Web Mining and Recommender Systems

Temporal data mining: Regression for Sequence Data
Learning Goals

• Discuss how to use regression to predict temporally evolving data
Temporal models

This topic will 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
Previously – 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 \textbf{real-valued} events as accurately as possible.
Here, we’d like to predict sequences of **real-valued** events as accurately as possible.

Given: a time series:

\[(x_1, \ldots, x_N) \in \mathbb{R}^N\]

Suppose we’d like to minimize the MSE (as usual!) of the final part of some continuous portion of the sequence

\[
\frac{1}{u-v+1} \sum_{t=u}^{v} (f_t(x_1, \ldots, x_{u-1}) - x_t)^2
\]
Time-series regression

**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:

\[ f(x_1, \ldots, x_{m+1}) = \]
**Method 1:** maintain a “moving average” using a window of some fixed length

\[ f(x_1, \ldots, x_m) = \frac{1}{K} \sum_{k=0}^{K-1} x_{m-k} \]

- This can be computed efficiently via dynamic programming:

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

“peel-off” the oldest point  
add the newest point
Time-series regression

Also useful to plot data:

BeerAdvocate, ratings over time

Sliding window (K=10000)

long-term trends

seasonal effects

Code on course webpage
Method 2: weight the points in the moving average by age

\[ f(x_1, \ldots, x_m) = \]
Method 2: weight the points in the moving average by age

\[ f(x_1, \ldots, x_m) = \frac{\sum_{k=0}^{K-1} (K-k)x_{m-k}}{\binom{K}{2}} \]

newest points have the highest weight
weight decays to zero after K points
**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
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?

\[ f(x_1, \ldots, x_m) = \]
**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
• Introduced several schemes to predict values in sequences
• Introduced autoregression
Web Mining and Recommender Systems

Temporal dynamics in social networks
Learning Goals

• Discuss how social networks change over time
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

Previously 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?

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

(From Leskovec, 2008 (CMU Thesis))
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. Different behavior on LinkedIn?
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.
Temporal dynamics of social networks

**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
Temporal dynamics of social networks

**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:

- Erdoes-Renyi – $a = 1.3$
- Pref. attachment model – $a = 1.2$
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?
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):
Learning Outcomes

• Discussed how social networks change over time
• Described some mechanisms to explain this phenomenon
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
Learning Goals

• Discuss how text can change over time
Bag-of-Words representations of text:

F_text = [150, 0, 0, 0, 0, 0, 0, ..., 0]

a     aardvark     zoetrope
Previously, 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 Schlinky “Schlinky” (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

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
- With in 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 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($\psi_{z_{di}}$)
**Topics over Time** (Wang & McCallum, 2006) is an approach to incorporate temporal information into topic models.

3.3. **draw a timestamp** \( t_{di} \) **from 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”

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

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

Results:
e-mails & conference proceedings
Latent Dirichlet Allocation

Results:
conference proceedings (NIPS)

Relative weights of various topics in 17 years of NIPS proceedings
• Discussed how text can change over time
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
Learning Goals

• Discuss how temporal dynamics can be incorporated into 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”
- Is the movie action-heavy?
- Are the special effects good?

**Compatibility**
Previously...

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:

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

(error) (regularizer)

(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**: 

(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)
To start with, let’s just assume that it’s only the \textbf{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 \textit{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:

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

- \text{mean rating date} for user \( u \)
- \text{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)
- **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

Fitted model

Netflix ratings over time
Temporal latent-factor models

time-dependent user bias can then be defined as:

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

- 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

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

- 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

$$\beta_u^{(2)}(t) = \beta_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma |t-t_l^u|} b_{tl}^u}{\sum_{l=1}^{k_u} e^{-\gamma |t-t_l^u|}}$$

- 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)
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:

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

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

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

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

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

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

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
• Discussed how temporal dynamics can be incorporated into recommender systems
• Discussed how this was useful for Netflix in particular
Further reading: “Collaborative filtering with temporal dynamics”
Yehuda Koren, 2009