

CitySense and CabSense

David Rosenberg

New York University

April 6, 2016

The CitySense and CabSense Problems

Sense Networks

- Startup company incorporated around 2006.
- Objective: Develop and leverage expertise in **location data** analytics.
- First product was called CitySense¹ (2008).
 - A real-time, data-driven guide to nightlife in San Francisco.

¹See "CitySense: Multiscale space time clustering of GPS points and trajectories" by Markus Loescher and Tony Jebara (2009).

<http://www.cs.columbia.edu/~jebara/papers/CitySense.JSM2009.pdf>

CitySense (2008)

The screenshot shows the CitySense mobile application running on an iPhone. The top status bar indicates signal strength, battery level, and the time (5:41 PM). The main screen features a map of San Francisco with a heatmap overlay showing nightlife activity levels. A red square highlights a specific location, with a callout bubble providing details: "Grove St & Hyde St" and "This location was previously very busy. Its activity level was relatively normal." Below the map, there's a line graph showing activity levels over time. At the bottom of the screen are standard iOS navigation buttons for back, forward, and search.

Citysense™
Live San Francisco Nightlife Activity

Where is everybody?

- How busy is the city? Know when to go out
- See the top nightlife hotspots in real-time
- Find out what's there in one click
- Find out where everyone's going next

[More info](#)

For real-time nightlife on your iPhone®, visit the App Store

Also available for the BlackBerry®
Go to www.citysense.com on your BlackBerry® to download.

(Sadly, no longer in the App Store.)

CitySense: Use Cases

Two use cases:

- ① I'm new to the city – where does everybody hang out at night?
- ② I know the city, but is there anything **special** going on tonight?

CitySense: Data Source

- Taxi GPS data for sale in San Francisco

The screenshot shows the homepage of the Yellow Cab Cooperative website. At the top, there is a navigation bar with links for HOME, SERVICE, FAQS, ACCOUNTS, ABOUT, CONTACT, and SAN FRANCISCO. Below the navigation bar, the date is listed as TUESDAY, MAY 01, 2012, and there is a TEXT SIZE adjustment button. On the left side, there is a large yellow taxi cab image with the text "GET A CAB NOW" and the phone number "415.333.3333". Below this, there is a "BOOK A CAB ONLINE" button. To the right of this section, there is a large image of a yellow taxi cab driving on a city street with the text "YELLOW MAKES IT EASY." and a descriptive paragraph about the auto-dispatch system. At the bottom of the page, there are two sections: "Our History" and "Our Community".

CitySense

- Main Idea: Taxi destinations are a proxy for where people are going.
- Can use taxi data to bootstrap
 - Once we had users, we could use the locations from their phones.
- Taxi feed is **real-time**, so can use it to find those big secret parties.

CitySense

Data Science Strategy

- ① Model “typical” behavior of each area of the city.
- ② Rank areas with activity levels that are “most unusual”.

We'll discuss modeling strategies shortly.

CabSense (2010): Second Product

Business objective

I'm in NYC, and I need a taxi. Where's the best place to find one?

Data Source

NYC Taxi and Limousine Commission provided GPS data from all taxi cabs. However, **not real-time**.

Main Idea

Historical taxi pickup frequency are predictive of future frequencies. Need to model pickup rates based on historical data.

CabSense (2010): Second Product



(Recently updated with fresh data.)

Picture courtesy of Blake Shaw.

Plan for this lecture

- We're going to focus on the CitySense "anomaly detection" problem.
- But use the NYC taxi pickup data, since we live in NYC.
- Our dataset is from 2009.
- Currently (2016/04/06) you can download 2013 data from
http://chriswhong.com/open-data/foil_nyc_taxi/
- You can also request data directly from the NYC Taxi and Limousine Commission via the Freedom of Information Law.
<http://www.nyc.gov/html/tlc/html/passenger/records.shtml>

The Case for Probability Models

Predicting Probability Distributions

- So far we've solved decision problems with a two types of action spaces:
 - $\mathcal{A} = \{-1, 1\}$ [hard classification, e.g. decision trees as used in AdaBoost]
 - $\mathcal{A} = \mathbf{R}$ [regression or soft classification]
 - $\mathcal{A} = \{1, 2, \dots, k\}$ [hard multiclass classification]
 - $\mathcal{A} = \mathbf{R}^k$ [soft multiclass classification]
- Today we consider a third type of action space:

$$\mathcal{A} = \{\text{Probability distributions on space } \mathcal{Y}\}$$

- Why?

The Joy of Probability Distributions

- Output space $\mathcal{Y} = \mathbb{R}$.
- Machine computes conditional probability density on \mathcal{Y} given $x \in \mathcal{X}$:

$$x \mapsto p(y | x)$$

- If we know $p(y | x)$, we can find a \hat{y} that minimizes any other loss function:
 - For square loss, give the mean of $p(y | x)$. [From homework #1]
 - For ℓ_1 loss, give the median of $p(y | x)$. [From homework #1]
 - etc.
- Gives idea of the possibilities of knowing the whole distribution.

Probability Distributions for CitySense

- Model predicts distribution $p(y | x) = \text{Poisson}(40)$ for number of pickups.
 - in a region R ,
 - between 9pm and 10pm.
- Actual number of taxi pickups = 100
- Is 100 an anomaly?
- $\mathbb{P}(y \geq 100 | x) = \sum_{y=100}^{\infty} p(y | x)$ measures how unusual this event is.

Probability Distributions for Prediction Intervals

- Given a probability distribution,
 - it's straightforward to give **prediction intervals**.
- A 95% prediction interval is an interval $[a, b]$ such that

$$\mathbb{P}(y \in [a, b] \mid x) \approx .95$$

- We can get $[a, b]$ by finding the 2.5% and 97.5% quantiles of $p(y \mid x)$.
- [Alternatively, can do this with **quantile regression**.]

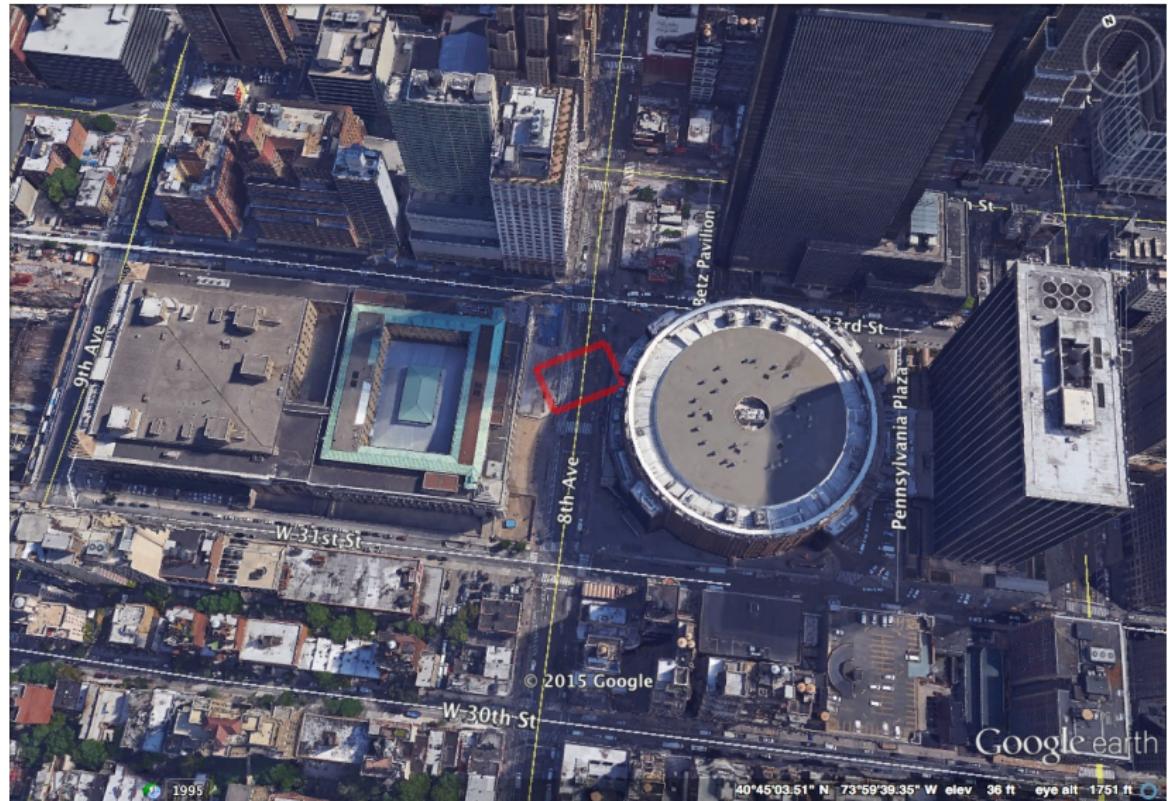
The Grid Cells

The Basic Approach

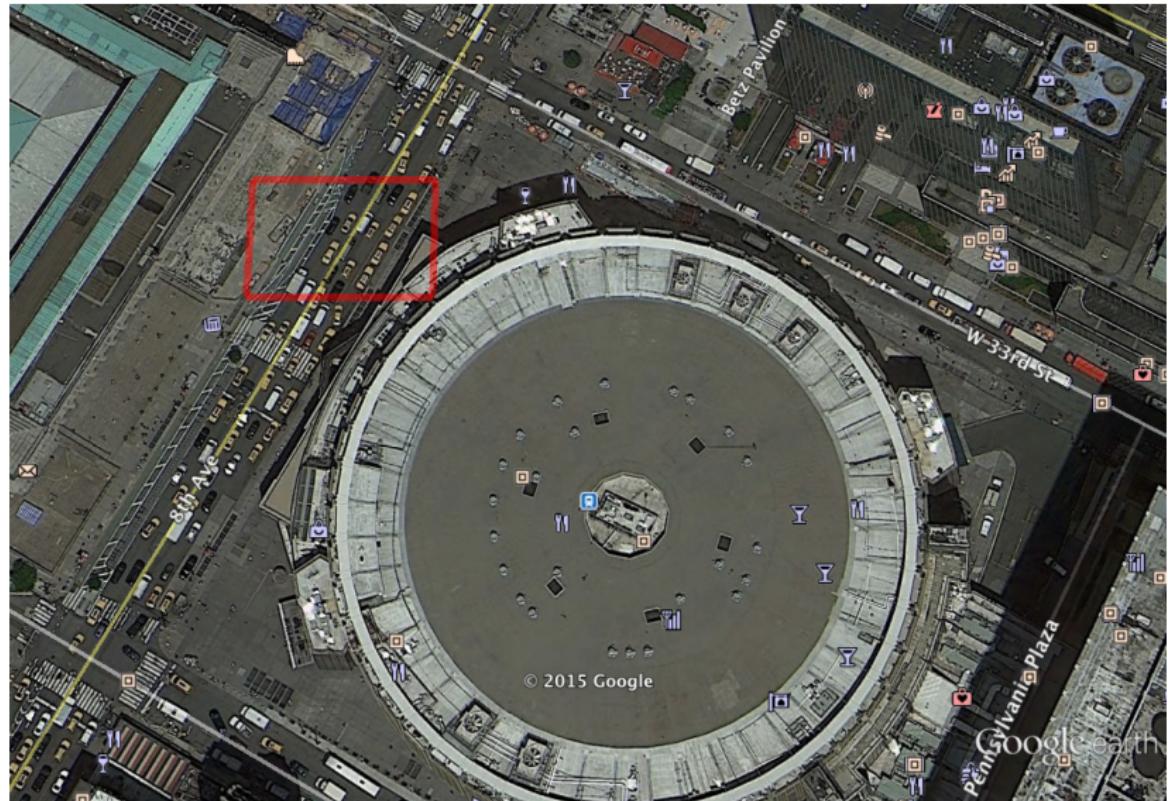
- Raw input is [roughly] continuous in
 - space (lat/lon) and
 - time (seconds since 1970-01-01).
- To make it easier to handle, we partition space and time into buckets.
- Spatial partitioning
 - Divide earth into regularly spaced grid cells.
 - About 400,000 grid cells to cover NYC
- Time partitioning
 - Only consider times at the hour level.
- Aggregate taxi pickup counts at the Grid Cell / Hour level.

Initial data analysis, including aggregation by grid cell and hour, was done by Blake Shaw.

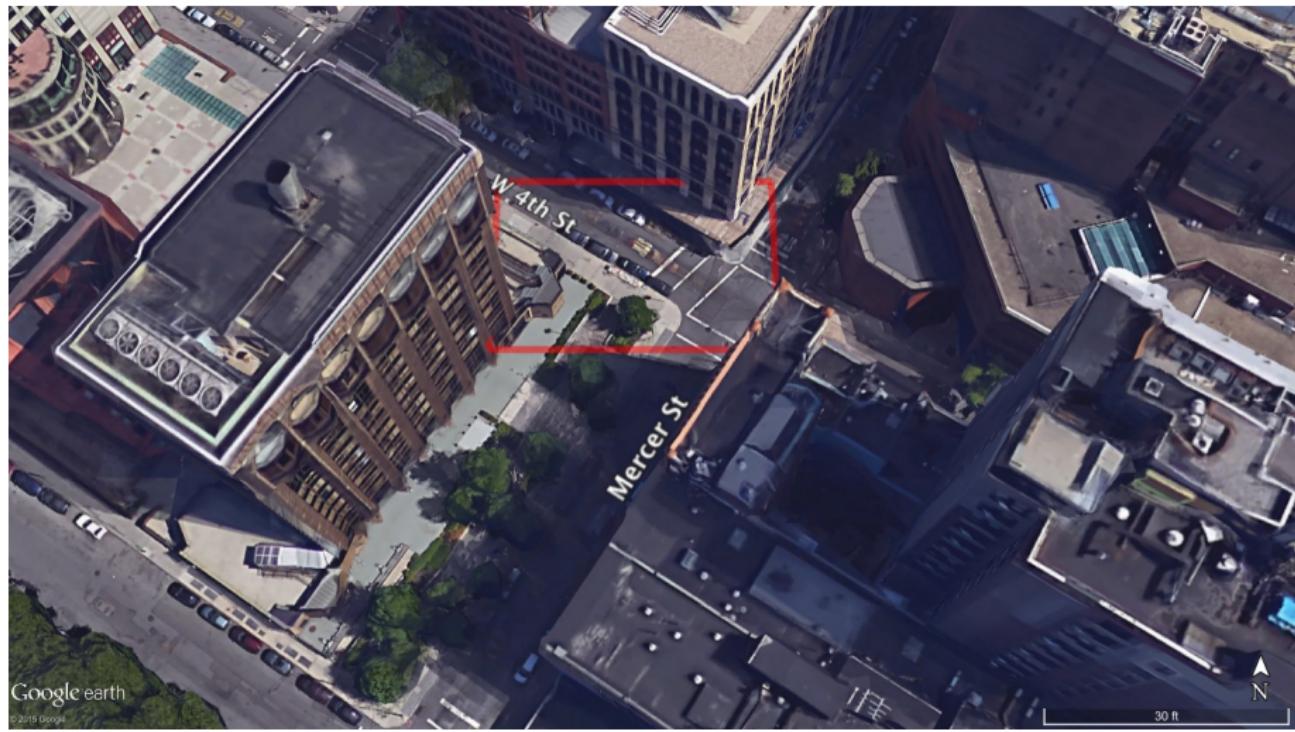
Most Active Grid Cell: Penn Station (Grid ID 7750)



Most Active Grid Cell: Penn Station (Grid ID 7750)



Courant Institute (Grid ID 21272)

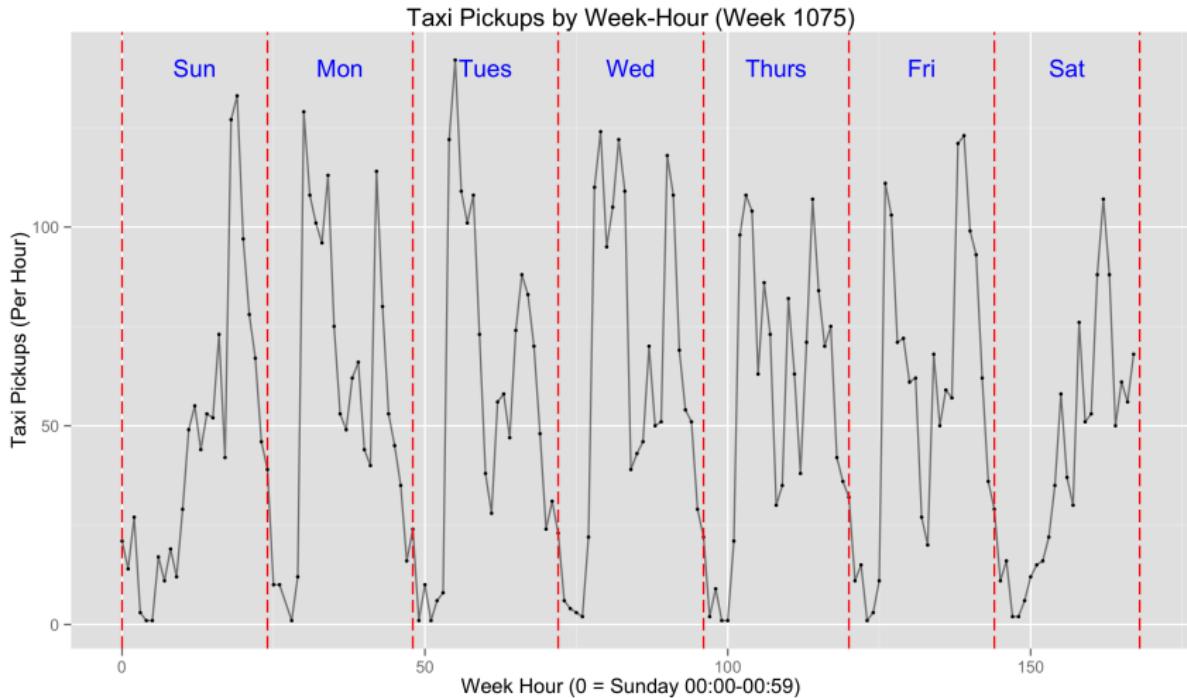


Google earth

© 2015 Google

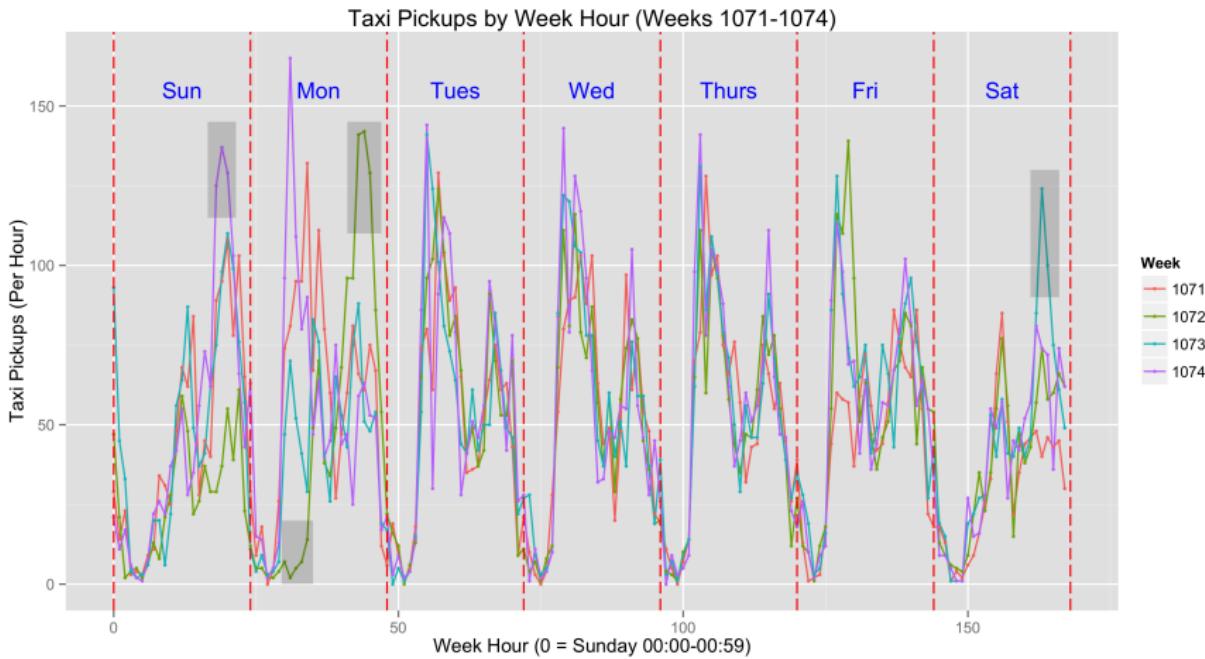
Data Visualization

Penn Station (Cell 7750): 1300 Taxi Pickups Per Day

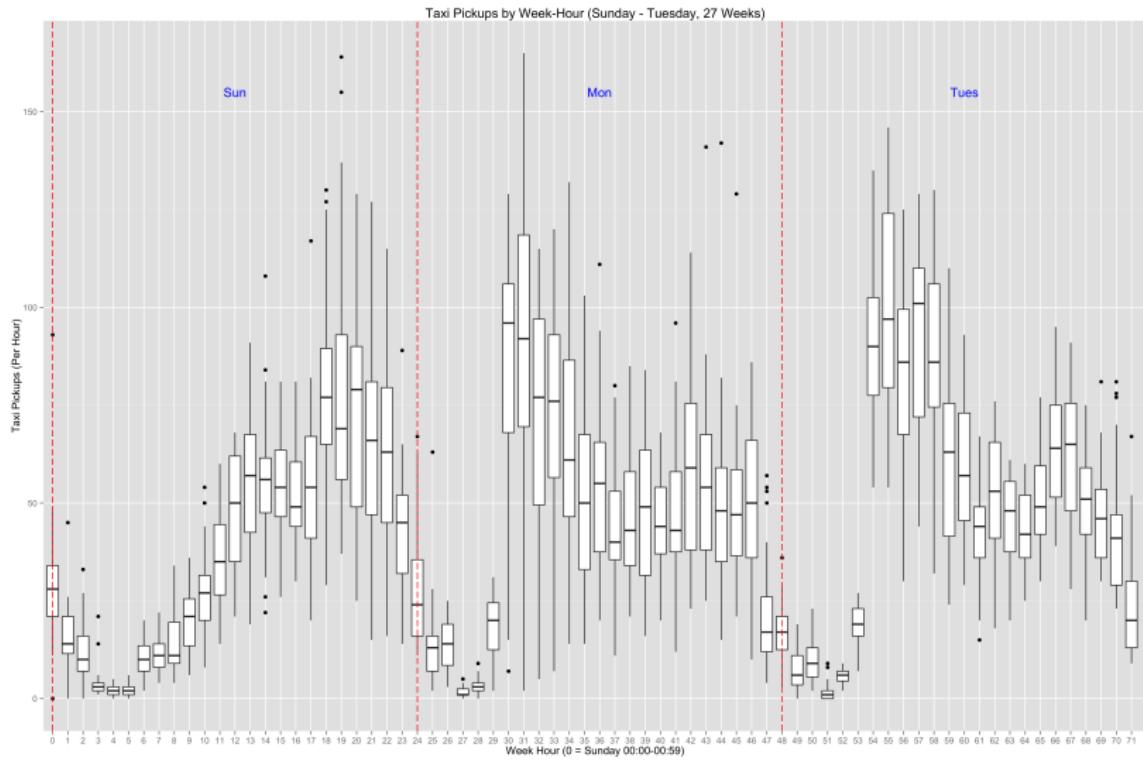


Note difference between weekend and weekday patterns.

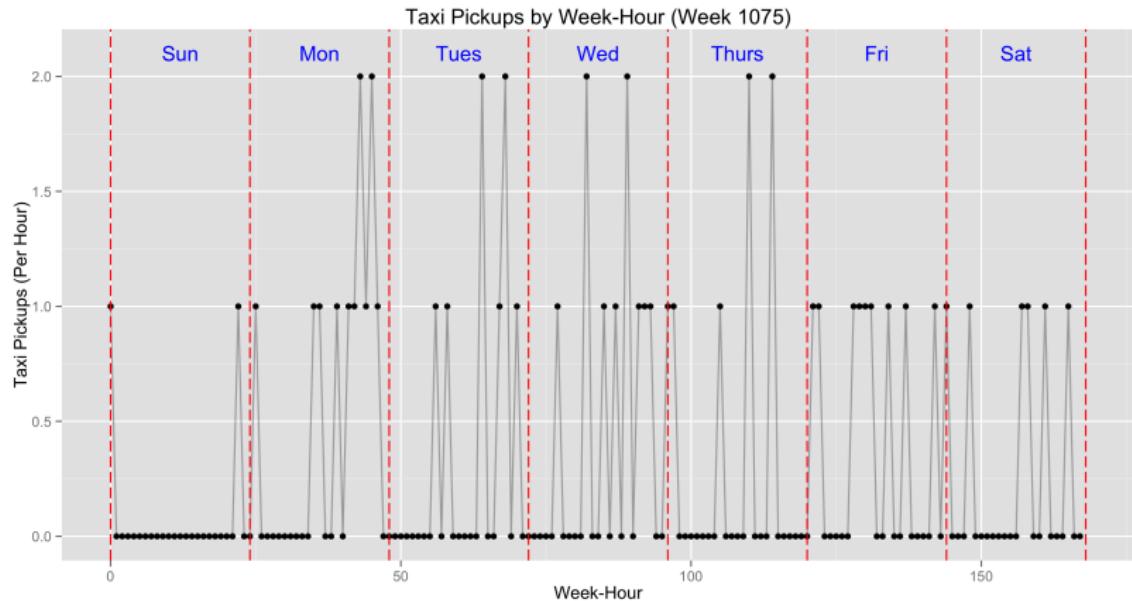
Penn Station (Cell 7750): Four Weeks, Some Outliers



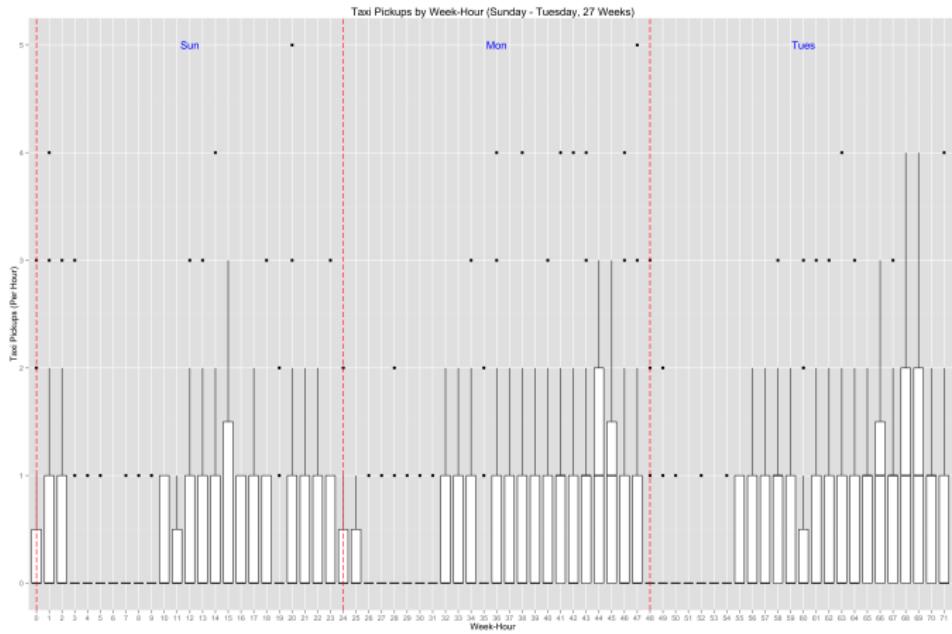
Penn Station: Sunday-Tuesday, 27 Weeks



Courant (Week 1075): 12 Taxi Pickups Per Day



Courant Institute: Sunday-Tuesday, 27 Weeks



Note: At least 25%, sometimes 75%+ of counts are zero.
Box plot clearly shows extreme values (ranging up to 5).

The Prediction Problem

The Prediction Problem

Somebody queries a **grid cell** and a **week-hour**, we tell them what to expect.

- Input space: $\mathcal{X} = \{(g, h) \mid g \in \{1, \dots, 398245\} \text{ and } h \in \{0, \dots, 167\}\}$, where
 - g is the grid Cell ID and
 - h is the week-hour
 - Possible future inputs: Holiday? Raining? Special event?
- Action space: $\mathcal{A} = \{\text{Probability distributions on number of pickups}\}$
- Output space: $\mathcal{Y} = \{0, 1, 2, 3, \dots\}$
 - Actual number of taxi pickups.
- Evaluation? Loss function? We'll come back to these questions...

Setting up the Learning Problem

- Labeled data look like:
 - (Grid Cell = 10321, Week Hour = 120) \mapsto Count = 3
 - (Grid Cell = 192001, Week Hour = 6) \mapsto Count = 12
 - (Grid Cell = 1271, Week Hour = 154) \mapsto Count = 0
- How to split the data into a training set and a test set?

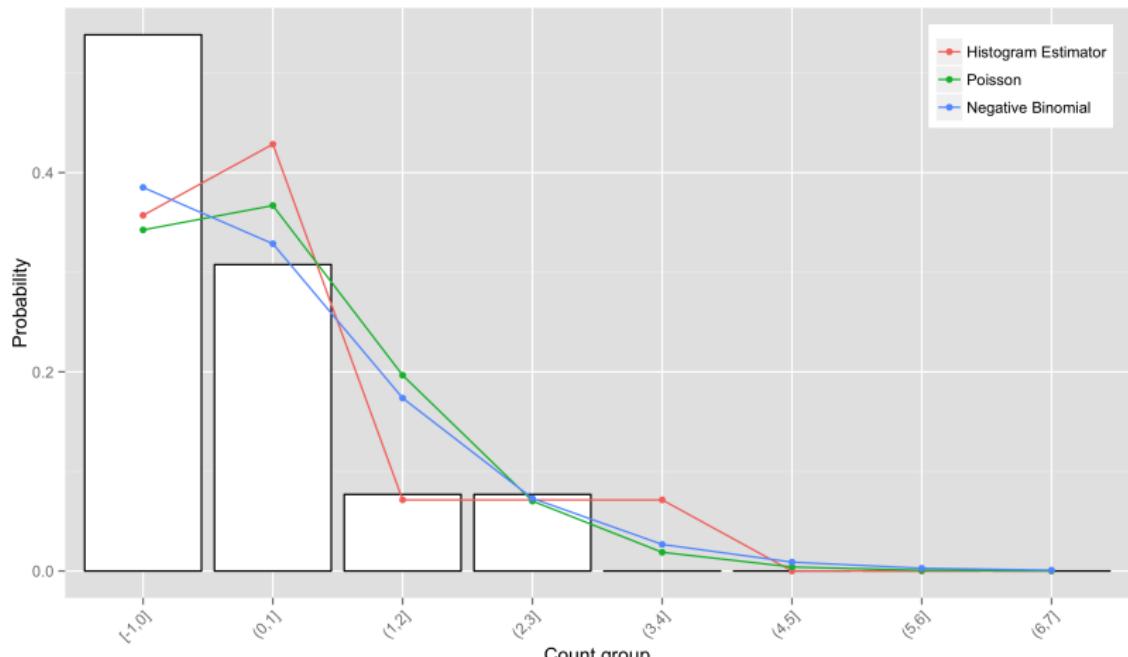
Train / Test Splits For Time-Indexed Datasets

- First idea is to randomly split instances into training and test.
- This does not reflect reality at deployment time:
 - Model trained on historical data from before deployment
 - With random split, training examples can occur after test examples.
 - For time series prediction (e.g. stock price prediction),
 - Usually want to predict some amount of time into the future.
 - Training and test data should reflect that application.
- Our approach:
 - First 14 weeks are **training set**.
 - Last 13 weeks are **test set**.

Stratification Approaches

Approach 1: Full Stratification (Courant, Tuesdays 7-8pm)

- Estimate distribution for each grid cell / week hour pair.
- Colored lines are from training. White bars are from test.



Terminology: Stratification and Bucketing

Definition

We say we are **stratifying** if we partition our input space into groups, and treat each group separately. For example, in modeling we would build a separate model for each group, without information sharing across groups.

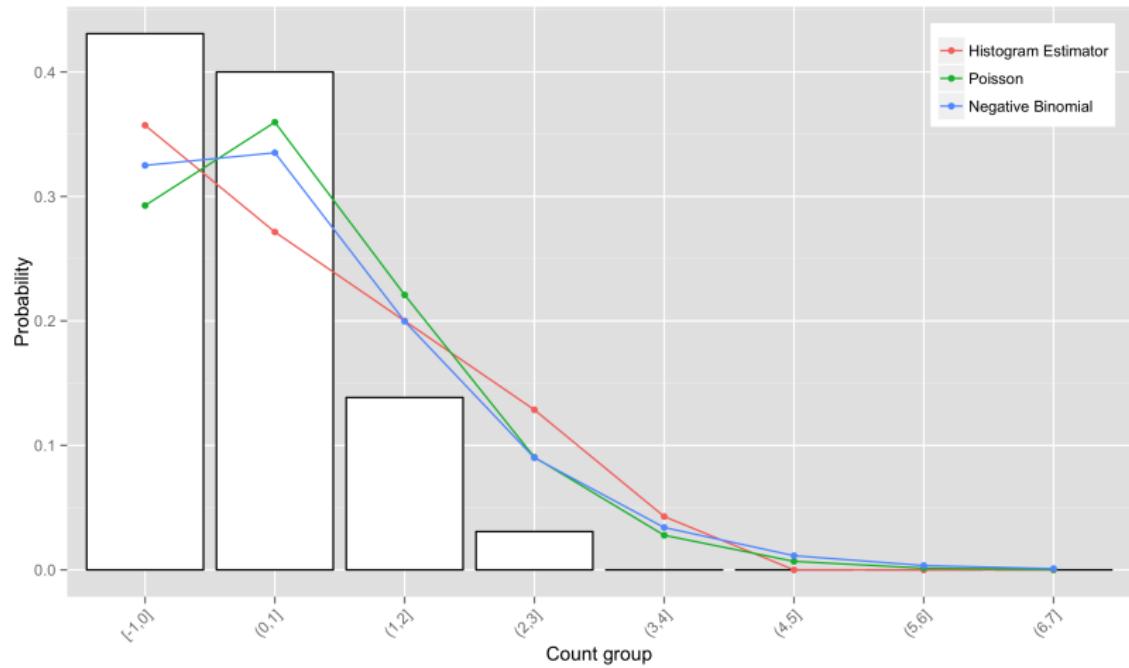
On the other hand,

Definition

We say we are **bucketing** (or **binning**) if we are combining natural groups in the data into a single group, rather than building a separate model for each group. For example, combining all weekdays together would be “bucketing”.

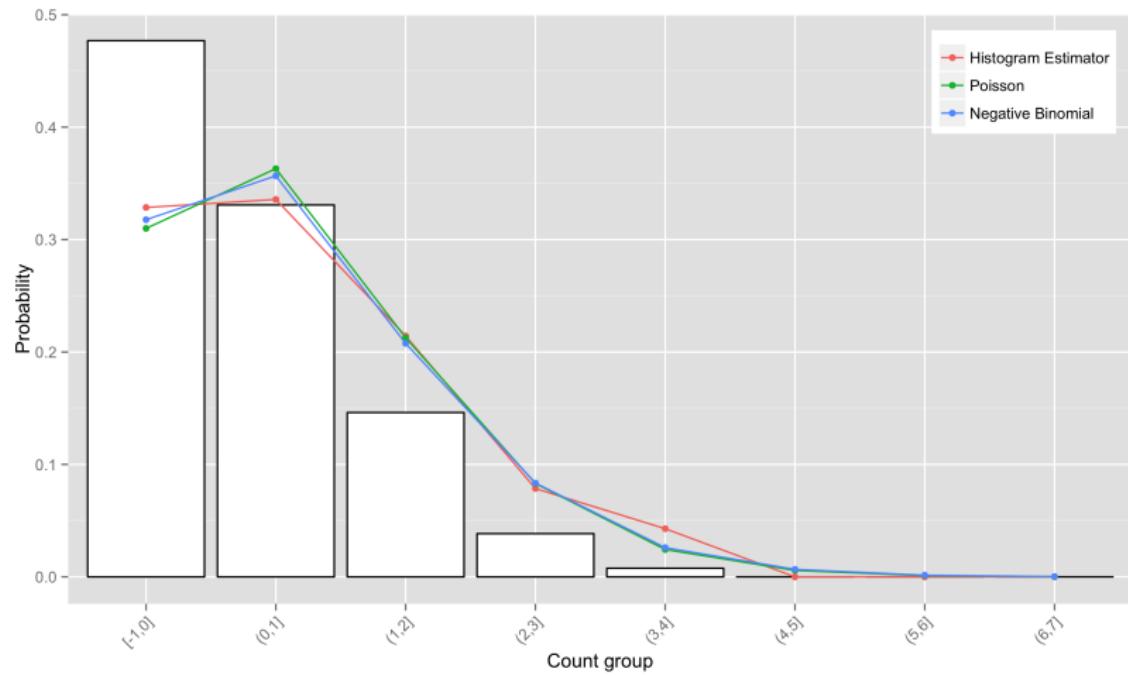
Approach 2: Weekday Bucketing (Courant, M-F 7-8pm)

- Data inspection suggests that day patterns are similar Mon-Fri.



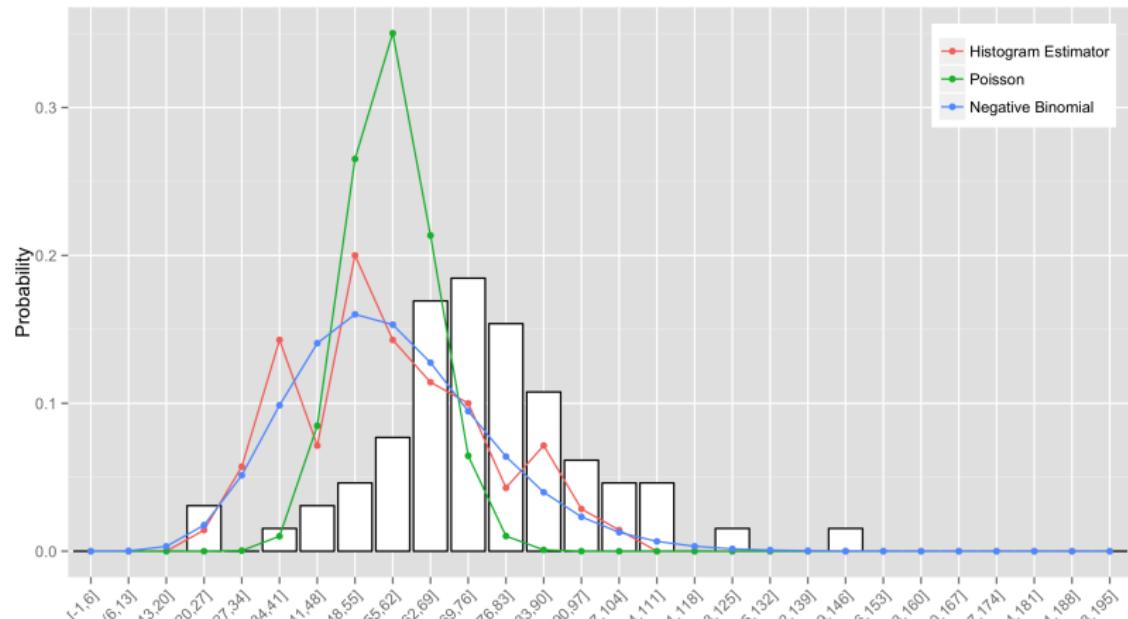
Approach 3: (Courant, M-F 6-8pm)

- Also, 6-7pm looks similar to 7-8pm, so join together



Penn Station, M-F 7-8pm

- Negative binomial fits empirical much better than Poisson.
(overdispersion)
- Massive shift between train and test!



The Bias / Variance Tradeoff of Stratification

- With a separate model for every grid cell / week-hour pair, model is highly specific!
- Could capture idiosyncrasy of Friday @5pm that we would miss if combining all weekdays.
 - That is, we're minimizing the bias.
- With relatively little data in a particular stratum, estimates will have high variance.
- By "bucketing", or combining strata:
 - We can reduce variance.
 - It may cost us in bias.
 - By bucketing in a smart way, you can minimize bias increase.

Is there a more convenient way?

- We can tradeoff between bias and variance by varying the stratification and the bucketing.
- It's a great way to start your data analysis.
 - You get a feel for the data and gain some intuition.
- This technique can be used for classification and regression as well.
- Our classification and regression techniques also trade off between bias and variance:
 - We had to choose our features.
 - We had to tune our regularization parameter.
- Can we do something similar for predicting distributions?
- Yes – this is **generalized regression**, where the action space is a distribution over outcomes...