CSE 162 Mobile Computing

Lecture 15: Location Programming and Processing

Hua Huang

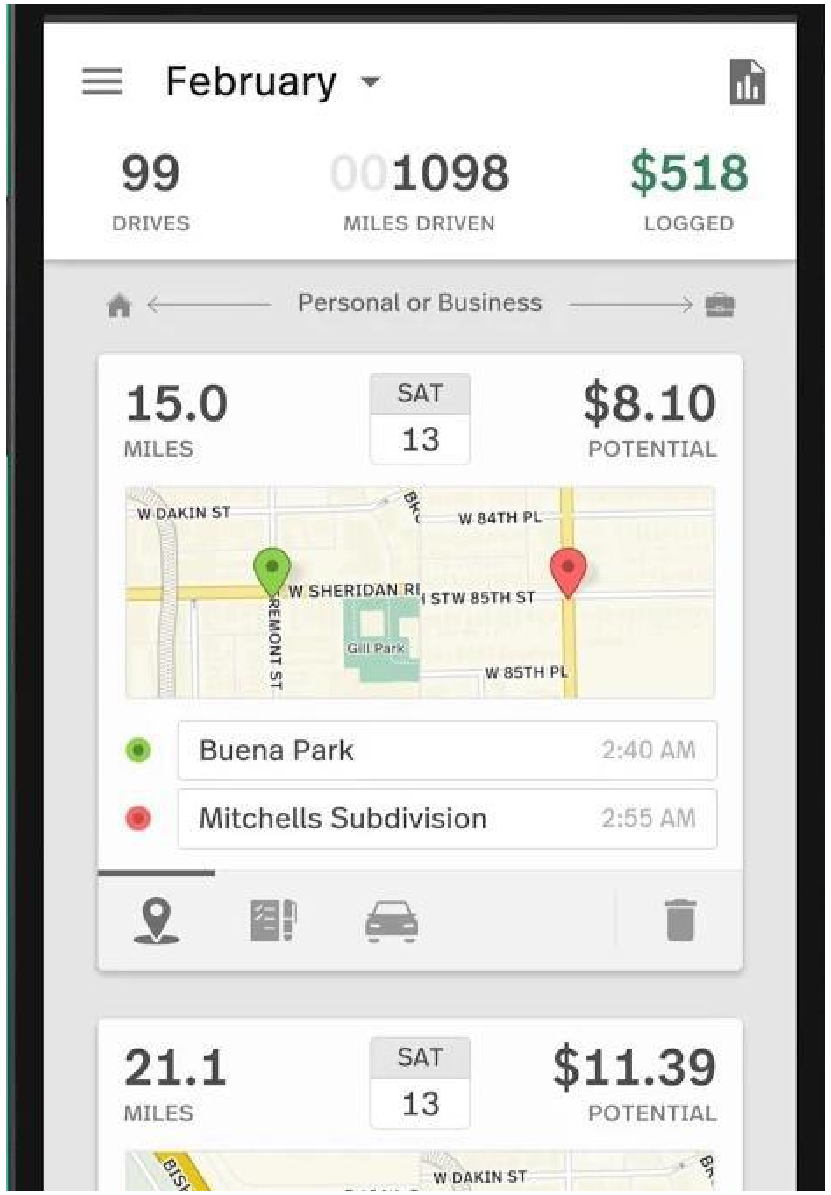
Department of Computer Science and Engineering

University of California, Merced

Some Interesting Location-Aware Apps

MileIQ

- **The Problem:** Mileage tracking is useful but a burden.
    - IRS deductions on taxes
    - Some companies reimburse employees for mileage,
- Passively, automatically tracks business mileage, IRS compliant
- Swipe right after drive to indicate it was a business trip


February

99 DRIVES | 001098 MILES DRIVEN | $518 LOGGED

Personal or Business

15.0 MILES | SAT 13 | $8.10 POTENTIAL

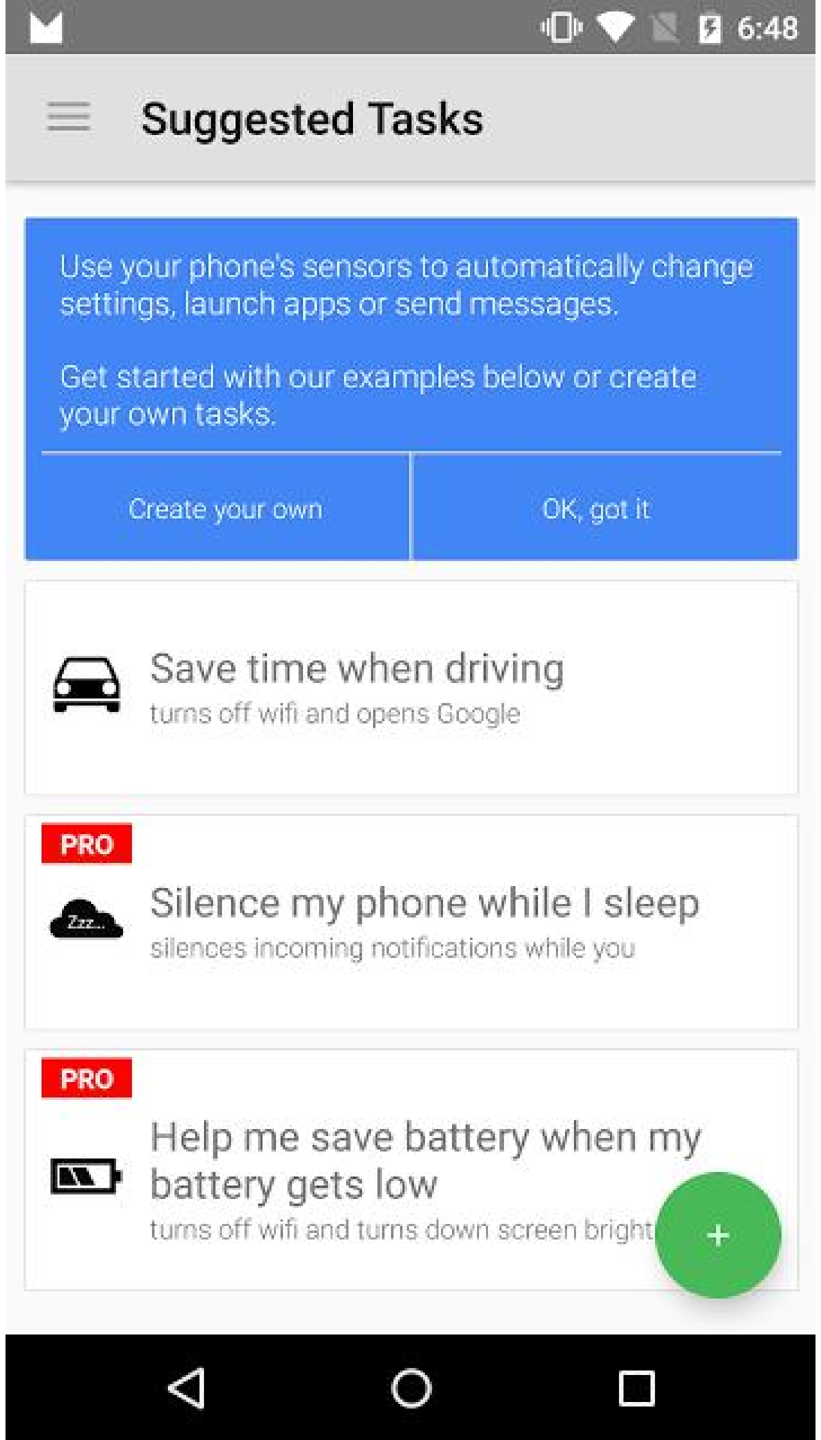
Buena Park 2:40 AM

Mitchells Subdivision 2:55 AM

21.1 MILES | SAT 13 | $11.39 POTENTIAL

Trigger

- Use geofences, NFC, bluetooth, WiFi connections, etc to set auto-behaviors
  - Battery low -> turn off bluetooth + auto sync
  - Silence phone every morning when you get to work
  - Turn off mobile data when you connect to your home WiFi
  - Silence phone and set alarm once I get into bed
  - Use geofence for automatic foursquare checkin
  - Launch maps when you connect to your car's bluetooth network


6:48

Suggested Tasks

Use your phone's sensors to automatically change settings, launch apps or send messages.

Get started with our examples below or create your own tasks.

Save time when driving
turns off wifi and opens Google

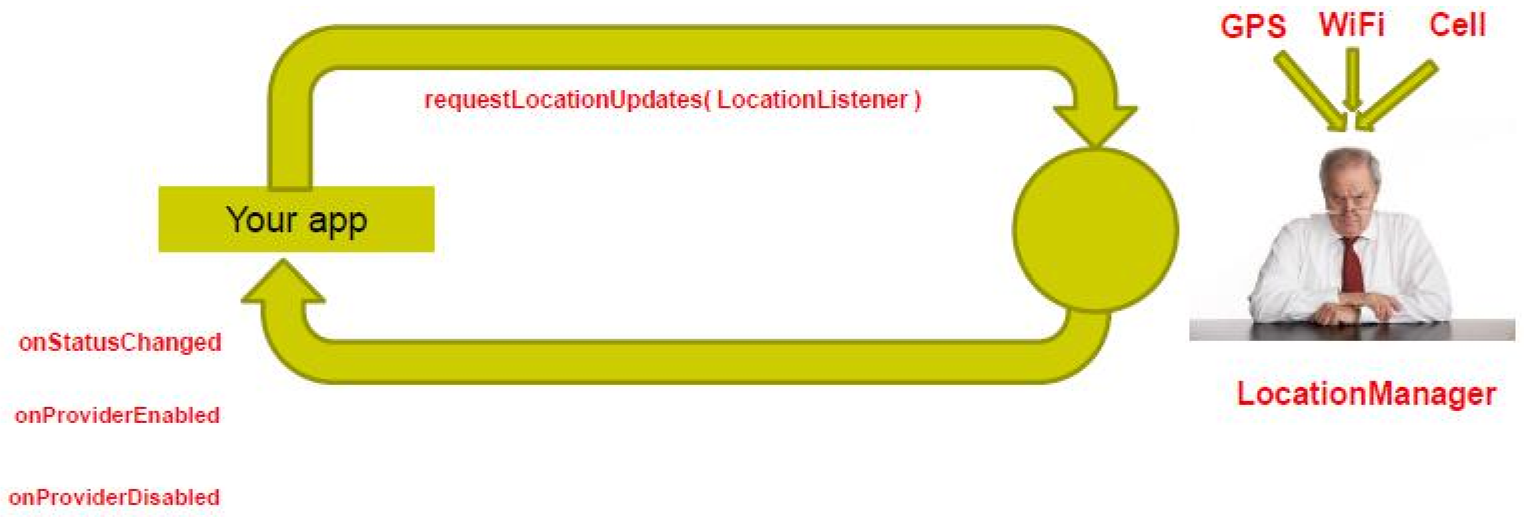
PRO
Silence my phone while I sleep
silences incoming notifications while you

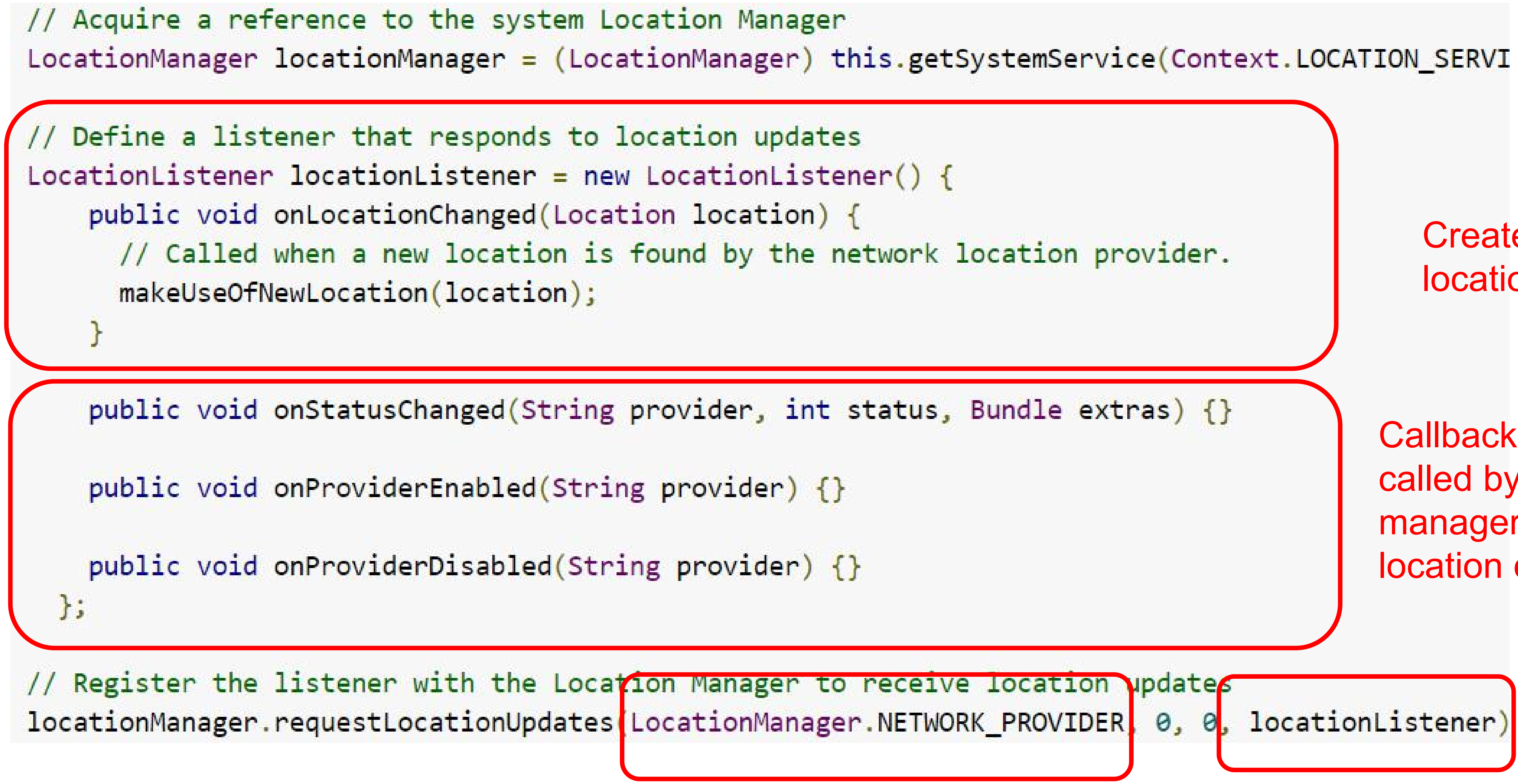
PRO
Help me save battery when my battery gets low
turns off wifi and turns down screen bright

Location Sensing in Android Apps

The Basic Location APIs

- **LocationManager:**
  - Android module receives location updates from GPS, WiFi, etc
  - App registers/requests location updates from LocationManager





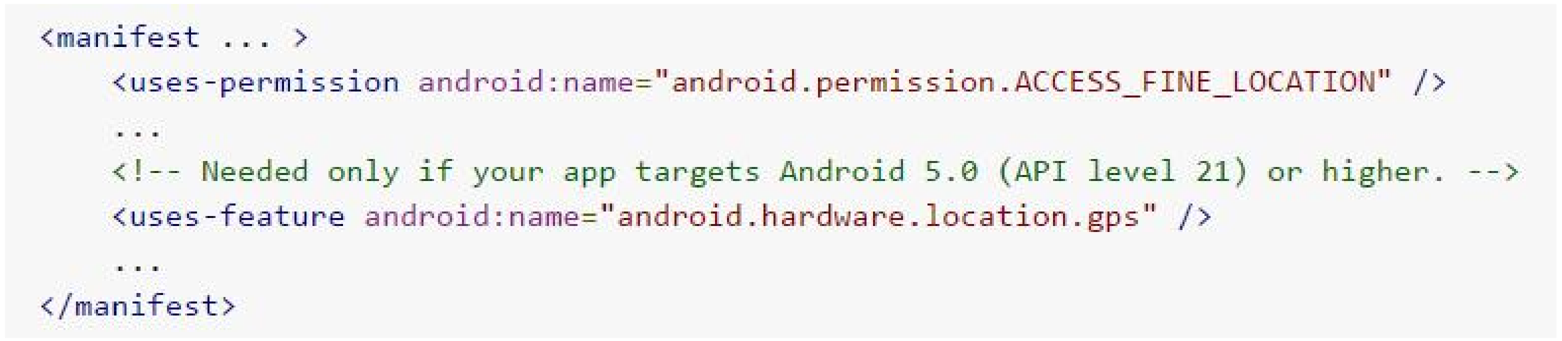
```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVI

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
      // Called when a new location is found by the network location provider.
      makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
  };

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener)
```

Create listener for location info

Callback methods called by Location manager (e.g. when location changes))

Requesting User Permissions

- Need smartphone owner’s permission to use their GPS



```
<manifest ... >
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    ...
    <!-- Needed only if your app targets Android 5.0 (API level 21) or higher. -->
    <uses-feature android:name="android.hardware.location.gps" />
    ...
</manifest>
```

- **ACCESS_FINE_LOCATION:** GPS
- **ACCESS_COARSE_LOCATION:** WiFi or cell towers

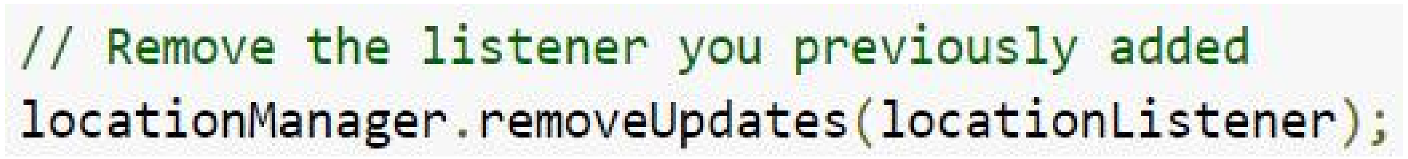
Getting Cached Copy of Location (Fast)

- Getting current location may take a while
- Can choose to use location cached (possibly stale) from Location Manager

```java
String locationProvider = LocationManager.NETWORK_PROVIDER;
// Or use LocationManager.GPS_PROVIDER

Location lastKnownLocation = locationManager.getLastKnownLocation(locationProvider);
```

Stopping Listening for Location Updates

- Location updates consume battery power
- Stop listening for location updates whenever you no longer need



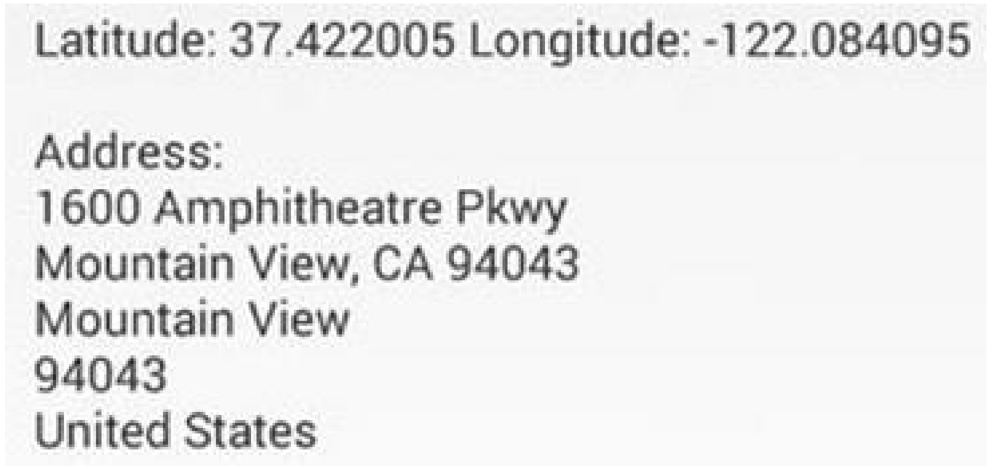
```
// Remove the listener you previously added
locationManager.removeUpdates(locationListener);
```

Location Representation in Android

Semantic Location

- GPS represents location as <longitude,latitude>
- **Semantic location** is better for reasoning about locations
- **E.g.** Street address (140 Park Avenue, Worcester, MA) or (building, floor, room)
- **Android supports:**
  - **Geocoding:** Convert addresses into longitude/latitude coordinates
  - **Reverse geocoding:** convert longitude/latitude coordinates into human readable address

- **Android Geocoding API:** access to **geocoding** and **reverse geocoding**services using HTTP requests

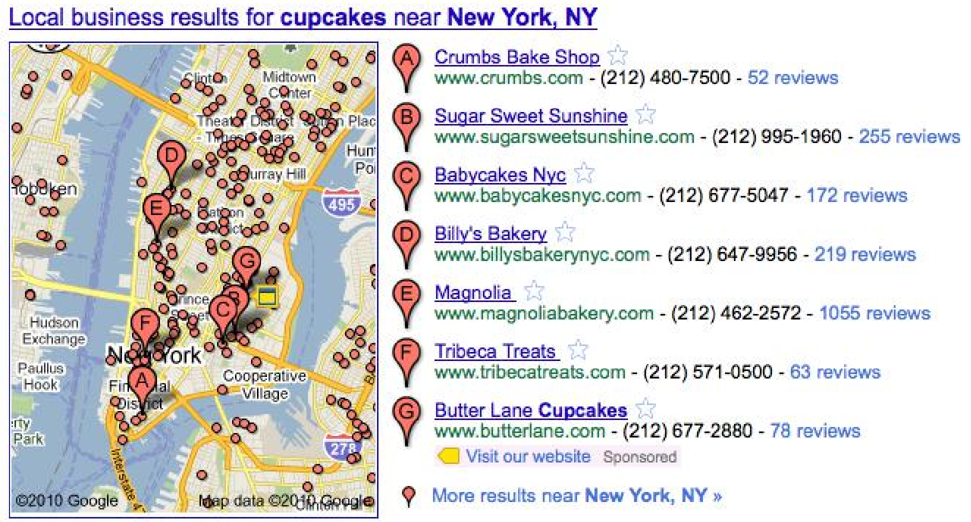


Latitude: 37.422005 Longitude: -122.084095

Address:
1600 Amphitheatre Pkwy
Mountain View, CA 94043
Mountain View
94043
United States

Google Places API Overview

- Access **high-quality photos** of a place
- Users can also add place information to the database
  - E.g. business owners can add their business as a place in Places database
  - Other apps can then retrieve info after moderation
- **On-device caching:** Can cache places data locally on device to avoid roundtrip delays on future requests


Local business results for **cupcakes** near **New York, NY**

A Crumbs Bake Shop
www.crumbs.com - (212) 480-7500 - 52 reviews

B Sugar Sweet Sunshine
www.sugarsweetsunshine.com - (212) 995-1960 - 255 reviews

C Babycakes Nyc
www.babycakesnyc.com - (212) 677-5047 - 172 reviews

D Billy's Bakery
www.billysbakerynyc.com - (212) 647-9956 - 219 reviews

E Magnolia
www.magnoliabakery.com - (212) 462-2572 - 1055 reviews

F Tribeca Treats
www.tribecatreats.com - (212) 571-0500 - 63 reviews

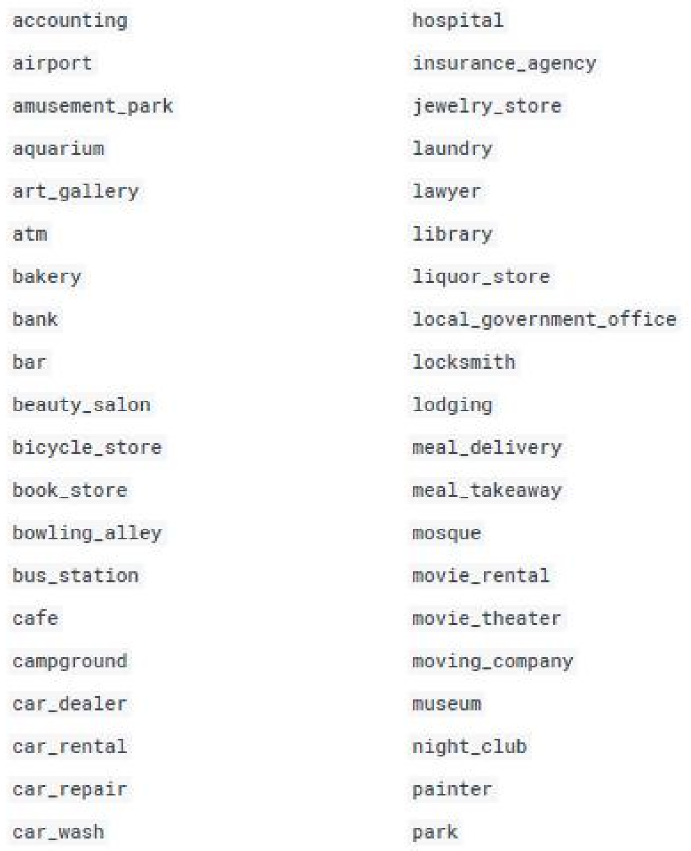
G Butter Lane **Cupcakes**
www.butterlane.com - (212) 677-2880 - 78 reviews
Visit our website Sponsored

More results near **New York, NY** »

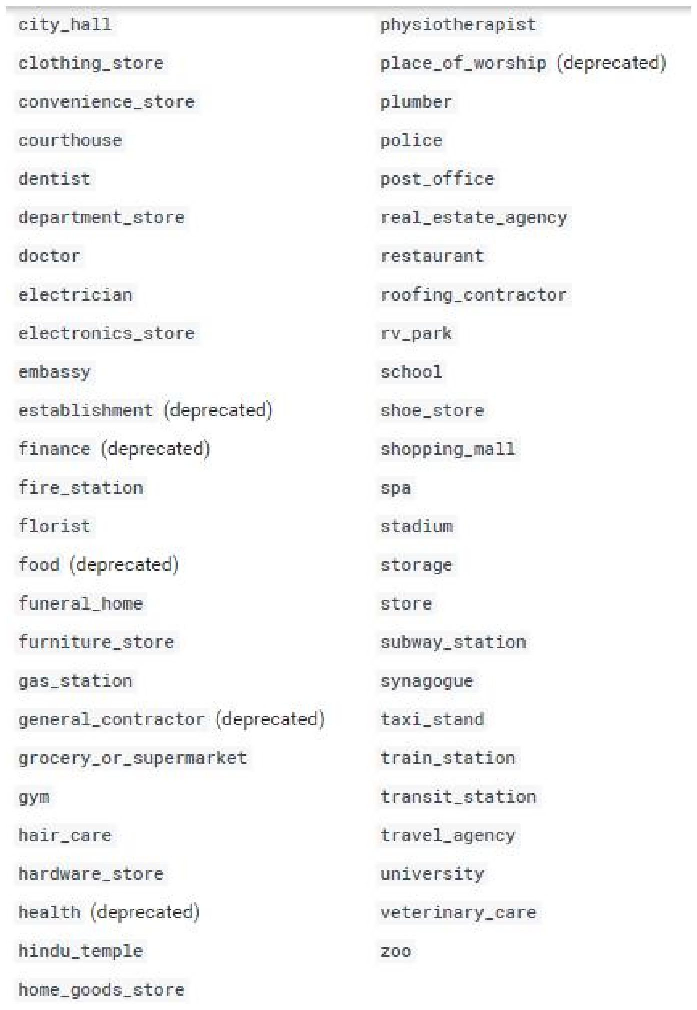
Google Places

- **Place:** physical space that has a name (e.g. local businesses, points of interest, geographic locations)
  - E.g Logan airport, place type is **airport**
- **API:** Provides Contextual information about places near device.
  - **E.g:** name of place, address, geographical location, place ID, phone number, place type, website URL, etc.
- Compliments geographic-based services offered by Android location services

Sample Place Types



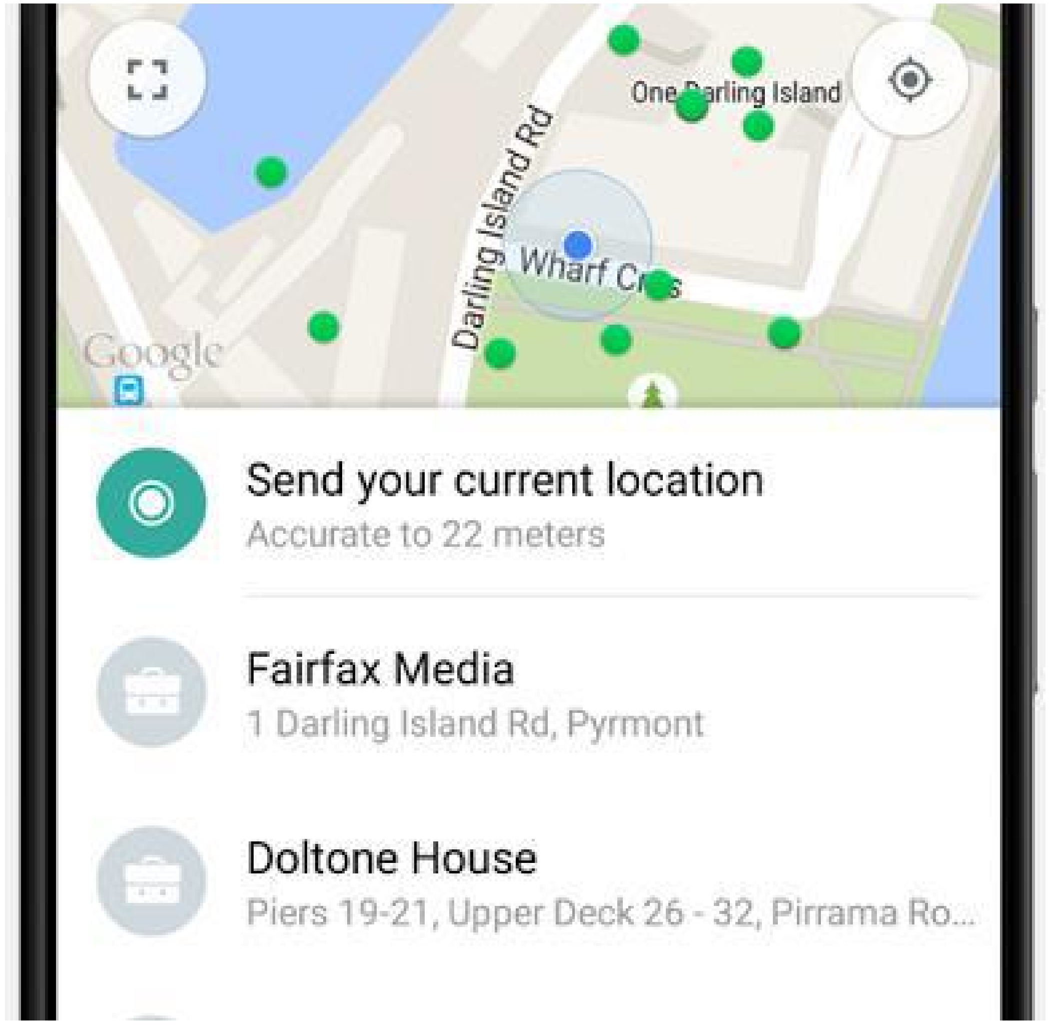
accounting	hospital
airport	insurance_agency
amusement_park	jewelry_store
aquarium	laundry
art_gallery	lawyer
atm	library
bakery	liquor_store
bank	local_government_office
bar	locksmith
beauty_salon	lodging
bicycle_store	meal_delivery
book_store	meal_takeaway
bowling_alley	mosque
bus_station	movie_rental
cafe	movie_theater
campground	moving_company
car_dealer	museum
car_rental	night_club
car_repair	painter
car_wash	park



city_hall	physiotherapist
clothing_store	place_of_worship (deprecated)
convenience_store	plumber
courthouse	police
dentist	post_office
department_store	real_estate_agency
doctor	restaurant
electrician	roofing_contractor
electronics_store	rv_park
embassy	school
establishment (deprecated)	shoe_store
finance (deprecated)	shopping_mall
fire_station	spa
florist	stadium
food (deprecated)	storage
funeral_home	store
furniture_store	subway_station
gas_station	synagogue
general_contractor (deprecated)	taxi_stand
grocery_or_supermarket	train_station
gym	transit_station
hair_care	travel_agency
hardware_store	university
health (deprecated)	veterinary_care
hindu_temple	zoo
home_goods_store	

Google Places API Overview

- **Use Place picker UI:** allows users select place from “possible place” on a map
- **Get current place:** place where device is last known to be located
  - Returns **list** of likely places + likelihood device is in that place



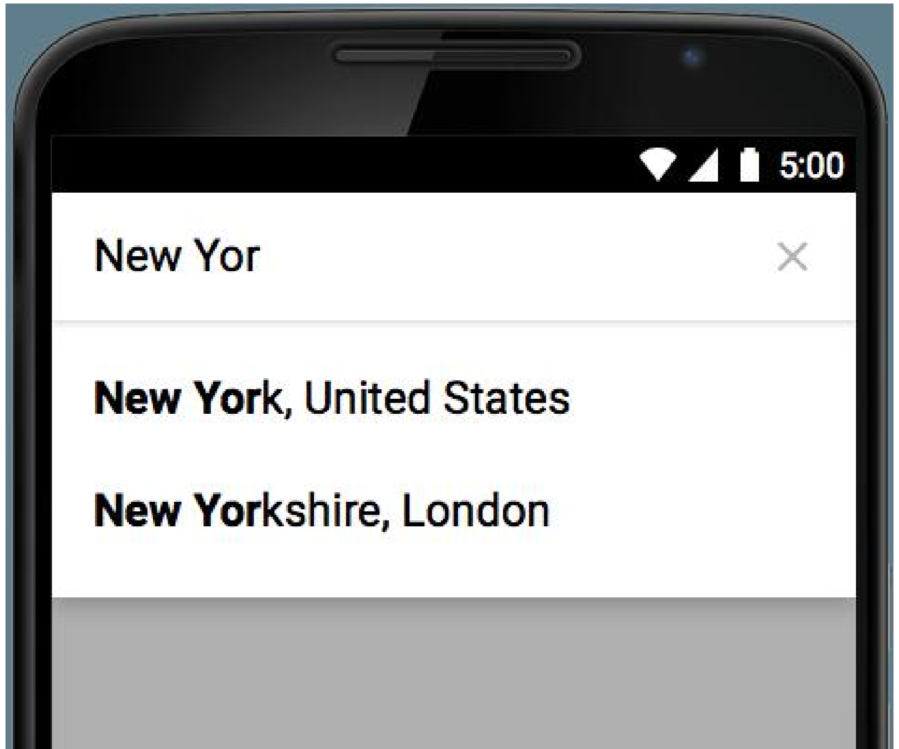
Send your current location
Accurate to 22 meters

Fairfax Media
1 Darling Island Rd, Pyrmont

Doltone House
Piers 19-21, Upper Deck 26 - 32, Pirrama Ro...

Google Places API Overview

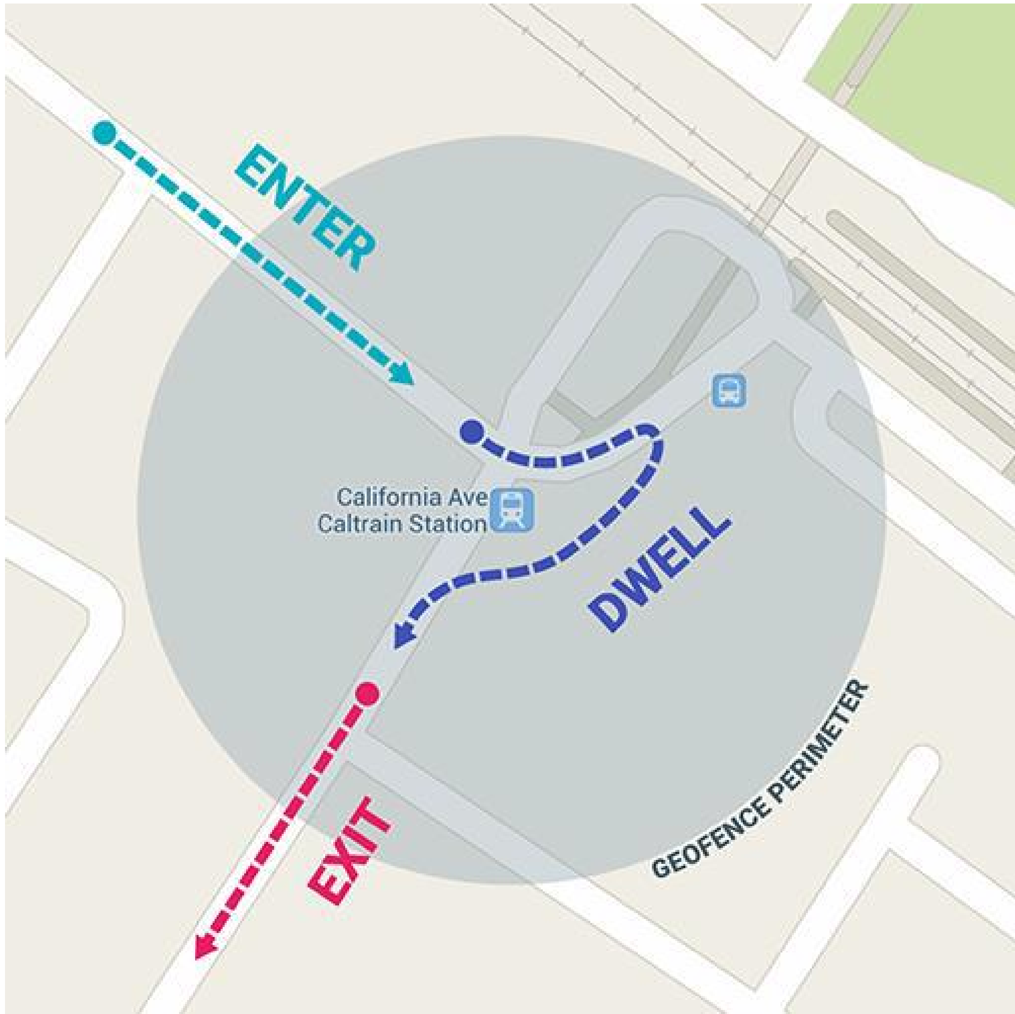
- **Autocomplete:** queries the location database as users type, suggests nearby places matching letters typed in



Other Useful Google Maps/Location APIs

GeoFencing

- **Geofence:** Sends alerts when user is within a certain radius to a location of interest
- Can be configured to send:
  - **ENTER** event when user enters circle
  - **EXIT** event when user exits circle
- Can also specify a duration or **DWELL** user must be in circle before triggering event



Other Maps/Useful Location APIs

- **Maps Directions API:** calculates directions between locations (walking, driving) as well as public transport directions
- **Distance Matrix API:** Calculate travel time and distance for multiple destinations
- **Elevation API:** Query locations on earth for elevation information, calculate elevation changes along routes

5000
4000
3000
2000
1000
0
Elevation in Meters

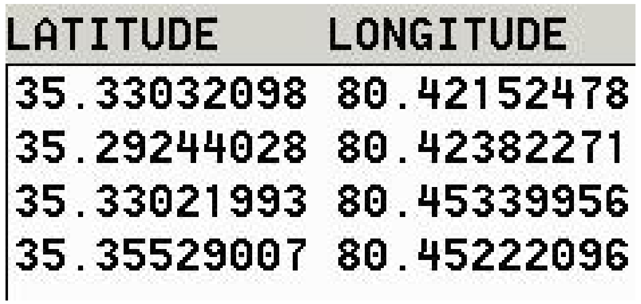
Other Useful Maps/Location APIs

- **Roads API:**
  - snaps set of GPS coordinates to road user was likely travelling on (best fit)
  - Returns posted speed limits for any road segment (premium plan)
- **Time Zone API:** request time zone for location on earth

GPS Clustering & Analytics

Determining Points of Interest from GPS Location Sequences

- **Points of Interest:** Places where a person spends lots of time (e.g. home, work, café, etc)
- **Given a sequence GPS <longitude, latitude>** points, how to infer points of interest
- **General steps:**
  - **Pre-process sequence of GPS points** (remove outliers, etc)
  - **Cluster points**
  - **Convert to semantic location**



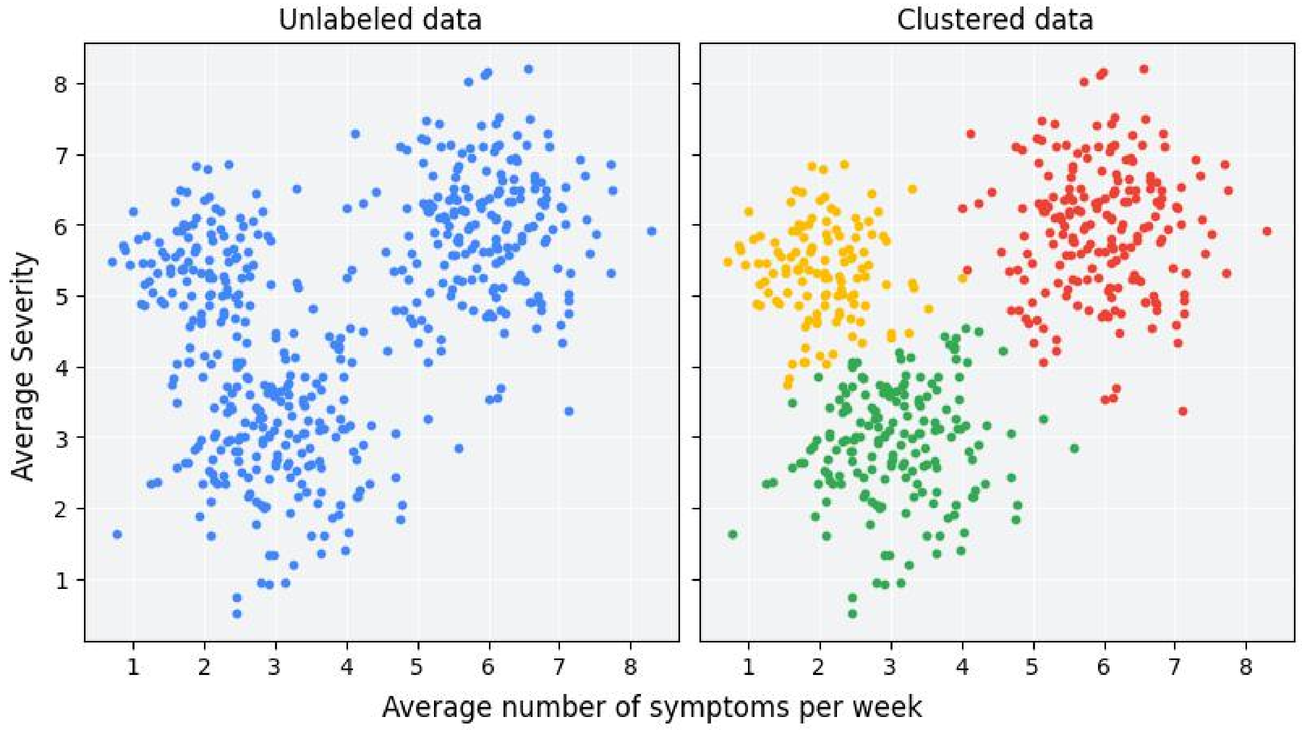
LATITUDE	LONGITUDE
35.33032098	80.42152478
35.29244028	80.42382271
35.33021993	80.45339956
35.35529007	80.45222096

Step 1: Pre-Processing GPS Points (Remove Noise and Outliers)

- **Remove low density points (few neighbors):**
  - i.e. places where little time was spent
  - E.g. radius of 20 meters, keep only clusters with at least 50 points
  - If GPS coordinates retrieved every minute, only considering places where you spent at least 50 minutes
- **Remove points with movement:**
  - GPS returns speed as well as <longitude, latitude> coordinates
  - If speed user is moving, discard that GPS point
- **Reduce data for stationary locations:**
  - When user is stationary at same location for long time, too many points generated (e.g. sitting at at chair)
  - Remove some points to speed up processing

Step 2: Cluster GPS Points

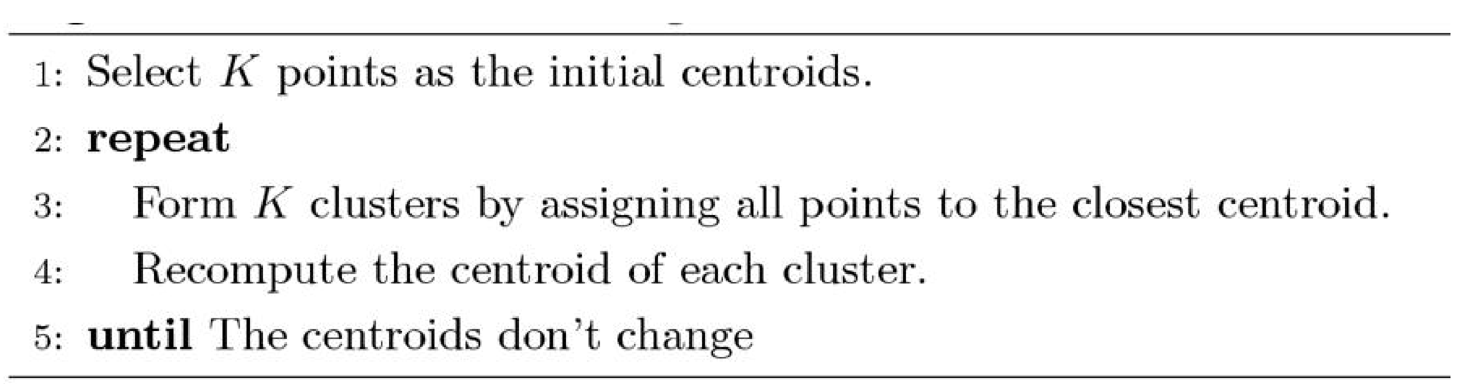
- **Cluster Analysis:** Group points



- Two main clustering approaches
  - K-means clustering
  - DBSCAN

K-Means Clustering

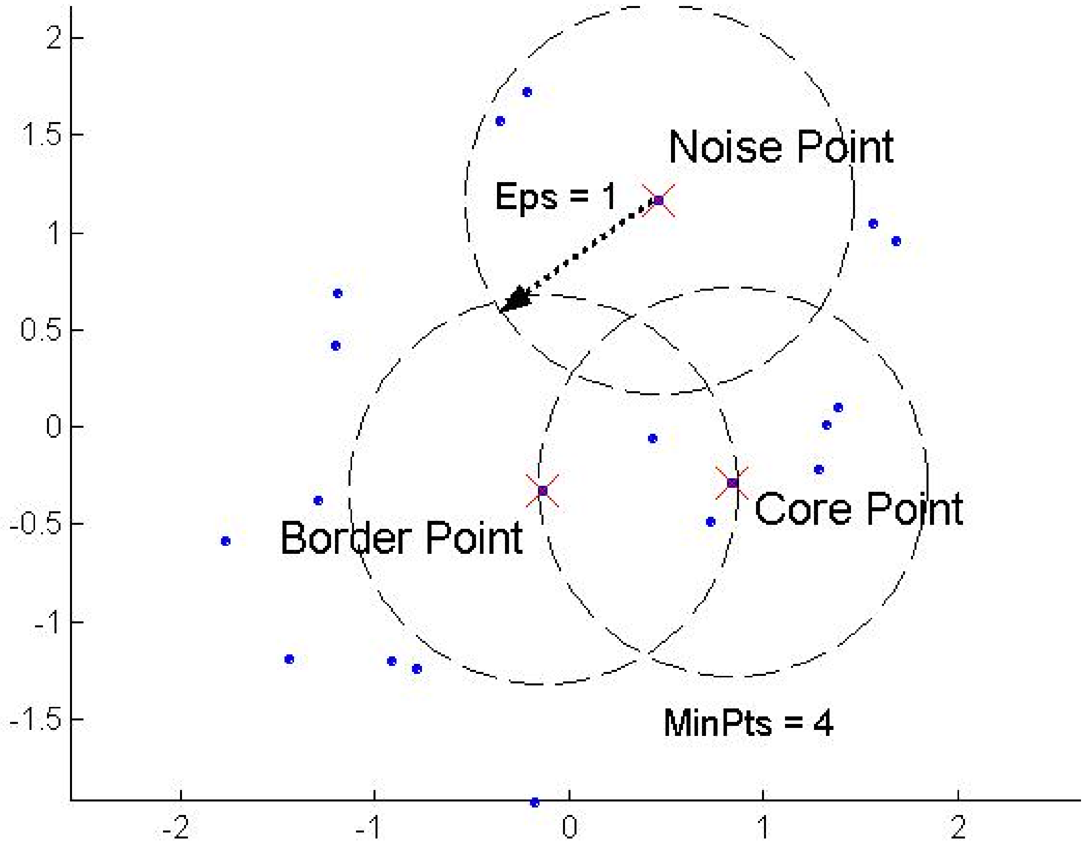
- Each cluster has a center point (centroid)
- Each point associated to cluster with closest centroid
- Number of clusters, *K*, must be specified



```
1: Select $K$ points as the initial centroids.
2: **repeat**
3:     Form $K$ clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: **until** The centroids don't change
```

DBSCAN Clustering

- Density-based clustering
- **Density:** Number of points within specified radius (Eps)
- **Core points:** has > minPoints density
- **Border point:** has < minPoints density but within neighborhood of core point
- **Noise point:** not core point or border point



DBSCAN Algorithm

- Eliminate noise points
- **Cluster remaining points**

```
$current\_cluster\_label \leftarrow 1$
**for** all core points **do**
  **if** the core point has no cluster label **then**
    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
    Label the current core point with cluster label $current\_cluster\_label$
  **end if**
  **for** all points in the $Eps$-neighborhood, except $i^{th}$ the point itself **do**
    **if** the point does not have a cluster label **then**
      Label the point with cluster label $current\_cluster\_label$
    **end if**
  **end for**
**end for**
```

Converting Clusters to Semantic Locations

- Can simply call reverse geocoding or Google Places on the centroid of the clusters
- Determining work? Cluster where user spends longest time most time (9-5pm)
- Determining home? Cluster where user spends most time 6pm –6am