the user's friends. A large standard deviation indicates multiple 'clusters' of friends in different areas. A small standard deviation, renders mean distance more predictive.





## RESULTS

This model can predict whether or not a USER lives in the same metro/  
micro area as their largest cluster of FRIENDS with accuracy between  
84-92%:

- ~ 90% accuracy predicting TRUE POSITIVES (prediction = T, actual = T)
- ~ 70% accuracy predicting TRUE NEGATIVES (prediction = F, actual = F)



CONTACT@SERIALMETRICS.COM

408.506.3000



S E R I A L  
M E T R I C S