Real-Time Trip Information Service for a Large Taxi Fleet

Rajesh Krishna Balan
School of Information Systems, Singapore Management University
Joint work with Nguyen Xuan Khoa and Jiang Lingxiao

High-Level Idea & Solution

• Question: Accurately predict the expected fare and time for any given taxi trip.
  – Mean Error under USD $0.80 and 3 minutes

• Solution: Use historical taxi trip data
  – Factors in optimal routing and road pricing

• Challenge: Real data is *REALLY* noisy!
  – Solution: Deploy a full complement of filters, heuristics, and enhancements

Background

• Singapore has three different world class public transportation systems
  – Buses (2 operators)
  – Train (MRT)
  – Taxis (6 different operators)

• Taxis can be flagged on the streets
  – Relatively cheap (~USD 6-10 per fare)
  – Found everywhere (like regular traffic)

Taxi System

• ~ 25,000 taxicabs
  ~ 55,000 drivers

• Owned by 7 companies
  – 2 largest owned by the same parent company

• Companies rent taxis to drivers
  • Drivers earn by collecting metered fares
  • Fares have a base amount, distance-based charges, time-based charges, and location-based charges
Data Collected
- From one of the taxi companies
  - Owning about 15,000 taxis
  - All taxis have a GPS receiver and a communication link (either 3G or radio) to a back-end server

- Data Collected
  - Instantaneous updates
  - Trip information (one record for every paid trip)
  - Booking information (not used in this work)

Data Collected (more details)
- Instantaneous updates
  - Location (GPS coordinates)
  - State (Free, Person-on-board, Oncall, Busy)
  - Speed (not very accurate)

- Trip Information (for every paid trip)
  - Starting point (GPS coordinates)
  - Ending point (GPS coordinates)
  - Distance
  - Time

Testing Methodology
- Use 21 months of data (Jan 09 – Aug 10)
  - ~12 mill. rec. per mth. ~250 mill rec. in total
  - Entire Sep 10 data used as the test set
  - Remaining 20 months used for training

- For each test record, compare predicted values with actual values
  - For fare, distance, time
  - Calculate a difference measure
  - Compare the average difference for each predictor

Basic Data Accuracy Sanity Check
- Sampled Dataset (~10,000 observations ~0.08% of one day’s data)
Start Of Adventure - “Obvious” Solution

- Use existing routing systems like Google Maps etc. to find route & time
  - Extrapolate fare from this

- Does not work well in practice
  - Tested on 12 million trips
  - Mean error about 35% for time and 40% for fare

- Has practical deployment issues
  - Operator needs a SLA with Google for a “good to have” service (additional cost!)

Solution: Use Human Knowledge

- Drivers know & use the optimal routes

- Use historical data to predict upcoming trip stats

- Given enough data, anomalies (inefficient routing) will be factored out
  - Factors in time of day, road pricing, and other effects

Key Challenges

1. Identifying the right runtime values
   a. Location size to use
   b. Prediction model to
   c. Amount of history to use

2. Finding and eliminating sources of errors

3. Must be production ready
   - Real time perf. on commodity hardware
   - Work well on all inputs

Challenge 1a: Zoning Singapore

- 50 km across by 25 km wide
  - ~ 710 km$^2$ (~275 miles$^2$)

- Too many roads to handle individually

- Taxi company split Singapore into 87 zones
  - Of different sizes, shapes, and popularity
  - 7.85 km$^2$ (stdev. 9.23) average area
  - 13.47 km (stdev. 8.33) average perimeter
  - Inefficient in practice (~30% error for fare and time)
Solution 1a: Small Fixed-Size Zones

- Tested initially using spatial queries
  
  "Find all trips within 100 meters of starting point that end within 100 meters of ending point"
  
  - Slow. Spatial queries are not fast
  - Fragile. Frequent problems with insufficient data

- Solution: Use static partitions instead
  
  - Ranging in size from 50x50m to 500x500m
  - ~162K possibilities for 50x50m
  - Full space is 162K²

End-to-End: Static Zoning Solution

1. Pre-process data
   - Add static zones, day of the week, trip time

2. Generate per-month histories
   - Using various predictors (time of day etc.)
     - Max, min, stdev, mean

3. Generate accumulated history
   - Multi-month history
   - Use partial derivatives for means and stdevs

4. Perform real-time predictions

Static Zoning Results (Fare)

Sigh... Nothing is That Easy!!

- LOC
- DOW
- PEAK
- HR
- DOW x HR

Average Prediction Error (cents)

Zone Size (metres)

- 250m
- 200m
- 150m
- 100m
- 50m

Hit Rate (%)

Number of Months
The Great Adventure: Story Up To Now

- Spatial queries are slow
  - Use pre-defined small zones
  - Accuracy is really good
  - Hit rate is terrible

- Solution: Use dynamic clustering
  - KD-Trees under the covers
  - Can benefit from extra months of data
  - Big memory hog requiring careful tuning
  - Accuracy is not as high as we would like

Dynamic Zoning: Sources of Error

1. Non-standard Routes
   - Drivers who take longer than expected routes

2. Non Point-to-Point Routes
   - Far more common than we thought!

3. Rain!! Rain!! Rain!!
   - Thunderstorms are the norm
   - Has a big impact on traffic conditions
Real World Deployment Lessons 1

• Choice of Database Matters!
  – The seductive appeal of MySQL

• Spatial Queries are slow
  – Using zones is 2 orders of magnitude faster than doing spatial queries
    • Even with spatial indexes

• Simplicity wins
  – Especially when the output is millions of records
  – And when explaining to operators!

Real World Deployment Lessons 2

• Real data is quite noisy
  – Proper cleaning is needed to prevent anomalies

• Deploying a real system takes a LONG time
  – So many levels of validation and verification
  – Deployed system frequently 1 or 2 levels behind research version
  – Very hard to instrument production systems
  – Simplicity wins over many other things
    • Especially to end users and maintenance folks

Main Takeaways & Future Work

• Possible to predict taxi trips from history
  – Enough trip duplication & density in Singapore
  – Still requires “fuzzy” zoning to really work

• “Errors” can significantly affect accuracy
  – External (rain) and internal (inefficient routing, non p2p routes) sources need to be tackled

• Upcoming: Tell drivers likely passenger loc.
  – Using just supply-side data
  – Distributed human routing & UI problems
Thank You!