CSE 258 – Lecture 15/16
Web Mining and Recommender Systems

Temporal data mining
Temporal models
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
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

\[f(data) \rightarrow labels\]
Here, we’d like to predict sequences of **real-valued** events as accurately as possible.
**Method 1**: maintain a “moving average” using a window of some fixed length

\[ f(x_1, \ldots, x_m) = \]
Method 1: maintain a “moving average” using a window of some fixed length

- This can be computed efficiently via dynamic programming:

\[ f(x_1, \ldots, x_{m+1}) = \]
Time-series regression

Also useful to plot data:

BeerAdvocate, ratings over time

Code on:
http://jmcauley.ucsd.edu/code/week10.py
Time-series regression

**Method 2:** weight the points in the moving average by age

\[ f(x_1, \ldots, x_m) = \]
Method 3: weight the most recent points exponentially higher

\[ f(x_1) = \]

\[ f(x_1, \ldots, x_m) = \]
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?

\[ f(x_1, \ldots, x_m) = \]
Time-series regression

**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
CSE 258 – Lecture 15/16
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
Temporal dynamics of social networks

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?

(from Leskovec, 2008 (CMU Thesis))

**A:** Doesn’t seem to be an obvious trend, so what do networks have in common as they evolve?
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 nodes are shortlived, with a long-tail of older nodes
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?

- 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.
Temporal dynamics of social networks

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

![Graphs showing temporal dynamics of social networks with citations and authorship](image)
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:

![Graph 1: Erdos-Renyi model (a = 1.3)](image1)

![Graph 2: Pref. attachment model (a = 1.2)](image2)
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
Temporal dynamics of social networks

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?

![Graphs showing effective diameter over time for different networks](c) Affiliation network (ATP-ASTRO-PH)  
(d) US patent citation network (CIT-PATENTS)
Temporal dynamics of social networks

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

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
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”)
Latent Dirichlet Allocation

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)
Latent Dirichlet Allocation

Topics and documents are both described using stochastic vectors:

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

$$\theta_d \in \Delta^K$$ 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

$$\phi_k \in \Delta^D$$ 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

e.g.

- 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
Latent Dirichlet Allocation

**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 Dir.(\( \beta \))
2. For each document \( d \), draw a topic vector \( \theta_d \) from 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”

\[
p.d.f.: \quad x^{\alpha-1} (1-x)^{\beta-1} \frac{1}{B(\alpha, \beta)}
\]
Latent Dirichlet Allocation

Results:
Political addresses – the model seems to capture realistic “bursty” and gradually emerging topics

<table>
<thead>
<tr>
<th>Mexican War</th>
<th>Panama Canal</th>
<th>Cold War</th>
<th>Modern Tech</th>
</tr>
</thead>
<tbody>
<tr>
<td>states 0.02032</td>
<td>government 0.02928</td>
<td>world 0.01875</td>
<td>energy 0.03902</td>
</tr>
<tr>
<td>mexico 0.01832</td>
<td>united 0.02132</td>
<td>states 0.01717</td>
<td>national 0.01534</td>
</tr>
<tr>
<td>government 0.01670</td>
<td>states 0.02067</td>
<td>security 0.01710</td>
<td>development 0.01448</td>
</tr>
<tr>
<td>united 0.01521</td>
<td>islands 0.01167</td>
<td>soviet 0.01664</td>
<td>space 0.01436</td>
</tr>
<tr>
<td>war 0.01059</td>
<td>canal 0.01014</td>
<td>united 0.01491</td>
<td>science 0.01227</td>
</tr>
<tr>
<td>congress 0.00951</td>
<td>american 0.00872</td>
<td>nuclear 0.01454</td>
<td>technology 0.01227</td>
</tr>
<tr>
<td>country 0.00906</td>
<td>cuba 0.00834</td>
<td>peace 0.01408</td>
<td>oil 0.01178</td>
</tr>
<tr>
<td>texas 0.00852</td>
<td>made 0.00747</td>
<td>nations 0.01069</td>
<td>make 0.00994</td>
</tr>
<tr>
<td>made 0.00727</td>
<td>general 0.00731</td>
<td>international 0.01024</td>
<td>effort 0.00969</td>
</tr>
<tr>
<td>great 0.00611</td>
<td>war 0.00660</td>
<td>america 0.00987</td>
<td>administration 0.00957</td>
</tr>
</tbody>
</table>
Latent Dirichlet Allocation

Results:
e-mails & conference proceedings

<table>
<thead>
<tr>
<th>Faculty Recruiting</th>
<th>ART Paper</th>
<th>MALLETT</th>
<th>CVS Operations</th>
<th>Recurrent NN</th>
<th>Game Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs</td>
<td>0.03572</td>
<td>0.02113</td>
<td>0.05668</td>
<td>0.04473</td>
<td>0.05963</td>
</tr>
<tr>
<td>april</td>
<td>0.02724</td>
<td>0.01814</td>
<td>0.04212</td>
<td>0.04070</td>
<td>0.03765</td>
</tr>
<tr>
<td>faculty</td>
<td>0.02341</td>
<td>0.01601</td>
<td>0.04073</td>
<td>0.03828</td>
<td>0.03616</td>
</tr>
<tr>
<td>david</td>
<td>0.02012</td>
<td>0.01408</td>
<td>0.03085</td>
<td>0.03587</td>
<td>0.02462</td>
</tr>
<tr>
<td>lunch</td>
<td>0.01766</td>
<td>0.01366</td>
<td>0.02947</td>
<td>0.03083</td>
<td>0.02402</td>
</tr>
<tr>
<td>schedule</td>
<td>0.01656</td>
<td>0.01238</td>
<td>0.02479</td>
<td>0.02539</td>
<td>0.02057</td>
</tr>
<tr>
<td>candidate</td>
<td>0.01560</td>
<td>0.01238</td>
<td>0.02080</td>
<td>0.02519</td>
<td>0.01300</td>
</tr>
<tr>
<td>talk</td>
<td>0.01355</td>
<td>0.01152</td>
<td>0.01664</td>
<td>0.02317</td>
<td>0.01242</td>
</tr>
<tr>
<td>bruce</td>
<td>0.01273</td>
<td>0.01067</td>
<td>0.01421</td>
<td>0.02277</td>
<td>0.01154</td>
</tr>
<tr>
<td>visit</td>
<td>0.01232</td>
<td>0.00960</td>
<td>0.01352</td>
<td>0.02156</td>
<td>0.01013</td>
</tr>
<tr>
<td>state</td>
<td></td>
<td></td>
<td></td>
<td>0.05963</td>
<td>0.03765</td>
</tr>
<tr>
<td>recurrent</td>
<td></td>
<td></td>
<td></td>
<td>0.04070</td>
<td>0.03616</td>
</tr>
<tr>
<td>sequence</td>
<td></td>
<td></td>
<td></td>
<td>0.03828</td>
<td>0.02462</td>
</tr>
<tr>
<td>sequences</td>
<td></td>
<td></td>
<td></td>
<td>0.03587</td>
<td>0.02402</td>
</tr>
<tr>
<td>time</td>
<td></td>
<td></td>
<td></td>
<td>0.03083</td>
<td>0.02057</td>
</tr>
<tr>
<td>states</td>
<td></td>
<td></td>
<td></td>
<td>0.02519</td>
<td>0.01300</td>
</tr>
<tr>
<td>transition</td>
<td></td>
<td></td>
<td></td>
<td>0.02317</td>
<td>0.01346</td>
</tr>
<tr>
<td>finite</td>
<td></td>
<td></td>
<td></td>
<td>0.02277</td>
<td>0.01242</td>
</tr>
<tr>
<td>length</td>
<td></td>
<td></td>
<td></td>
<td>0.02156</td>
<td>0.01154</td>
</tr>
<tr>
<td>strings</td>
<td></td>
<td></td>
<td></td>
<td>0.05963</td>
<td>0.03765</td>
</tr>
<tr>
<td>game</td>
<td></td>
<td></td>
<td></td>
<td>0.04070</td>
<td>0.03616</td>
</tr>
<tr>
<td>strategy</td>
<td></td>
<td></td>
<td></td>
<td>0.03828</td>
<td>0.02462</td>
</tr>
<tr>
<td>play</td>
<td></td>
<td></td>
<td></td>
<td>0.03587</td>
<td>0.02402</td>
</tr>
<tr>
<td>games</td>
<td></td>
<td></td>
<td></td>
<td>0.03083</td>
<td>0.02057</td>
</tr>
<tr>
<td>player</td>
<td></td>
<td></td>
<td></td>
<td>0.02519</td>
<td>0.01300</td>
</tr>
<tr>
<td>agents</td>
<td></td>
<td></td>
<td></td>
<td>0.02317</td>
<td>0.01346</td>
</tr>
<tr>
<td>expert</td>
<td></td>
<td></td>
<td></td>
<td>0.02277</td>
<td>0.01242</td>
</tr>
<tr>
<td>opponents</td>
<td></td>
<td></td>
<td></td>
<td>0.02156</td>
<td>0.01154</td>
</tr>
<tr>
<td>nash</td>
<td></td>
<td></td>
<td></td>
<td>0.05963</td>
<td>0.03765</td>
</tr>
<tr>
<td>0.02850</td>
<td></td>
<td></td>
<td></td>
<td>0.02378</td>
<td>0.01490</td>
</tr>
<tr>
<td>0.01473</td>
<td></td>
<td></td>
<td></td>
<td>0.01451</td>
<td>0.01346</td>
</tr>
<tr>
<td>0.01281</td>
<td></td>
<td></td>
<td></td>
<td>0.01123</td>
<td>0.01088</td>
</tr>
<tr>
<td>0.00848</td>
<td></td>
<td></td>
<td></td>
<td>0.02378</td>
<td>0.01490</td>
</tr>
</tbody>
</table>
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
CSE 258 – Lecture 15/16
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"
- HP's (item) "properties"
- 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:

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

(e.g. using stochastic gradient descent)
Temporal latent-factor models

To build a reliable system (and to win the Netflix prize!) we need to account for **temporal dynamics**:

![Graph showing Netflix ratings over time](image1)

*Netflix ratings over time*

![Graph showing Netflix ratings by movie age](image2)

*Netflix ratings by movie age*

(Netflix changed their interface)

(People tend to give higher ratings to older movies)

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
Start with a simple model of drifting dynamics for users:

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

- \text{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:

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

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

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

Real data

Netflix ratings over time

Fitted model
Temporal latent-factor models

time-dependent user bias can then be defined as:

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

overall user bias

sign and scale for deviation term

• 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

\[ \beta_{u}^{(2)}(t) = \beta_{u} + \sum_{l=1}^{k_{u}} \frac{e^{-\gamma|t-t_{l}^{u}|} b_{t_{l}^{u}}}{\sum_{l=1}^{k_{u}} e^{-\gamma|t-t_{l}^{u}|}} \]

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

\[ \beta_{u}^{(2)}(t) = \beta_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma |t-t_l^u|} b_{tu}}{\sum_{l=1}^{k_u} e^{-\gamma |t-t_l^u|}} \]

- This is now a reasonably flexible model, but still only captures *gradual drift*, i.e., it can’t handle sudden changes (e.g. a user simply having a bad day)

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

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

- global offset
- gradual deviation (or splines)
- item bias
- gradual item bias drift
- user bias
- single-day dynamics
Temporal latent-factor models

Finally, we can add a time-dependent scaling factor:

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

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.
Temporal latent-factor models

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
Incredible assignments
Reddit Sarcasm

- "Self-annotated" dataset ("\s" tag)
- Data includes the comment, author, subreddit, parent comment, score, and label
- Data from 1/1/2009 to 12/31/2016
- ~1 million comments

Ambareesh Jayakumari, Farheen Ahluwalia
Kenta Asai, Marlon Gamez, Jonathan Kiger, Robert Koepp
Andy Ruan, Alex Mao, Thant Htoo Zaw
Reddit Sarcasm

Baseline (text only):

\[ f(\text{text}) \rightarrow \text{label} \]

\[ \text{label} \approx \alpha + \sum_{w \in \text{text}} \text{count}(w) \cdot \theta_w \]

Predictive features:

- Contextual similarity to parent
- Sentiment analysis of sentence (off-the-shelf)
- Punctuation/length
- Accuracy of \(~70\%\) (compared to 61\% with text)

<table>
<thead>
<tr>
<th>TABLE I: Top Frequency of Sarcasm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subreddits</td>
</tr>
<tr>
<td>creepsyPMs</td>
</tr>
<tr>
<td>rage</td>
</tr>
<tr>
<td>MensRights</td>
</tr>
<tr>
<td>ShitRedditSays</td>
</tr>
<tr>
<td>Libertarian</td>
</tr>
<tr>
<td>Bad_Cop_No_Donut</td>
</tr>
<tr>
<td>worldnews</td>
</tr>
<tr>
<td>facepalm</td>
</tr>
<tr>
<td>Conservative</td>
</tr>
<tr>
<td>fatlogic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II: Low Frequency of Sarcasm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subreddits</td>
</tr>
<tr>
<td>gameofthrones</td>
</tr>
<tr>
<td>Jokes</td>
</tr>
<tr>
<td>BlackPeopleTwitter</td>
</tr>
<tr>
<td>science</td>
</tr>
<tr>
<td>fantasyfootball</td>
</tr>
<tr>
<td>skyrim</td>
</tr>
<tr>
<td>fo4</td>
</tr>
<tr>
<td>aww</td>
</tr>
<tr>
<td>4chan</td>
</tr>
<tr>
<td>Games</td>
</tr>
</tbody>
</table>
Outcomes of "Player Unknown's Battlegrounds" (PUBG) matches

• Each game is a survival/deathmatch between 100 users
• Each player ends the match with a ranking from 1 to 100
• The goal is to predict player's rankings from features of the player/match

Data:
• Records from 65,000 games (Kaggle dataset) - ~4.5M training records and ~2M test records

Features:
• Features include walking distance, number of kills, #weapons acquired, swimming distance, total damage dealt, match duration (etc.)
PUBG Match Outcomes

Feature correlations:

Target variable: winPlace Perc

MAE = 0.0201 - top 25 on leaderboard!
## Clothing fit

<table>
<thead>
<tr>
<th>Analysis/Dataset</th>
<th>RentTheRunWay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Transactions</td>
<td>192544</td>
</tr>
<tr>
<td>Number of Users</td>
<td>105571</td>
</tr>
<tr>
<td>Number of Items</td>
<td>5850</td>
</tr>
<tr>
<td>Number of Categories</td>
<td>68</td>
</tr>
<tr>
<td>Average Rating</td>
<td>9.09237</td>
</tr>
<tr>
<td>User with 1 Transaction</td>
<td>71824</td>
</tr>
<tr>
<td>Item with 1 Transaction</td>
<td>341</td>
</tr>
</tbody>
</table>

### Feature Description

<table>
<thead>
<tr>
<th>Feature Description</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>user’s height in cm</td>
<td>0</td>
</tr>
<tr>
<td>user’s weight in lb</td>
<td>1</td>
</tr>
<tr>
<td>item’s size</td>
<td>2</td>
</tr>
<tr>
<td>category (one-hot encoding)</td>
<td>3-70</td>
</tr>
<tr>
<td>body type (one-hot encoding)</td>
<td>71-77</td>
</tr>
</tbody>
</table>

---

Dingmei Gu, Junyu Lai, Yingzhen Qu
Jinrong Gong, Oliver Noss
Chenglong Yang, Chen Zhang
Eddie Tseng, Hsiao-Chen Huang
Clothing fit

\[ f(s, t) \text{ could be (for e.g.) a latent factor model indicating the user's true size and the item's true size} \]

\[
L(y_{ij}, f_w(s_i, t_j)) = \begin{cases} 
\text{if } y_{ij} \text{ is small:} & \max\{0, 1 - f_w(s_i, t_j) + b_2\} \\
\text{if } n \text{ is fit:} & \max\{0, 1 + f_w(s_i, t_j) - b_2\} + \\
\text{if } n \text{ is large:} & \max\{0, 1 - f_w(s_i, t_j) + b_1\} \\
\text{if } \max\{0, 1 - f_w(s_i, t_j) - b_1\} 
\end{cases}
\]

Acc:

<table>
<thead>
<tr>
<th></th>
<th>Ave</th>
<th>Small</th>
<th>Fit</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.3208</td>
<td>0.0329</td>
<td>0.1250</td>
<td>0.8045</td>
</tr>
<tr>
<td>Latent Factor</td>
<td>0.6049</td>
<td>0.6316</td>
<td>0.5536</td>
<td>0.6297</td>
</tr>
</tbody>
</table>
Steam video game data

- 10,947 games
- 87,626 users
- 5,153,209 purchases

Mean average percent error:

\[ M = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| , \]

\((A_t = \text{actual}, F_t = \text{predicted})\)

Best test error around 5% with an SVM classifier

Features:
- How long does this user typically play games?
- How long do people typically spend on this game?
- Rating, review text, bundle containment, etc.

Sashaank Pasumarthi, Saksham Beotra
Kai Li, Beier Li
Bike sharing

Chicago bikesharing data
- 776,246 trips
- Features include time, subscription status, gender, day of week, weather, temperature, location (and various distance metrics)
- Predict trip duration

<table>
<thead>
<tr>
<th>RMSE/Models</th>
<th>XGBoost</th>
<th>Random Forest</th>
<th>Ridge Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>3.8776</td>
<td>4.1126</td>
<td>4.8105</td>
</tr>
<tr>
<td>All Features</td>
<td>3.188</td>
<td>3.2623</td>
<td>3.584</td>
</tr>
<tr>
<td>No Haversine</td>
<td>3.2538</td>
<td>3.3120</td>
<td>3.6512</td>
</tr>
<tr>
<td>No Manhattan</td>
<td>3.2046</td>
<td>3.2731</td>
<td>3.5958</td>
</tr>
<tr>
<td>No hours</td>
<td>3.2702</td>
<td>3.3058</td>
<td>3.5984</td>
</tr>
<tr>
<td>No cluster</td>
<td>3.2286</td>
<td>3.3108</td>
<td>3.5923</td>
</tr>
<tr>
<td>No weekday</td>
<td>3.2237</td>
<td>3.2912</td>
<td>3.6099</td>
</tr>
<tr>
<td>No direction</td>
<td>3.2221</td>
<td>3.2925</td>
<td>3.5922</td>
</tr>
</tbody>
</table>
Rating and match prediction for Speed Dating

(1) gender (1 for male, 0 for female)
(2) difference of age
(3) same race or not
(4) interest correlation
(5) how often do the user go on a date
(6) expected happiness gained from speed dating
(7) expected number of matches
(8) attractiveness of partner
(9) attractiveness by partner
(10) attractiveness of self
(11) sincerity of partner
(12) sincerity by partner
(13) sincerity of self
(14) intelligence of partner
(15) intelligence by partner
(16) intelligence of self
(17) funny or not of partner
(18) funny or not by partner
(19) funny or not of self
(20) ambition of partner
(21) ambition by partner
(22) ambition of self
(23) share interests of partner
(24) share interests by partner
(25) like (0-10 rating of partner)
(26) like_o (0-10 rating by partner)
(27) match (1 for match, 0 otherwise)

Stated preference (L) vs. Actual decision making (R)

- 8378 male-female pairs (~277 males, 275 females)
- Estimate an "overall rating" (0-10)
- Models include various alternatives of latent factor models
- Best MSE of around 1 (bias only around 3)
- Attractiveness and "fun" are key factors, moreso than race, dating order, on interests

Chun-Han Yao, Hao-Kun Wu, Yi-An Lai
NBA Shot prediction

Predicting shots from NBA games:
- 128,069 shot entries (Kaggle)
- ~66% accuracy

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>game_id</td>
<td>The ID of the game</td>
</tr>
<tr>
<td>matchup</td>
<td>The date and teams involved in the game</td>
</tr>
<tr>
<td>location</td>
<td>Home or away game for the current player</td>
</tr>
<tr>
<td>win/lose</td>
<td>Outcome of game</td>
</tr>
<tr>
<td>final_margin</td>
<td>Difference in final score</td>
</tr>
<tr>
<td>shot_number</td>
<td>The nth shot a player has taken this game</td>
</tr>
<tr>
<td>period</td>
<td>The quarter of basketball a shot is taken</td>
</tr>
<tr>
<td>game_clock</td>
<td>The current time elapsed in the game</td>
</tr>
<tr>
<td>shot_clock</td>
<td>The current time of the shot clock at the shot.</td>
</tr>
<tr>
<td></td>
<td>In the NBA, a shot clock is 24 seconds</td>
</tr>
<tr>
<td>dribbles</td>
<td># of dribbles prior to shot</td>
</tr>
<tr>
<td>touch_time</td>
<td># seconds player touches ball prior to shot</td>
</tr>
<tr>
<td>pts_type</td>
<td>Whether shot is a 2-point or 3-point shot</td>
</tr>
<tr>
<td>shot_result</td>
<td>String for make or miss</td>
</tr>
<tr>
<td>closest_defender</td>
<td>Name of closest defender</td>
</tr>
<tr>
<td>closest_defender_id</td>
<td>Id of closest defender</td>
</tr>
<tr>
<td>close_def_dist</td>
<td>Distance in feet of closest defender</td>
</tr>
<tr>
<td>fgm</td>
<td>Whether or not shot was made (0 or 1)</td>
</tr>
<tr>
<td>pts</td>
<td># points earned from shot (0, 2, or 3)</td>
</tr>
<tr>
<td>player_name</td>
<td>Name of player who shot the ball</td>
</tr>
<tr>
<td>player_id</td>
<td>ID of player who shot the ball</td>
</tr>
</tbody>
</table>

Ryan Le, James Zeng
Eric Mugnier, Bartholomew Tam, Vu Dang
Course evaluations

• Please evaluate the course on

https://academicaffairs.ucsd.edu/Modules/Evals/Evaluate.aspx?id=1901486 (CSE 258)
https://academicaffairs.ucsd.edu/Modules/Evals/Evaluate.aspx?id=1937272 (MGMT 495)
Thanks!