Web Mining and Recommender Systems

Temporal data mining
This week we’ll look back on some of the topics already covered in this class, and see how they can be adapted to make use of temporal information.

1. **Regression** – sliding windows and autoregression
2. **Social networks** – densification over time
3. **Text mining** – “Topics over Time”
4. **Recommender systems** – some results from Koren
Web Mining and Recommender Systems

Regression for sequence data
Week 1 – Regression

Given **labeled training data** of the form

\[ \{(data_1, label_1), \ldots, (data_n, label_n)\} \]

Infer the function

\[ f(data) \rightarrow \text{labels} \]
Time-series regression

Here, we’d like to predict sequences of real-valued events as accurately as possible.
Time-series regression

**Method 1:** maintain a “moving average” using a window of some fixed length

\[ f(x_1, \ldots, x_m) = \frac{\sum_{k=0}^{K} x_{n-k}}{K} \]
**Method 1:** maintain a “moving average” using a window of some fixed length

- This can be computed efficiently via dynamic programming:

\[
f(x_1, \ldots, x_{m+1}) = \frac{K f(x_1, \ldots, x_m) - x_{n-K+1} + x_{n+1}}{K}
\]

\(O(n+K)\) vs. \(O(n \times K)\)
Time-series regression

Also useful to plot data:

BeeraAdvocate, ratings over time

Scatterplot

Sliding window (K=10000)

long-term trends

seasonal effects

Code on:
http://jmcauley.ucsd.edu/code/week10.py
**Method 2:** weight the points in the moving average by age

\[
f(x_1, \ldots, x_m) = \frac{\sum_{k=0}^{K} (K-k)x_{n-k}}{1+2+\ldots+K}
\]
Method 3: weight the most recent points exponentially higher

\[ f(x_1) = x_1 \]

\[ f(x_1, \ldots, x_m) = \alpha f(x_1 + \ldots + x_{n-1}) + (1-\alpha)x_n \]
Methods 1, 2, 3

Method 1: Sliding window
Method 2: Linear decay
Method 3: Exponential decay
Method 4: all of these models are assigning **weights** to previous values using some predefined scheme, why not just **learn** the weights?

\[
f(x_1, \ldots, x_m) = \sum_{k=0}^{k-1} \Theta_k x_{n-k} + \Theta_0 x_n + \cdots + \Theta_{k-1} x_{n-1} \\
\Theta = \arg\min_{\Theta} \frac{1}{m} \sum_{n=1}^{m} (f(x_1, \ldots, x_n) - x_{n+1})^2
\]
**Method 4:** all of these models are assigning **weights** to previous values using some predefined scheme, why not just **learn** the weights?

- We can now fit this model using least-squares
- This procedure is known as **autoregression**
- Using this model, we can capture **periodic** effects, e.g. that the traffic of a website is most similar to its traffic 7 days ago
Web Mining and Recommender Systems

Temporal dynamics of social networks
How can we **characterize, model, and reason about** the structure of social networks?

1. Models of network structure
2. Power-laws and scale-free networks, “rich-get-richer” phenomena
3. Triadic closure and “the strength of weak ties”
4. Small-world phenomena
5. Hubs & Authorities; PageRank
Two weeks ago we saw some processes that model the generation of social and information networks

- Power-laws & small worlds
- Random graph models

These were all defined with a “static” network in mind. But if we observe the order in which edges were created, we can study how these phenomena change as a function of time

First, let’s look at “microscopic” evolution, i.e., evolution in terms of individual nodes in the network
**Q1:** How do networks grow in terms of the number of nodes over time?

- **Flickr** (exponential)
- **Del.icio.us** (linear)
- **Linkedln** (exponential)
- **Answers** (sub-linear)

(from Leskovec, 2008 (CMU Thesis))

**A:** Doesn’t seem to be an obvious trend, so what do networks have in common as they evolve?
Temporal dynamics of social networks

Q2: When do nodes create links?
- x-axis is the age of the nodes
- y-axis is the number of edges created at that age

A: In most networks there’s a “burst” of initial edge creation which gradually flattens out. Very different behavior on LinkedIn (guesses as to why?)
**Q3:** How long do nodes “live”?

- x-axis is the diff. between date of last and first edge creation
- y-axis is the frequency

**A:** Node lifetimes follow a power-law: many many nodes are shortlived, with a long-tail of older nodes.
Temporal dynamics of social networks

What about “macroscopic” evolution, i.e., how do global properties of networks change over time?

**Q1:** How does the # of nodes relate to the # of edges?

\[ E = 0.0 \times N^{1.69} \]

- A few more networks: citations, authorship, and autonomous systems (and some others, not shown)
- **A:** Seems to be linear (on a log-log plot) **but** the number of edges grows **faster** than the number of nodes as a function of time
Q1: How does the # of nodes relate to the # of edges?  
A: seems to behave like

\[ E(t) \propto N(t)^a \]

where

\[ 1 \leq a \leq 2 \]

• \( a = 1 \) would correspond to constant out-degree – which is what we might traditionally assume
• \( a = 2 \) would correspond to the graph being fully connected
• What seems to be the case from the previous examples is that \( a > 1 \) – the number of edges grows faster than the number of nodes
Q2: How does the degree change over time?

A: The average out-degree increases over time
**Q3:** If the network becomes **denser**, what happens to the (effective) diameter?

- **A:** The diameter seems to decrease.
- In other words, the network becomes **more** of a small world as the number of nodes increases.
Q4: Is this something that **must** happen – i.e., if the number of edges increases faster than the number of nodes, does that mean that the diameter must decrease?

A: Let’s construct random graphs (with $a > 1$) to test this:
So, a decreasing diameter is **not** a “rule” of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model.

**Q5:** is the degree distribution of the nodes sufficient to explain the observed phenomenon?

**A:** Let’s perform **random rewiring** to test this.

Random rewiring preserves the degree distribution, and randomly samples amongst networks with observed degree distribution.
So, a decreasing diameter is not a “rule” of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model.

Q5: is the degree distribution of the nodes sufficient to explain the observed phenomenon?

(c) Affiliation network (ATP-ASTRO-PH)

(d) US patent citation network (CIT-PATENTS)
So, a decreasing diameter is **not** a “rule” of a network whose number of edges grows faster than its number of nodes, though it is consistent with a preferential attachment model.

**Q5:** is the degree distribution of the nodes sufficient to explain the observed phenomenon?

**A:** Yes! The fact that real-world networks seem to have decreasing diameter over time can be explained as a result of their degree distribution **and** the fact that the number of edges grows faster than the number of nodes.
Temporal dynamics of social networks

Other interesting topics...

"memetracker"
Temporal dynamics of social networks

Other interesting topics...

Aligning query data with disease data – Google flu trends:
https://www.google.org/flutrends/us/#US

Sodium content in recipe searches vs. # of heart failure patients – “From Cookies to Cooks” (West et al. 2013):
Further reading:

“Dynamics of Large Networks” (most plots from here)
Jure Leskovec, 2008

“Microscopic Evolution of Social Networks”
Leskovec et al. 2008

“Graph Evolution: Densification and Shrinking Diameters”
Leskovec et al. 2007
Web Mining and Recommender Systems

Temporal dynamics of text
**Bag-of-Words** representations of text:

$$F_{text} = [150, 0, 0, 0, 0, 0, 0, \ldots, 0]$$

- a
- aardvark
- zoetrope
Latent Dirichlet Allocation

In week 5, we tried to develop low-dimensional representations of documents:

What we would like:

87 of 102 people found the following review helpful

★★★★★ You keep what you kill, December 27, 2004
By Schtinky "Schtinky" (Washington State) - See all my reviews

This review is from: The Chronicles of Riddick [Widescreen Unrated Director's Cut] [DVD]

Even if I have to apologize to my Friends and Favorites, and my family, I have to admit that I really liked this movie. It's a Sci-Fi movie with a "Mad Maxx" appeal that, while changing many things, left Riddick from 'Pitch Black' to be just Riddick. They did not change his attitude or soften him up or bring him out of his original character, which was very pleasing to 'Pitch Black' fans like myself.

First off, let me say that when playing the DVD, the first selection to come up is Convert or Fight, and no explanation of the choices. This confused me at first, so I will mention off the bat that they are simply different menu formats, that each menu has the very same options, simply different background visuals. Select either one and continue with the movie.

(review of "The Chronicles of Riddick"")
We saw how **LDA** can be used to describe documents in terms of **topics**

- Each document has a **topic vector** (a stochastic vector describing the fraction of words that discuss each topic)
- Each topic has a **word vector** (a stochastic vector describing how often a particular word is used in that topic)
Topics and documents are both described using stochastic vectors:

Each document has a topic distribution which is a mixture over the topics it discusses:

$$\theta_d \in \Delta^K \text{ i.e., } \forall d \sum_k \theta_{d,k} = 1$$

Each topic has a word distribution which is a mixture over the words it discusses:

$$\phi_k \in \Delta^D \text{ i.e., } \forall k \sum_w \phi_{k,w} = 1$$
**Topics over Time** (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models.

- The topics discussed in conference proceedings progressed from neural networks, towards SVMs and structured prediction (and back to neural networks).
- The topics used in political discourse now cover science and technology more than they did in the 1700s.
- Within an institution, e-mails will discuss different topics (e.g., recruiting, conference deadlines) at different times of the year.


**Topics over Time** (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models

The ToT model is similar to LDA with one addition:

1. For each topic $K$, draw a word vector $\phi_k$ from $\text{Dir.}(\beta)$
2. For each document $d$, draw a topic vector $\theta_d$ from $\text{Dir.}(\alpha)$
3. For each word position $i$:
   1. draw a topic $z_{di}$ from multinomial $\theta_d$
   2. draw a word $w_{di}$ from multinomial $\phi_{z_{di}}$
   3. **draw a timestamp $t_{di}$ from Beta($\psi_{z_{di}}$)**
Latent Dirichlet Allocation

**Topics over Time** (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models

3.3. draw a timestamp $t_{di}$ from $\text{Beta}(\psi_{z_{di}})$

- There is now one Beta distribution per topic
- Inference is still done by Gibbs sampling, with an outer loop to update the Beta distribution parameters

Beta distributions are a flexible family of distributions that can capture several types of behavior – e.g. gradual increase, gradual decline, or temporary “bursts”

$$p.d.f.: \quad x^{\alpha-1} (1-x)^{\beta-1} / B(\alpha, \beta)$$
**Results:**
Political addresses – the model seems to capture realistic “bursty” and gradually emerging topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Mexican War</th>
<th>Panama Canal</th>
<th>Cold War</th>
<th>Modern Tech</th>
</tr>
</thead>
<tbody>
<tr>
<td>states</td>
<td>0.02032</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mexico</td>
<td>0.01832</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>government</td>
<td>0.01670</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>united</td>
<td>0.01521</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>war</td>
<td>0.01059</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>congress</td>
<td>0.00951</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>0.00906</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>texas</td>
<td>0.00852</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>made</td>
<td>0.00727</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>great</td>
<td>0.00611</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

fitted Beta distribution to this topic
## Results:

e-mails & conference proceedings

<table>
<thead>
<tr>
<th>Faculty Recruiting</th>
<th>ART Paper</th>
<th>MALLET</th>
<th>CVS Operations</th>
<th>Recurrent NN</th>
<th>Game Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs</td>
<td>xuernui</td>
<td>code</td>
<td>check</td>
<td>state</td>
<td>game</td>
</tr>
<tr>
<td>april</td>
<td>data</td>
<td>files</td>
<td>page</td>
<td>recurrent</td>
<td>strategy</td>
</tr>
<tr>
<td>faculty</td>
<td>word</td>
<td>mallet</td>
<td>version</td>
<td>sequence</td>
<td>play</td>
</tr>
<tr>
<td>david</td>
<td>research</td>
<td>java</td>
<td>cvs</td>
<td>sequences</td>
<td>games</td>
</tr>
<tr>
<td>lunch</td>
<td>topic</td>
<td>file</td>
<td>add</td>
<td>time</td>
<td>player</td>
</tr>
<tr>
<td>schedule</td>
<td>model</td>
<td>al</td>
<td>update</td>
<td>states</td>
<td>agents</td>
</tr>
<tr>
<td>candidate</td>
<td>andres</td>
<td>directory</td>
<td>latest</td>
<td>transition</td>
<td>expert</td>
</tr>
<tr>
<td>talk</td>
<td>sample</td>
<td>version</td>
<td>updated</td>
<td>finite</td>
<td>strategies</td>
</tr>
<tr>
<td>bruce</td>
<td>enron</td>
<td>pdf</td>
<td>checked</td>
<td>length</td>
<td>opponent...</td>
</tr>
<tr>
<td>visit</td>
<td>dataset</td>
<td>bug</td>
<td>change</td>
<td>strings</td>
<td>nash</td>
</tr>
</tbody>
</table>

| Probability | 0.03572   | 0.02113   | 0.05668   | 0.04473   |
| Probability | 0.02724   | 0.01814   | 0.04212   | 0.04070   |
| Probability | 0.02341   | 0.01601   | 0.04073   | 0.03828   |
| Probability | 0.02012   | 0.01408   | 0.03085   | 0.03587   |
| Probability | 0.01766   | 0.01366   | 0.02947   | 0.03083   |
| Probability | 0.01656   | 0.01238   | 0.02479   | 0.02539   |
| Probability | 0.01560   | 0.01238   | 0.02080   | 0.02519   |
| Probability | 0.01355   | 0.01152   | 0.01664   | 0.02317   |
| Probability | 0.01273   | 0.01067   | 0.01421   | 0.02277   |
| Probability | 0.01232   | 0.00960   | 0.01352   | 0.02156   |
| Probability | 0.05963   |           |           |           |
| Probability | 0.03765   |           |           |           |
| Probability | 0.03616   |           |           |           |
| Probability | 0.02462   |           |           |           |
| Probability | 0.02402   |           |           |           |
| Probability | 0.02057   |           |           |           |
| Probability | 0.01360   |           |           |           |
| Probability | 0.01242   |           |           |           |
| Probability | 0.01154   |           |           |           |
| Probability | 0.01013   |           |           |           |
| Probability |           |           |           | 0.02850   |
| Probability |           |           |           | 0.02378   |
| Probability |           |           |           | 0.01490   |
| Probability |           |           |           | 0.01473   |
| Probability |           |           |           | 0.01451   |
| Probability |           |           |           | 0.01346   |
| Probability |           |           |           | 0.01281   |
| Probability |           |           |           | 0.01123   |
| Probability |           |           |           | 0.01088   |
| Probability |           |           |           | 0.00848   |

Latent Dirichlet Allocation
Latent Dirichlet Allocation

Results:
conference proceedings (NIPS)

Relative weights of various topics in 17 years of NIPS proceedings
Further reading:
“Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends”
(Wang & McCallum, 2006)
http://people.cs.umass.edu/~mccallum/papers/tot-kdd06.pdf
Web Mining and Recommender Systems

Temporal recommender systems
**Recommender Systems** go beyond the methods we’ve seen so far by trying to model the **relationships** between people and the items they’re evaluating.

- Preference toward “action”
- Preference toward “special effects”

**Compatibility**

- Is the movie action-heavy?
- Are the special effects good?
Predict a user’s rating of an item according to:

\[ f(u, i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i \]

By solving the optimization problem:

\[
\arg\min_{\alpha, \beta, \gamma} \sum_{u,i} (\alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i - R_{u,i})^2 + \lambda \left[ \sum_u \beta_u^2 + \sum_i \beta_i^2 + \sum_i \|\gamma_i\|_2^2 + \sum_u \|\gamma_u\|_2^2 \right]
\]

(error) (regularizer)

(e.g. using stochastic gradient descent)
To build a reliable system (and to win the Netflix prize!) we need to account for **temporal dynamics**:

Netflix ratings by movie age (People tend to give higher ratings to older movies)

Netflix ratings over time (Netflix changed their interface)

So how was this actually done?

Figure from Koren: “Collaborative Filtering with Temporal Dynamics” (KDD 2009)
Temporal latent-factor models

To start with, let’s just assume that it’s only the bias terms that explain these types of temporal variation (which, for the examples on the previous slides, is potentially enough)

\[ b_{u,i}(t) = \alpha + \beta_u(t) + \beta_i(t) \]

Idea: temporal dynamics for items can be explained by long-term, gradual changes, whereas for users we’ll need a different model that allows for “bursty”, short-lived behavior
Temporal latent-factor models

temporal bias model:

\[ b_{u,i}(t) = \alpha + \beta_u(t) + \beta_i(t) \]

For item terms, just separate the dataset into (equally sized) bins:

\[ \beta_i(t) = \beta_i + \beta_i,\text{Bin}(t) \]

*in Koren’s paper they suggested ~30 bins corresponding to about 10 weeks each for Netflix or bins for periodic effects (e.g. the day of the week):

\[ \beta_i(t) = \beta_i + \beta_i,\text{Bin}(t) + \beta_i,\text{period}(t) \]

What about user terms?

• We need something much finer-grained

• But – for most users we have far too little data to fit very short term dynamics
Temporal latent-factor models

Start with a simple model of drifting dynamics for users:

\[ \text{dev}_u(t) = \text{sign}(t - t_u) \cdot |t - t_u|^x \]

- **mean** rating date for user \( u \)
- Hyperparameter (ended up as \( x=0.4 \) for Koren)
- Before (-1) or after (1) the mean date
- Days away from mean date
Temporal latent-factor models

Start with a simple model of drifting dynamics for users:

\[
\text{dev}_u(t) = \text{sign}(t - t_u) \cdot |t - t_u|^x
\]

- mean rating date for user \( u \)
- hyperparameter (ended up as \( x=0.4 \) for Koren)
- before \((-1)\) or after \((1)\) the mean date
- days away from mean date

Time-dependent user bias can then be defined as:

\[
\beta_u^{(1)}(t) = \beta_u + \alpha_u \cdot \text{dev}_u(t)
\]

- overall user bias
- sign and scale for deviation term
Temporal latent-factor models

Netflix ratings over time

Real data

Fitted model
Temporal latent-factor models

Time-dependent user bias can then be defined as:

$$\beta_u^{(1)}(t) = \beta_u + \alpha_u \cdot \text{dev}_u(t)$$

- Requires only two parameters per user and captures some notion of temporal “drift” (even if the model found through cross-validation is (to me) completely unintuitive)
- To develop a slightly more expressive model, we can interpolate smoothly between biases using splines
Temporal latent-factor models

\[ \beta_u^{(2)}(t) = \beta_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma |t-t_{ul}|} b_{tl}}{\sum_{l=1}^{k_u} e^{-\gamma |t-t_{ul}|}} \]

- number of control points for this user
  \( (k_u = n_u^{0.25} \text{ in Koren}) \)
- user bias associated with this control point
- time associated with control point (uniformly spaced)
Temporal latent-factor models

\[ \beta_u(t) = \beta_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma |t-t_l^u|} b_{tl}}{\sum_{l=1}^{k_u} e^{-\gamma |t-t_l^u|}} \]

- This is now a reasonably flexible model, but still only captures \textit{gradual drift}, i.e., it can’t handle sudden changes (e.g. a user simply having a bad day)
Temporal latent-factor models

- Koren got around this just by adding a “per-day” user bias:
  \[
  \beta_{u,t}
  \]
  bias for a particular day (or session)

- Of course, this is only useful for particular days in which users have a lot of (abnormal) activity
- The final (time-evolving bias) model then combines all of these factors:

\[
\beta_{u,i}(t) = \alpha + \beta_u + \alpha_u \cdot \text{dev}_u(t) + \beta_{u,t} + \beta_i + \beta_{i,Bin(t)}
\]
Finally, we can add a time-dependent scaling factor:

\[ \beta_{u,i}(t) = \alpha + \beta_u + \alpha_u \cdot \text{dev}_u(t) + \beta_{u,t} + (\beta_i + \beta_{i,\text{Bin}(t)}) \cdot c_u(t) \]

also defined as \( c_u + c_{u,t} \)

Latent factors can also be defined to evolve in the same way:

\[ \gamma_{u,k}(t) = \gamma_{u,k} + \alpha_{u,k} \cdot \text{dev}_u(t) + \gamma_{u,k,t} \]

factor-dependent user drift

factor-dependent short-term effects
Summary

• Effective modeling of temporal factors was absolutely critical to this solution outperforming alternatives on Netflix’s data.
• In fact, even with only temporally evolving bias terms, their solution was already ahead of Netflix’s previous (“Cinematch”) model.

On the other hand...

• Many of the ideas here depend on dynamics that are quite specific to “Netflix-like” settings.
• Some factors (e.g. short-term effects) depend on a high density of data per-user and per-item, which is not always available.
Summary

- Changing the setting, e.g. to model the stages of progression through the symptoms of a disease, or even to model the temporal progression of people’s opinions on beers, means that alternate temporal models are required.

(rows: models of increasingly “experienced” users)

(columns: review timeline for one user)
Further reading:
“Collaborative filtering with temporal dynamics”
Yehuda Koren, 2009
Web Mining and Recommender Systems

Incredible assignments
Multiclass classification (four common sport types). Predictive features include:

- Altitude (mountain vs. road biking)
- Speed
- Time (e.g. commuting is short)
- Variation in speed (e.g. for mountain biking)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Recorded speed in Miles per Hour</td>
</tr>
<tr>
<td>Altitude</td>
<td>Recorded altitude in Meters</td>
</tr>
<tr>
<td>Heart Rate</td>
<td>Recorded heart rate in Beats per Minute</td>
</tr>
<tr>
<td>Timestamp</td>
<td>UNIX timestamp</td>
</tr>
<tr>
<td>Longitude</td>
<td>Recorded longitude</td>
</tr>
<tr>
<td>Latitude</td>
<td>A Recorded latitude</td>
</tr>
<tr>
<td>ID</td>
<td>ID of this workout</td>
</tr>
<tr>
<td>URL</td>
<td>URL of this workout</td>
</tr>
<tr>
<td>User ID</td>
<td>ID of the user</td>
</tr>
<tr>
<td>Sport</td>
<td>Type of sport that user engages in</td>
</tr>
<tr>
<td>Gender</td>
<td>Male/Female/Unknown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>Accuracy</th>
<th>Balanced Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>Baseline</td>
<td>0.774</td>
<td>0.435</td>
</tr>
<tr>
<td>KNN</td>
<td>Baseline</td>
<td>0.779</td>
<td>0.442</td>
</tr>
<tr>
<td>Random Forest</td>
<td>Baseline</td>
<td>0.759</td>
<td>0.454</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>Engineered</td>
<td>0.826</td>
<td>0.465</td>
</tr>
<tr>
<td>KNN</td>
<td>Engineered</td>
<td>0.797</td>
<td>0.572</td>
</tr>
<tr>
<td>Random Forest</td>
<td>Engineered</td>
<td>0.902</td>
<td>0.705</td>
</tr>
</tbody>
</table>
Spatially Inspired Price Prediction for Car Rentals

- **Turo** (peer to peer rentals)
- 36,000 rental datapoints from a public github
- Use lat/lon data to extract zipcodes *(uszipcode library)*, and combine this with census data from *census.gov* to extract median incomes
- Scrape *Google Trends* listings to determine the popularity of each car

Extracted features:
- UserID/carID/rating
- Time to respond to a rental request
- Weekday, month
- Car popularity
- Etc.

Random Forest classifier:
\[ R^2 = 0.6115 \]

Farhood Ensan  
Kaushik Ganapathy  
Jiaxi Lei
Airline Flight Delay Prediction

- Predict delays at LAX
- Temporal features, airline features, geographical features

Accuracy ~0.65
F1 ~0.55

Delay vs. day/month
Delay vs. destination
Delay vs. airline

Yiluo Qin
Yijun Liu
Yu-Chieh Chen
AirBnB Price-Per Prediction

• 45,053 LA AirBnB listings from "Inside AirBnB"
• 85,273 London listings
• 48,895 NY listings

Features include:
• Geo / neighborhood
• Room types / # guests
• Amenities
• Ratings
• Description word-clouds
• Etc.

Price per neighborhood

Number of guests accommodated

Log(1 + price)
Predicting Passenger Flow

- Estimate number of passengers on Hangzhou Metro
- 70 million records (!) from 5 million passengers

Commuter ratio distribution

- Predict "flow" (e.g. # of passengers entering and exiting a station, # of passengers on a particular "edge")
- Features are mostly temporal, considering various granularities

<table>
<thead>
<tr>
<th>time</th>
<th>line ID</th>
<th>station ID</th>
<th>device ID</th>
<th>status</th>
<th>user ID</th>
<th>pay type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2 0:00</td>
<td>C</td>
<td>39</td>
<td>1824</td>
<td>0</td>
<td>B958313</td>
<td>1</td>
</tr>
<tr>
<td>1/2 0:01</td>
<td>B</td>
<td>8</td>
<td>384</td>
<td>0</td>
<td>Bdd932c</td>
<td>1</td>
</tr>
<tr>
<td>1/2 0:01</td>
<td>B</td>
<td>2</td>
<td>74</td>
<td>0</td>
<td>B32a6c9</td>
<td>1</td>
</tr>
<tr>
<td>1/2 0:02</td>
<td>C</td>
<td>55</td>
<td>2630</td>
<td>0</td>
<td>B18f450</td>
<td>1</td>
</tr>
</tbody>
</table>

Daily traffic for different days

<table>
<thead>
<tr>
<th>prediction method</th>
<th>station flow MSE</th>
<th>traffic flow MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (average of history)</td>
<td>2378.61</td>
<td>46611.7643</td>
</tr>
<tr>
<td>Naive Linear Regression</td>
<td>Worse than baseline</td>
<td>169034.8235</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>2741.42</td>
<td>125191.3970</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>2210.90</td>
<td>117560.0111</td>
</tr>
<tr>
<td>Polynomial Feature degree=2</td>
<td>1193.36</td>
<td>36116.8772</td>
</tr>
<tr>
<td>Random Forest Time as original value</td>
<td>890.91</td>
<td>32744.0437</td>
</tr>
</tbody>
</table>
New York City Taxi Fare Prediction

- Predict the total fare of a taxi trip
- 5,000,000 pickup/dropoff datapoints
- MSE and MAPE (Mean Absolute Percentage Error)

Fare distribution

<table>
<thead>
<tr>
<th>Features</th>
<th>Explanation</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLatDiff</td>
<td>Absolute difference in latitude</td>
<td>Baseline, Linear Regression, Random Forest</td>
</tr>
<tr>
<td>AbsLonDiff</td>
<td>Absolute difference in longitude</td>
<td>Baseline, Linear Regression, Random Forest</td>
</tr>
<tr>
<td>Passenger_count</td>
<td>Number of passengers per ride</td>
<td>Linear Regression, Random Forest</td>
</tr>
<tr>
<td>Haversine</td>
<td>Distance metrics taking into account the spherical shape of the Earth</td>
<td>Linear Regression, Random Forest</td>
</tr>
<tr>
<td>Fare-bin</td>
<td>Bin range of the fair amount</td>
<td>Upgraded liner regression</td>
</tr>
<tr>
<td>Color</td>
<td>Color of the car</td>
<td></td>
</tr>
<tr>
<td>distance</td>
<td>Sphere distance of pickup and drop-off locations</td>
<td>LGBM</td>
</tr>
<tr>
<td>bearing</td>
<td>Bearing distance of pickup and drop-off locations</td>
<td>LGBM</td>
</tr>
<tr>
<td>Pickup_latitude, pickup_long</td>
<td>Pickup location</td>
<td>LGBM</td>
</tr>
<tr>
<td>itude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dropoff_latitude, Dropoff_lon</td>
<td>Dropoff location</td>
<td>LGBM</td>
</tr>
<tr>
<td>gitude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hour, day, month, weekday,</td>
<td>Hour, day, month, weekday, year of pickup time</td>
<td>LGBM</td>
</tr>
<tr>
<td>year</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Predicting Wave Height using Embedded Sensors on Surfboards

"Smartfin" data from 135 surf sessions
• Accelerometer (A), Gyroscope (G), and Menetometer (M) measurements in x,y,z directions
• "Groundtruth" data collected from CDIP buoy
• 7,000,000 observations!

Figure 1: Distribution of A1, G1, and M1 means according to wave height.

A1, G1, M1 STDs vs. Significant Wave Heights

Figure 2: Distribution of A1, G1, and M1 standard deviations according to wave height.
Course evaluations!

MGT495: [https://academicaffairs.ucsd.edu/Modules/Evals?e5551126](https://academicaffairs.ucsd.edu/Modules/Evals?e5551126)
CSE158: [https://cape.ucsd.edu/students/](https://cape.ucsd.edu/students/)
CSE258: [https://academicaffairs.ucsd.edu/Modules/Evals?e5421125](https://academicaffairs.ucsd.edu/Modules/Evals?e5421125)
Thanks!