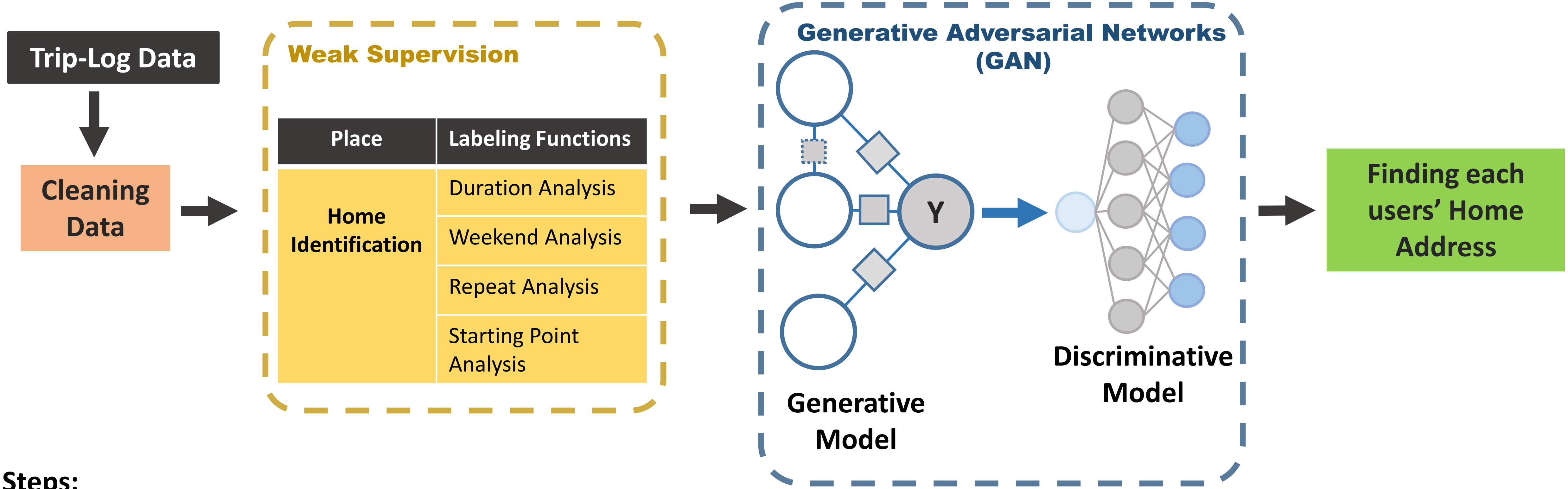
Weak Supervision and Generative Adversarial Networks to Identify Home

Armin Moridi

U.S. Patent Application Serial No. 16/720566;
Filed December 19, 2019
Reference No. H1193586US01

Home Identification Pipeline



Trip-Log Data

Cleaning Data

Weak Supervision

Place	Labeling Functions
Home Identification	Duration Analysis
	Weekend Analysis
	Repeat Analysis
	Starting Point Analysis

Generative Adversarial Networks (GAN)

Generative Model

Discriminative Model

Finding each users' Home Address

Steps:

Cleaning

- Noisy data & outliers
- Tuning GPS accuracy **(DBSCAN)**
- Modification for each analysis

Labeling Functions

- Study user's behavior
- Examine different LFs
- Prepare Weak Learners

GAN

- Tune (Hyper)parameters
- Generative Model
- Discriminative Model

Final Results

Adding 1 column to Trip-Data:
Each trip destination*

Cleaning Data

Noisy Data and Outliers

Trip Log has various information about car and driver behavior. We need to remove unrelated data and add more features by aggregating other features.

27 features dropped. Either they were unrelated or aggregated to make 4 more complicated features.

Different noises and outliers detected and removed from the data

The goal was preserving as much trip as possible:
Total number of Records: 386,939,957
Total Drivers: 624,365

Tuning GPS Accuracy[*]

GPS accuracy will decrease near buildings and we may not receive the same GPS for the exact location

People will not park at the same place when they arrive at their destination.
E.g. No designated parking at home, first-come first-serve parking area at work

Patterns can be extracted from places that visited several times and there is no info in places that visited just once in a while.
E.g. routine to grab coffee from café

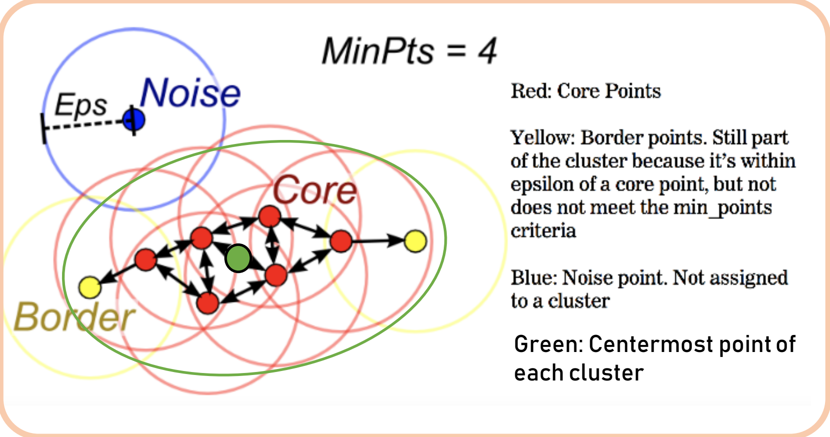
Using **DBSCAN algorithm** to reduce spatial data set size and aggregate coordinates that correspond to same location

Data for each LF is more robust and centermost point of each cluster results in increasing accuracy.
Besides, noise can be identified easier and GPS coordinate error will be minimized

Modification for each Analysis

Each LFs need their own features. E.g. Duration Analysis aggregates each location dwell time however repeat analysis just counts each user's destination repeats

DBSCAN Algorithm



Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

Green: Centermost point of each cluster

[*] Boeing, G. (2018, March 22). Clustering to Reduce Spatial Data Set Size. https://doi.org/10.31235/osf.io/nzhdc

Weak Supervision

In the **weak supervision setting**, our objective is the same as in the supervised setting, however instead of a ground-truth labeled training set we have:
Unlabeled data $X_u= x_1,\ldots, x_N$;
One or more weak supervision sources $p_i(y|x)$, i = 1 : M provided by a human subject matter expert (SME), such that each one has:

- A *coverage set* C_i , which is the set of points xx over which it is defined
- An accuracy, defined as the expected probability of the true label y^* over its coverage set, which we assume is < 1.0

In general, we are motivated by the setting where these weak label distributions serve as a way for human supervision to be provided more cheaply and efficiently: either by providing **higher-level, less precise** supervision (e.g. heuristic rules, expected label distributions), **cheaper, lower-quality** supervision (e.g. crowdsourcing), or taking opportunistic advantage of **existing resources** (e.g. knowledge bases, pre-trained models). These weak label distributions could thus take many well-explored forms. We will use **Weak Labels** as defined as below:

> The weak label distributions could be deterministic functions–in other words, we might just have a set of noisy labels for each data point in C_i. In our case, these noisy labels are generated by having different labeling functions(LFs) that based on the info given determine a coordinate is ‘Home’ or ‘Not Home’. Later, by using a GAN model we aggregate all results generated from LFs to make a strong learner with high accuracy.

Labeling Functions

In Supervised Learning studies, there is no magic analysis that can include all the possibilities. Therefore, We will use different analysis (LFs) to make the model as general as possible and increase its accuracy. E.g. Duration Analysis might not work for workaholic people or in Starting Point Analysis we understood ~90% of American people have morning shifts. So, this LF will not work for the other 10%. The purpose of this study is to define some weak learners (LFs) and by aggregating their results using ML/DL models, we can build a strong learner that is robust and can extract various people's driving patterns with high accuracy.

Duration Analysis

Dwell-time at each center point (resulted from DBSCAN)

Our studies showed that People spend most of their times at home.

Weekend Analysis

Same as Duration Analysis, but we just study locations' dwell-time in weekends

Repeat Analysis

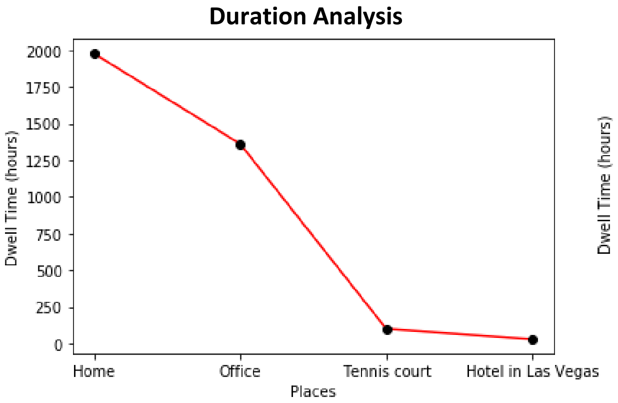
Total Number of trips starting from each center point (resulted from DBSCAN)

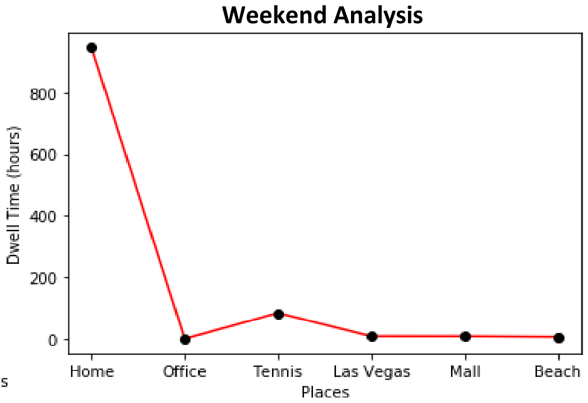
Home is the origin of most of our trips generally and if we monitor this for a one month period, the location with the highest repeats is more likely to be home.

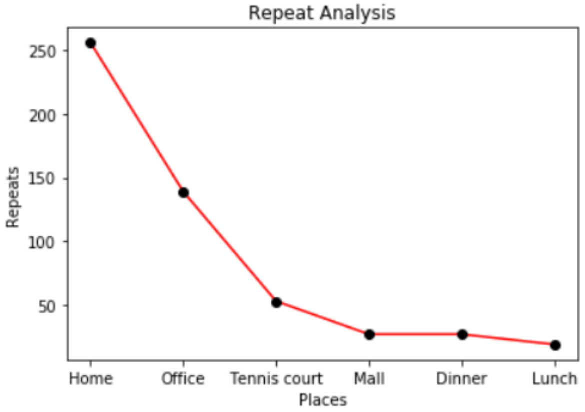
Starting Point Analysis

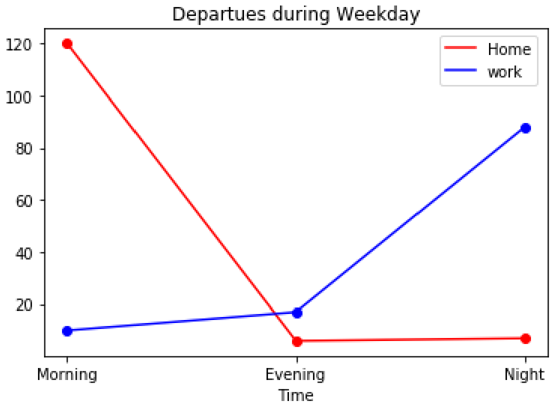
Starting point location in the morning

Most people start their day from home. So mode of the origin location in the morning in one month period can result in home









Generative Adversarial Network

Generative Model

Models the distribution of individual classes

We apply the LFs over unlabeled data and learns a generative model to combine the LFs’ outputs into **probabilistic labels**

We learns the accuracies of weak supervision sources **without access to ground truth** using a generative model. Furthermore, it also learns **correlations** and **other statistical dependencies** among sources, **correcting for dependencies** in labeling functions that skew the estimated accuracies

This step uses no ground-truth data, learning instead from the agreements and disagreements of the labeling functions

Discriminative Model

Learns the boundary between classes

The output of Generative model is a set of probabilistic labels that can be used to train a wide variety of state-of-the-art machine learning models, such as popular deep learning models.

While the generative model is essentially a re-weighted combination of the user-provided labeling functions—which tend to be precise but low-coverage—modern discriminative models can retain this precision while learning to generalize beyond the labeling functions, increasing coverage and robustness on unseen data.

We use these labels to train a discriminative classification model, such as a deep neural network.

Why we need Discriminative Model?*

There are three main advantages in using the predicted labels of the *generative model* as training labels for discriminative model:

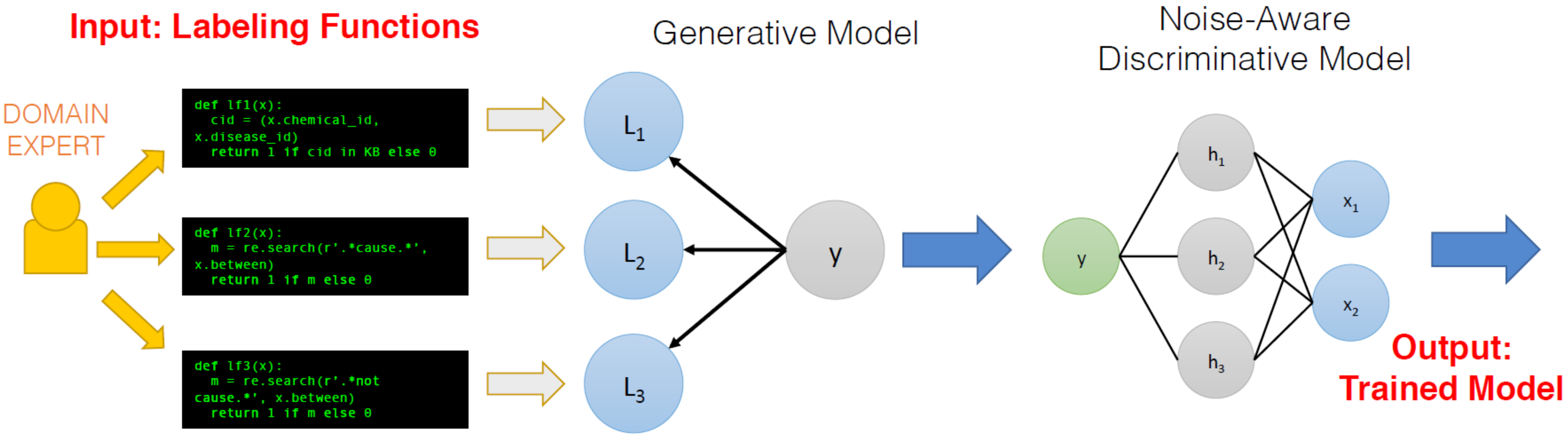
Generalization: Consider for example if you use 20 labeling functions to label some of your data. Those two rules may only label 60% of your data; on the other 40%, you have no votes. But those 40% of examples will likely have lots of features in common with examples in the 60% that you *do* have some noisy labels for. So you label whatever portion you can, and use those to learn weights over a much larger, richer feature set

Scaling with unlabeled data: The end discriminative model will improve significantly with more unlabeled data, letting you take advantage of an often abundant resource.

Cross-Modal Settings: You can train a discriminative model over a different or even entirely disjoint set of features than the generative model.

* In Home Identification example Generative model will build two classes of 'HOME' and 'NOT HOME' and distribute coordinates to these two classes by aggregating Labeling Functions result. After we have two labeled classes, discriminative model will learn the boundary between two classes by using deep Neural Nets.

Weak Supervision* Pipeline in Snorkel



```
def lf1(x):
  cid = (x.chemical_id,
x.disease_id)
  return 1 if cid in KB else 0
```

```
def lf2(x):
  m = re.search(r'.*cause.*',
x.between)
  return 1 if m else 0
```

```
def lf3(x):
  m = re.search(r'.*not
cause.*', x.between)
  return 1 if m else 0
```

Users write functions to label training data

We model functions' behavior to denoise it

We use estimated labels to train a model

*In weakly supervised learning, you use a **limited amount of labeled data** that is easy to get and/or makes a real difference and then learn the rest.