Introduction to Deep Learning

17. Sequence Models

STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

aws

Dependent Random Variables

Data

- So far …
  - Collect observation pairs $(x_i, y_i) \sim p(x, y)$ for training
  - Estimate $y \mid x \sim p(y \mid x)$ for unseen $x' \sim p(x)$
- Examples
  - Images & objects
  - Regression problem
  - House & house prices
- **The order of the data did not matter**

Recall - Interaction with Environment

- **Batch** (download a book)
  Observe training data (x_1,y_1) ... (x_l,y_l) then deploy
- **Online** (follow the class)
  Observe x, predict f(x), observe y (stock market, homework)
- **Active learning** (ask questions in class)
  Query y for x, improve model, pick new x
- **Bandits** (do well at homework)
  Pick arm, get reward, pick new arm (also with context)
- **Reinforcement Learning** (play chess, drive a car)
  Take action, environment responds, take new action

Recall - Stateful Systems

Recall - Training ≠ Testing

- **Generalization performance**
  (the empirical distribution lies)
- **Covariate shift**
  (the covariate distribution lies)
- **Logistic regression**
  (tools to fix shift)
- **Covariate shift correction**
- **Label shift**
  (the label distribution lies)
- **Nonstationary Environments**

$$p_{\text{emp}}(x, y) \neq p(x, y)$$

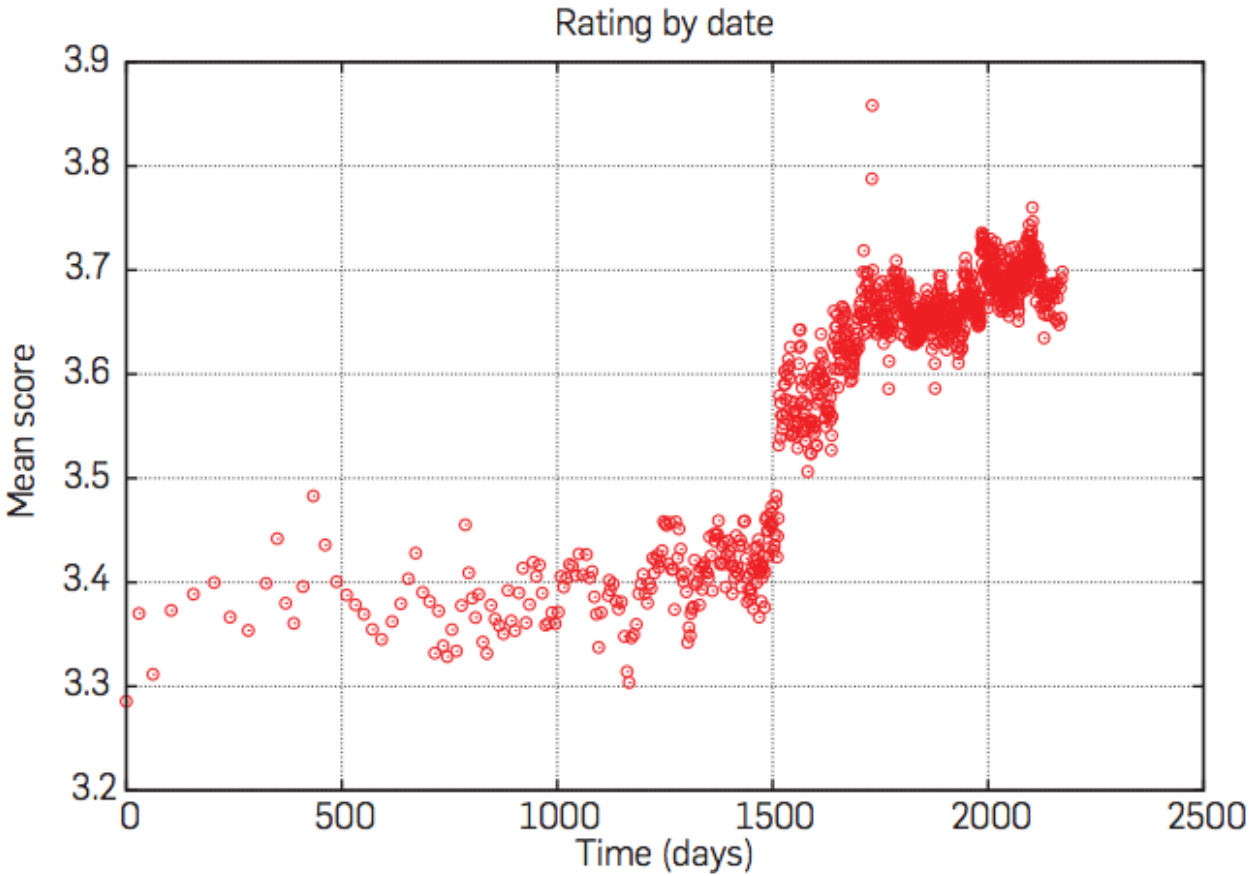
$$p(x) \neq q(x)$$

$$\log(1 + \exp(-yf(x)))$$

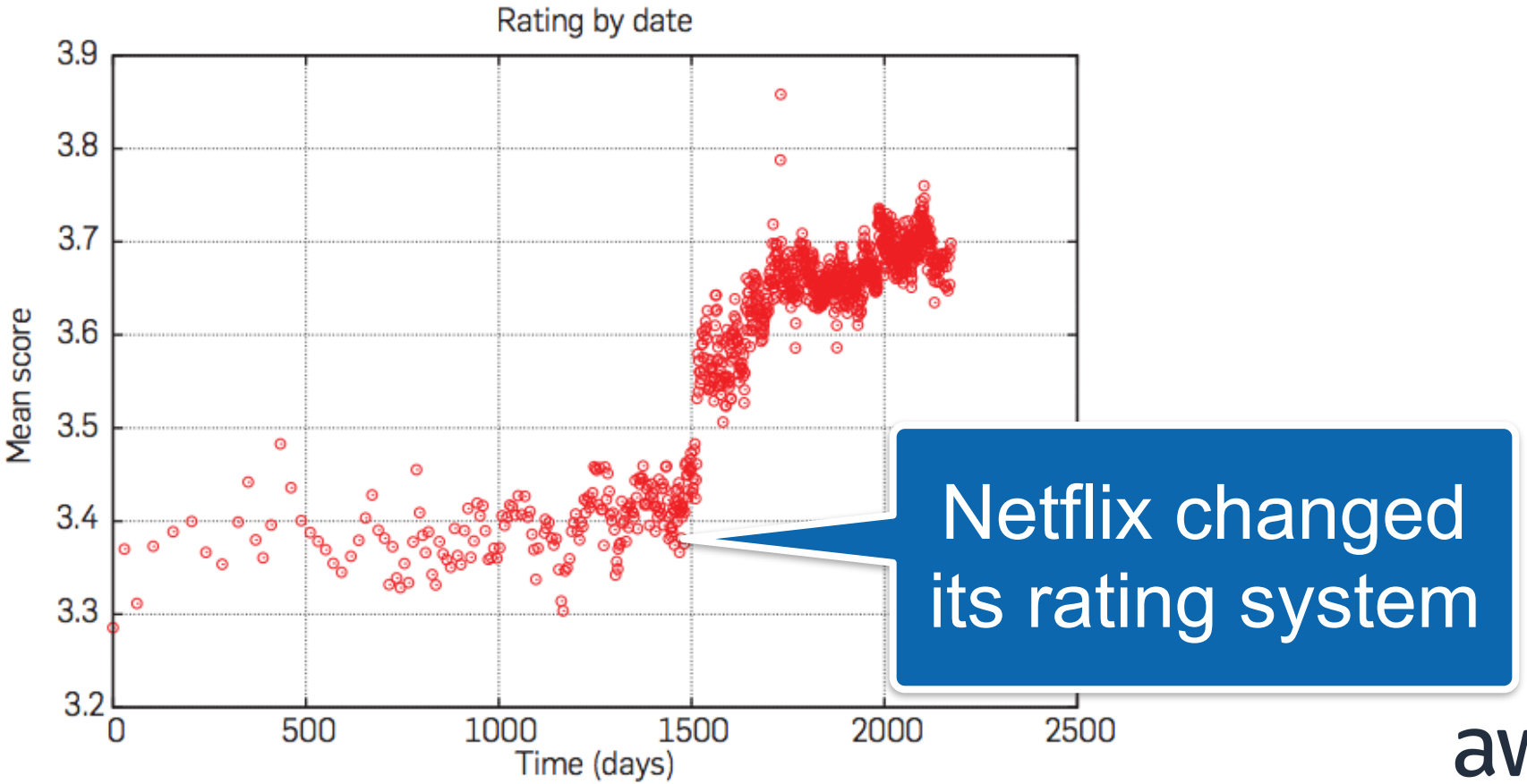
$$\frac{1}{2}\left(p(x)\delta(1, y) + q(x)\delta(-1, y)\right)$$

$$p(y) \neq q(y)$$

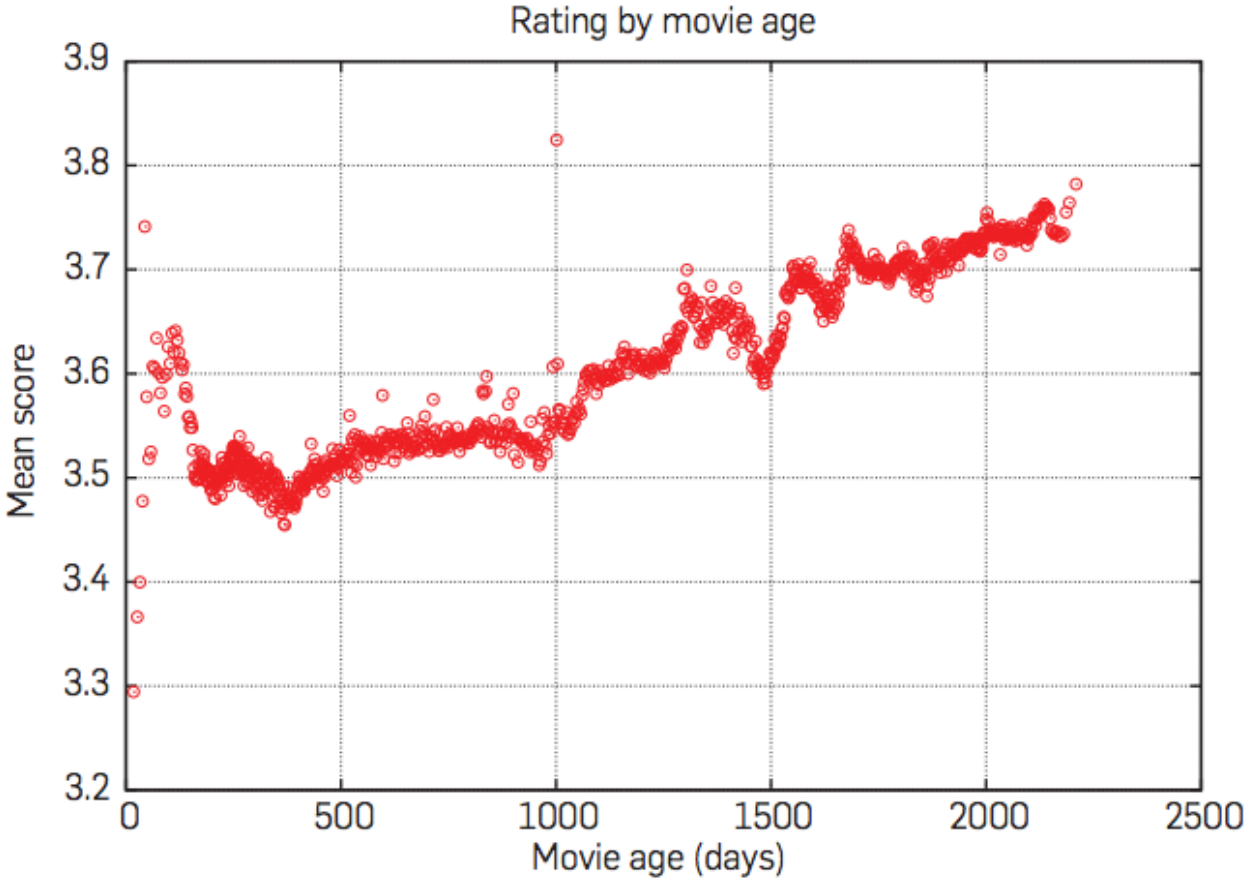
Time matters (Koren, 2009)



Time matters (Koren, 2009)



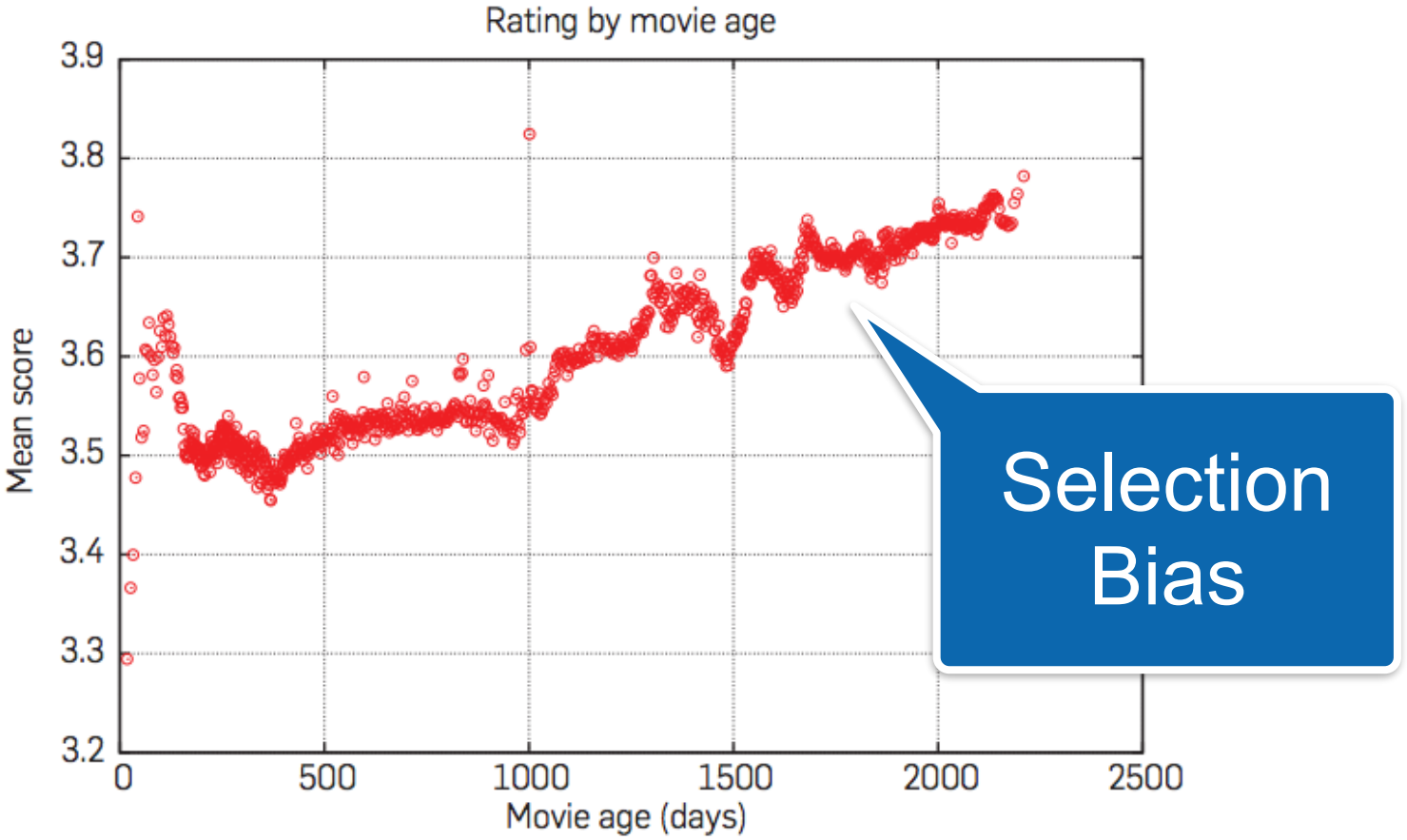
Time matters (Koren, 2009)



courses.d2l.ai/berkeley-stat-157

aws

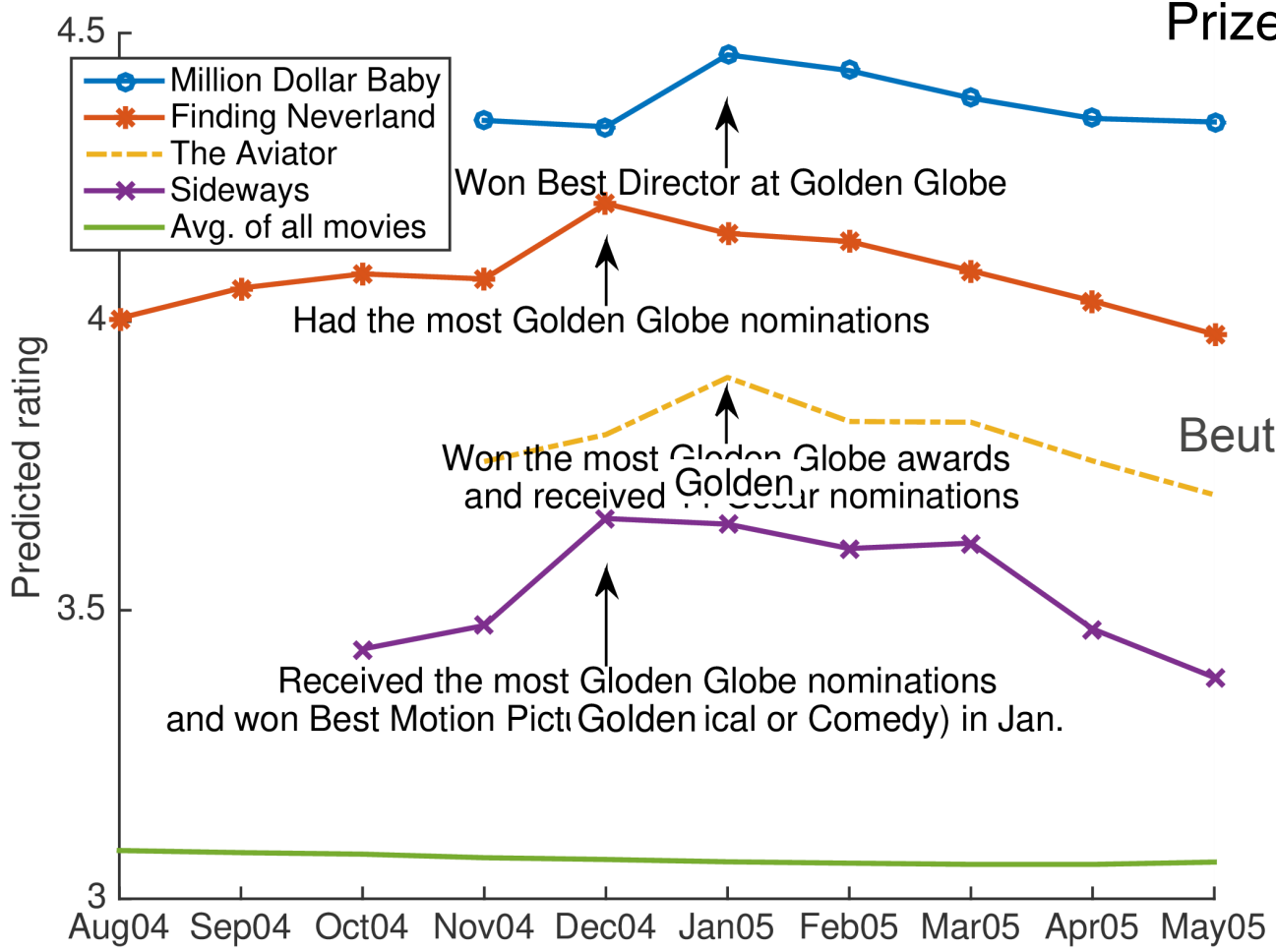
Time matters (Koren, 2009)



courses.d2l.ai/berkeley-stat-157

aws

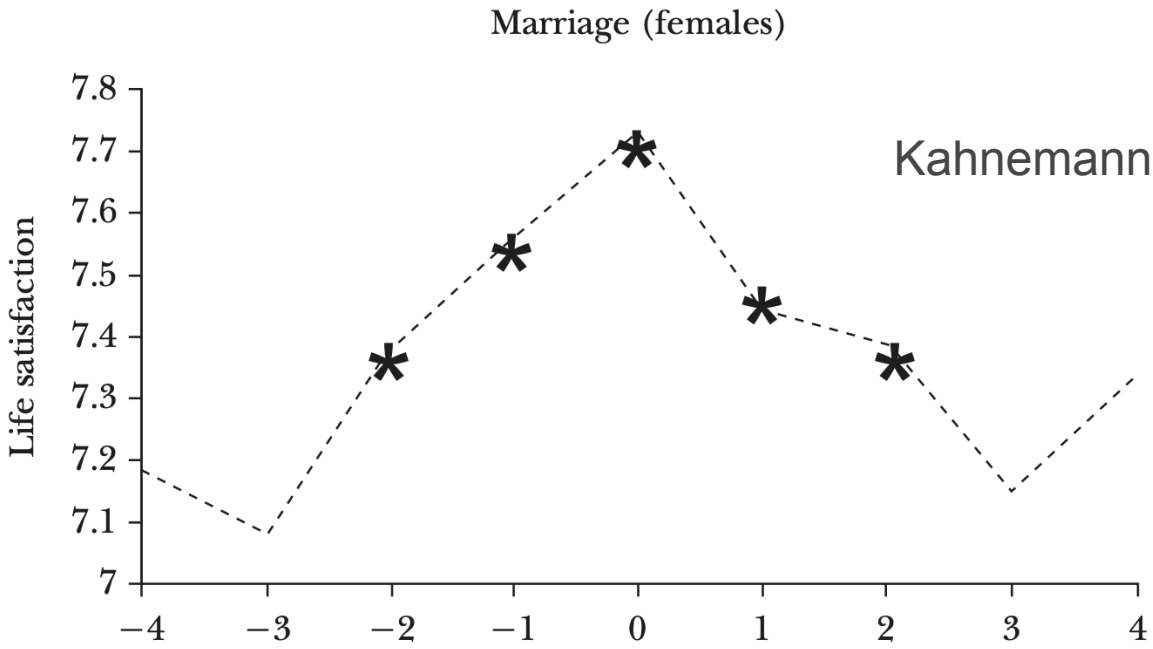
Prizes



Beutel et al., 2015

Average Life Satisfaction for a Sample of German Women

(by year of marriage $t = 0$)

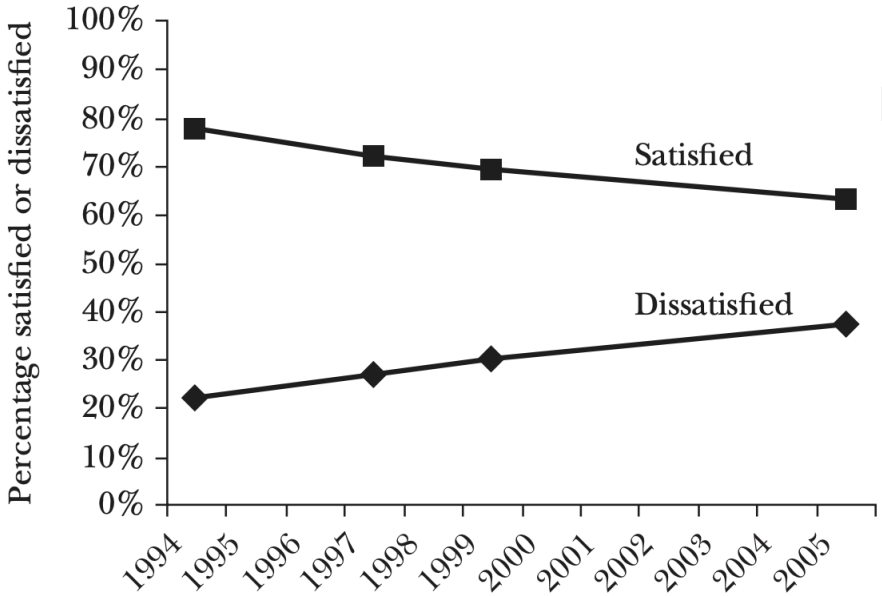


Kahnemann & Krueger, 2006

Source: Clark, Diener, Georgellis and Lucas (2003), using data from the German Socioeconomic Panel.
Note: An asterisk indicates that life satisfaction is significantly different from the baseline level.

Life Satisfaction in China as Average Real Income Rises by 250 Percent

Overall, how satisfied or dissatisfied are you with the way things are going in your life today? Would you say you are very satisfied, somewhat satisfied, somewhat dissatisfied, or very dissatisfied?

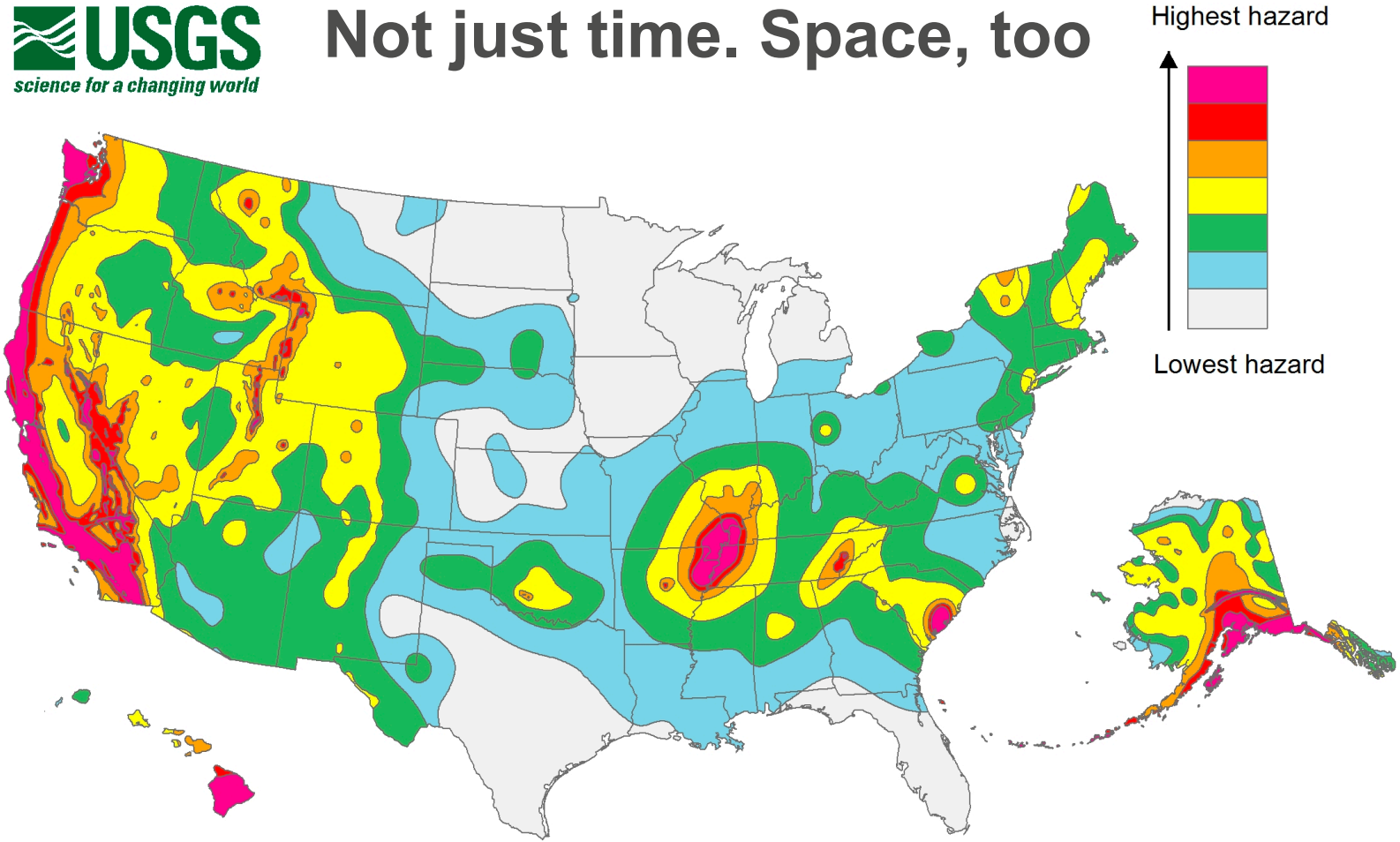


Kahnemann & Krueger, 2006

Source: Derived from Richard Burkholder, "Chinese Far Wealthier Than a Decade Ago—but Are They Happier?" The Gallup Organization, ⟨http://sww.gallup.com/poll/content/login.aspx?ci=14548⟩.

USGS
science for a changing world

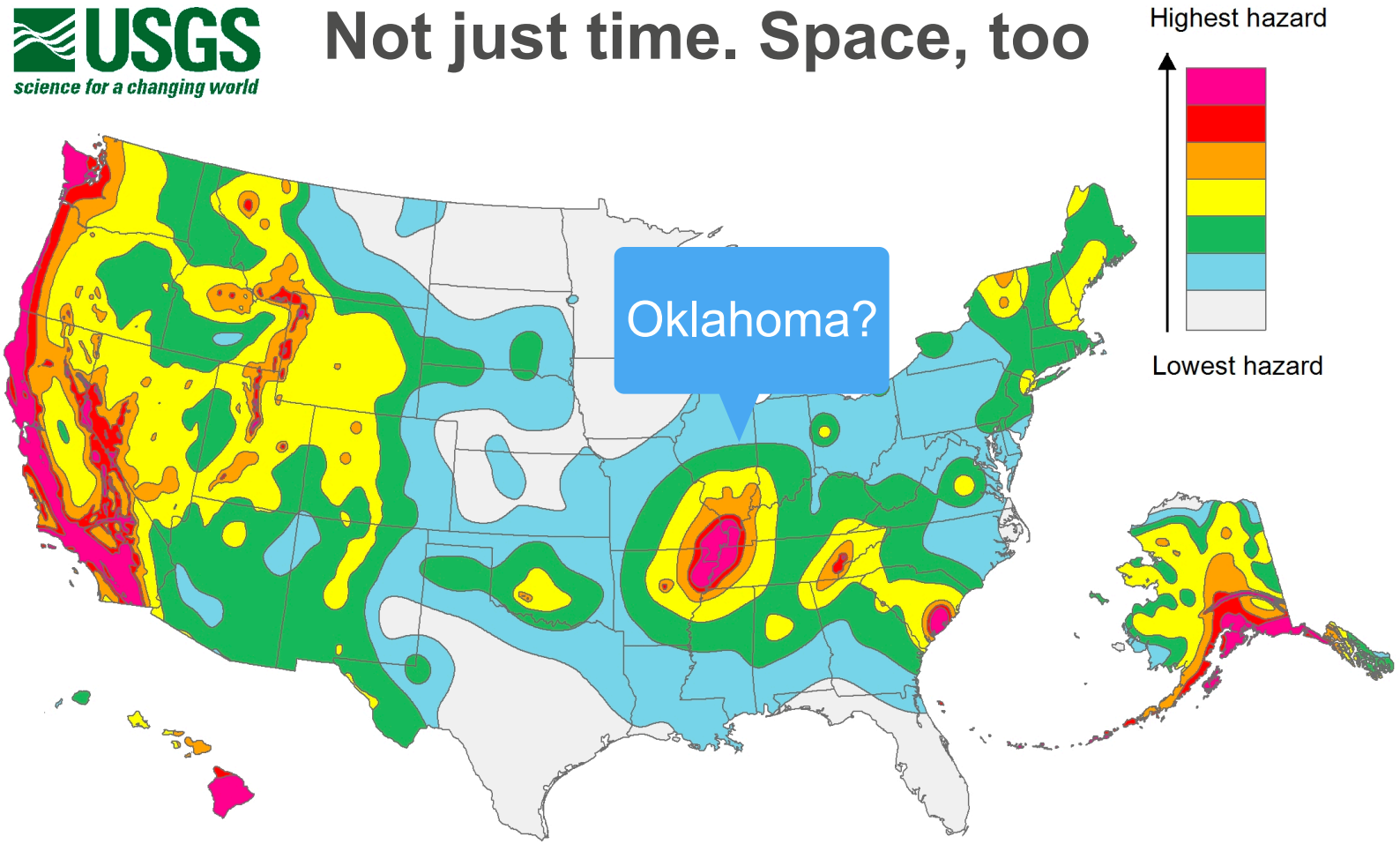
Not just time. Space, too



aws

USGS
science for a changing world

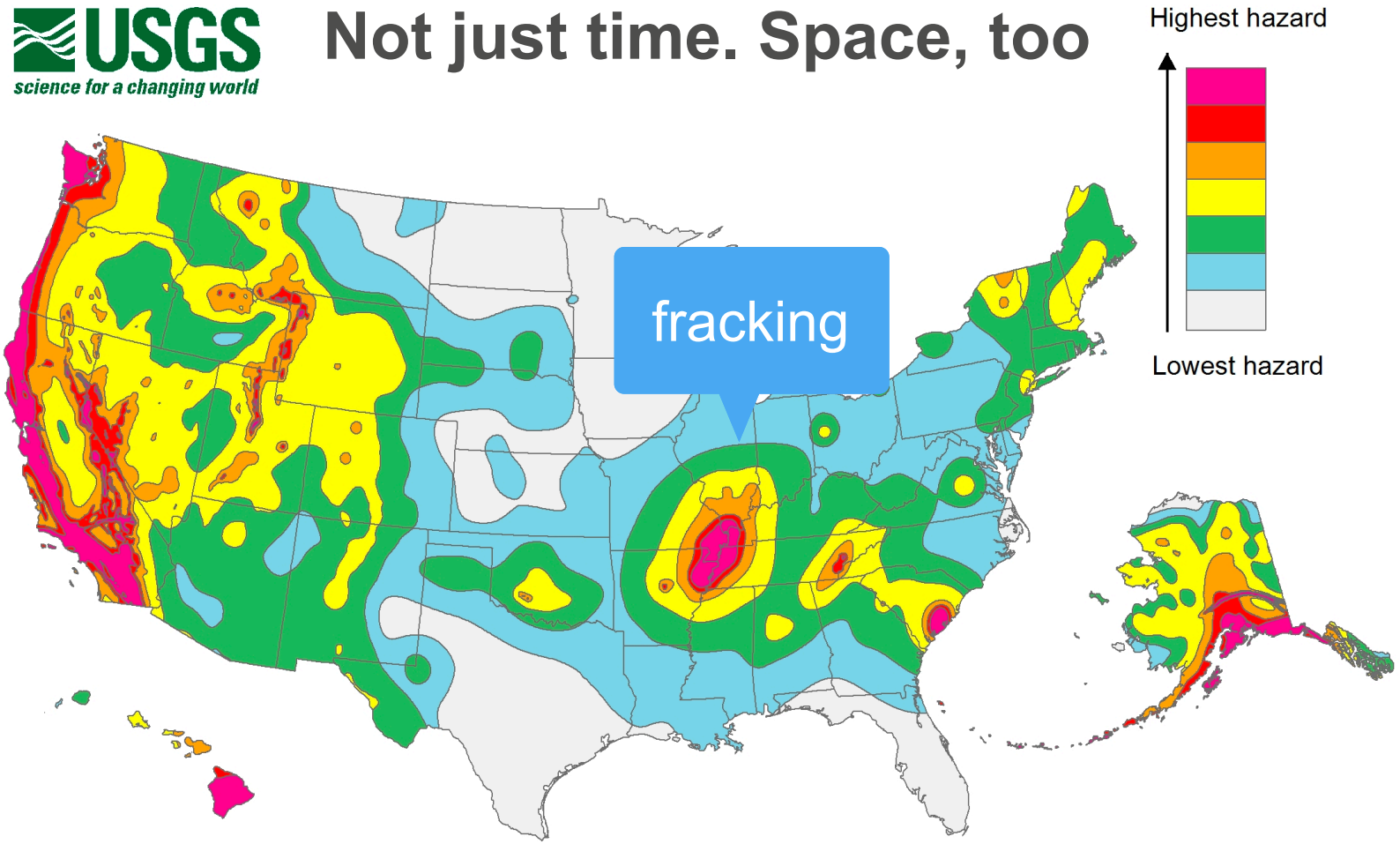
Not just time. Space, too



aws

USGS
science for a changing world

Not just time. Space, too



aws

FTSE 100

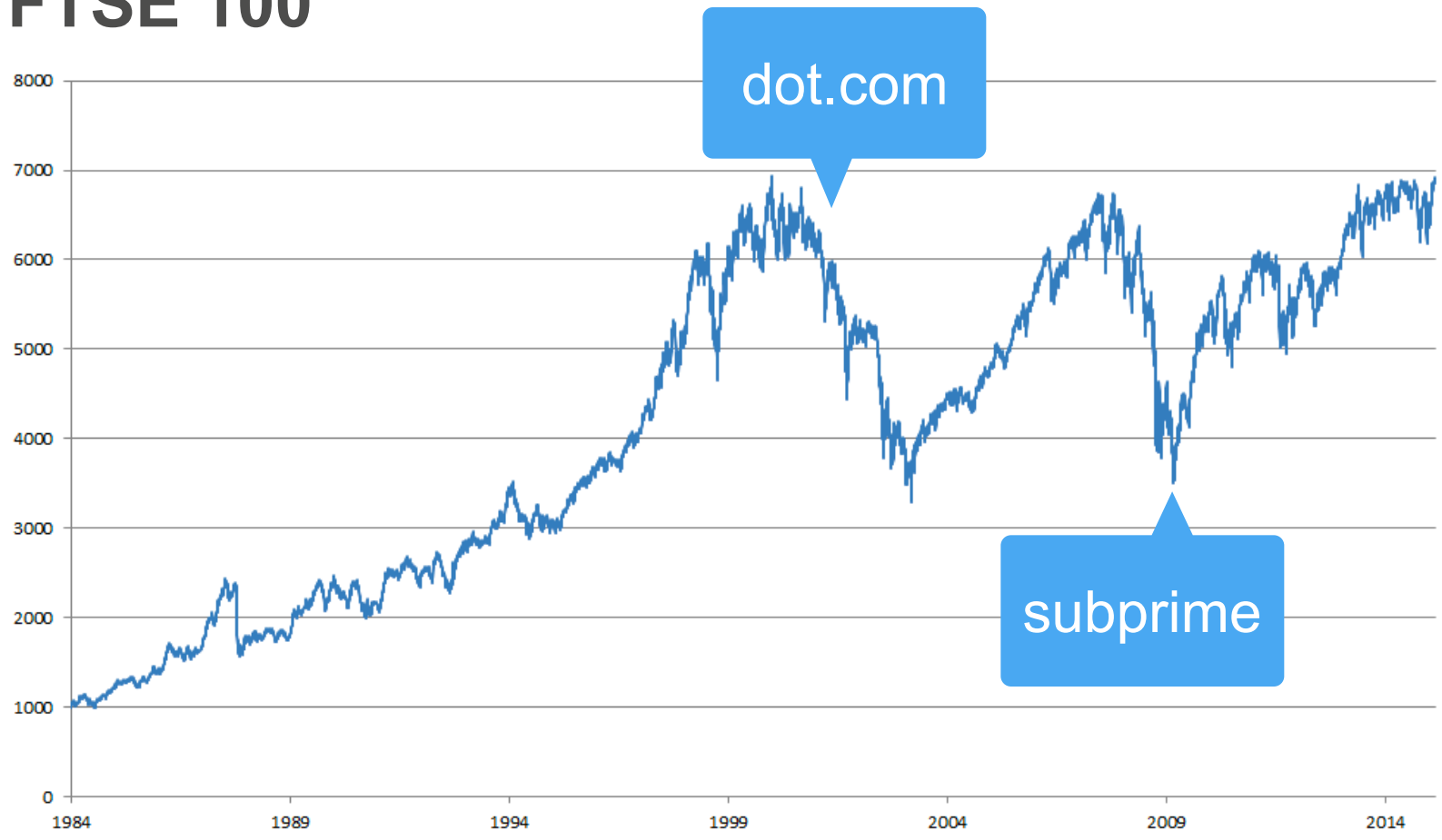

8000
7000
6000
5000
4000
3000
2000
1000
0
1984
1989
1994
1999
2004
2009
2014

FTSE 100

courses.d2l.ai/berkeley-stat-157

aws

FTSE 100


dot.com
subprime
8000
7000
6000
5000
4000
3000
2000
1000
0
1984
1989
1994
1999
2004
2009
2014

TL;DR - Data usually isn’t IID

Sequence Models

Sequence Model

- Dependent random variables

$$(x_1, \ldots x_T) \sim p(x)$$

- Conditional probability expansion

$$p(x) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_1, \ldots x_{T-1})$$

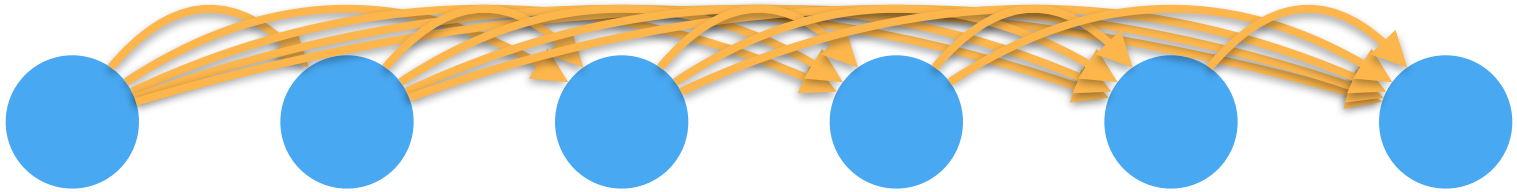
  - Can always find this expansion
  - Could also find reverse direction …

$$p(x) = p(x_T) \cdot p(x_{T-1} \mid x_T) \cdot p(x_{T-2} \mid x_{T-1}, x_T) \cdot \ldots p(x_1 \mid x_2, \ldots x_T)$$

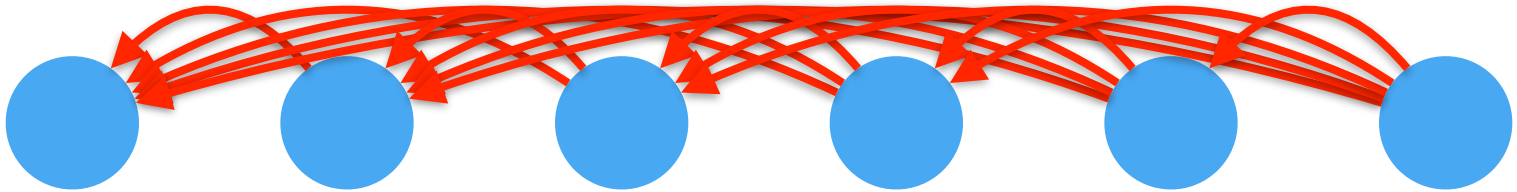
So why bother?

Sequence Model

$$p(x) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_1, \ldots x_{T-1})$$



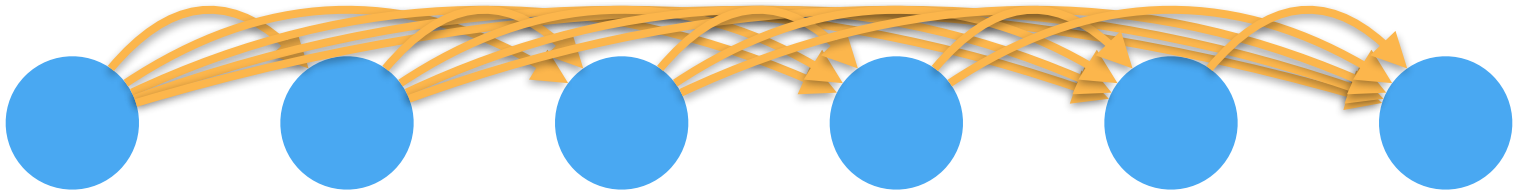
$$p(x) = p(x_T) \cdot p(x_{T-1} \mid x_T) \cdot p(x_{T-2} \mid x_{T-1}, x_T) \cdot \ldots p(x_1 \mid x_2, \ldots x_T)$$



- Causality (physics) prevents the reverse direction
- 'wrong' direction often much more complex to model

Sequence Model

$$p(x) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_1, \ldots x_{T-1})$$



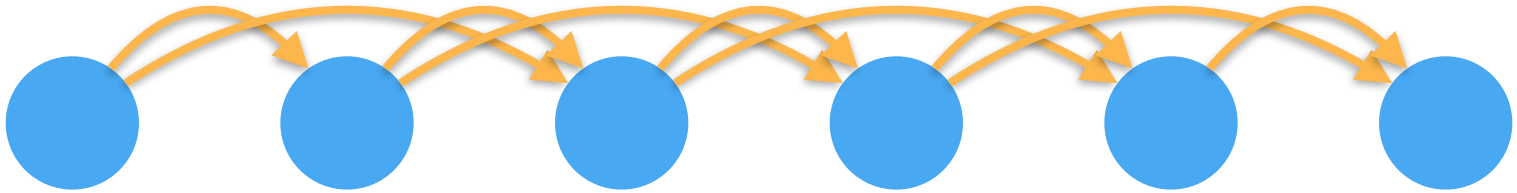
- Autoregressive model

$$p(x_t \mid x_1, \ldots x_{t-1}) = p(x_t \mid f(x_1, \ldots x_{t-1}))$$

Some function of previously seen data

Plan A - Markov Assumption

$$p(x) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_{T-\tau}, \ldots x_{T-1})$$



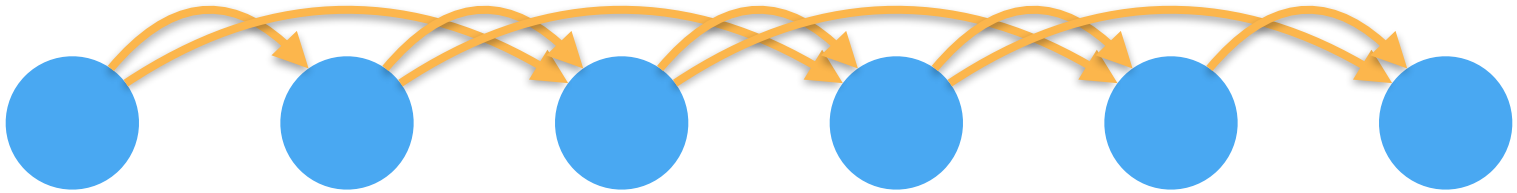
- Assume that only a few steps in the past matter
- Autoregressive model

$$p(x_t \mid x_1, \ldots x_{t-1}) = p(x_t \mid f(x_{t-\tau}, \ldots x_{t-1}))$$

Some function of previously seen data

Plan A - Markov Assumption

$$p(x) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_{T-\tau}, \ldots x_{T-1})$$



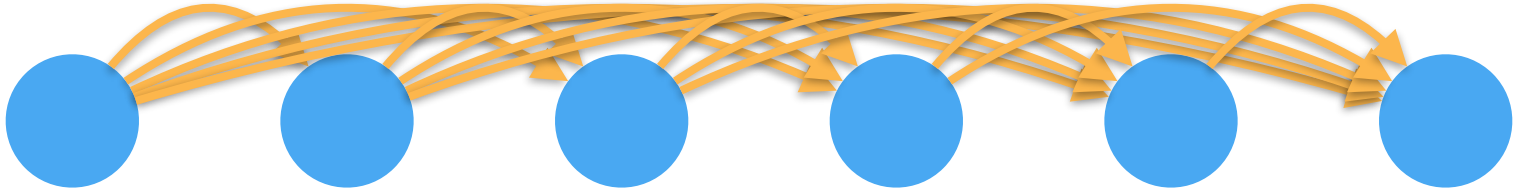
- In practice solve regression problem

$$\hat{x}_t = f(x_{t-\tau}, \ldots x_{t-1})$$

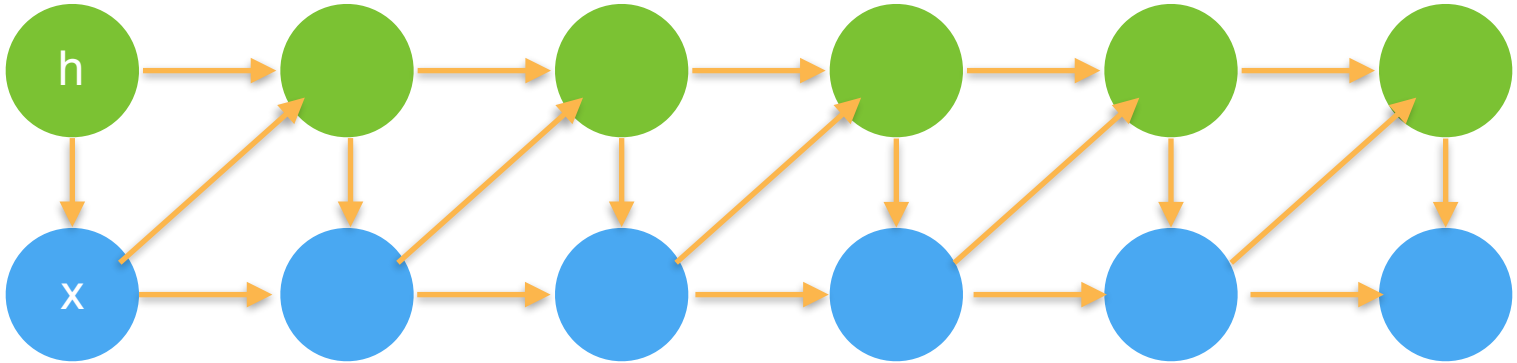
e.g. train an MLP on previously seen data

Plan B - Latent Variable Model

$$p(x) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_1, \ldots x_{T-1})$$



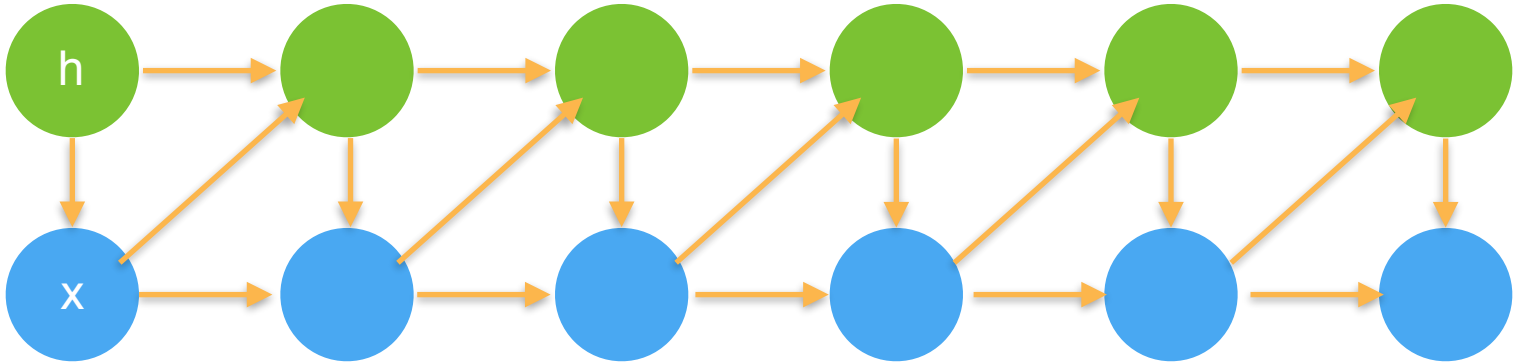
$p(h_t \mid h_{t-1}, x_{t-1})$ and $p(x_t \mid h_t, x_{t-1})$



Plan B - Latent Variable Model

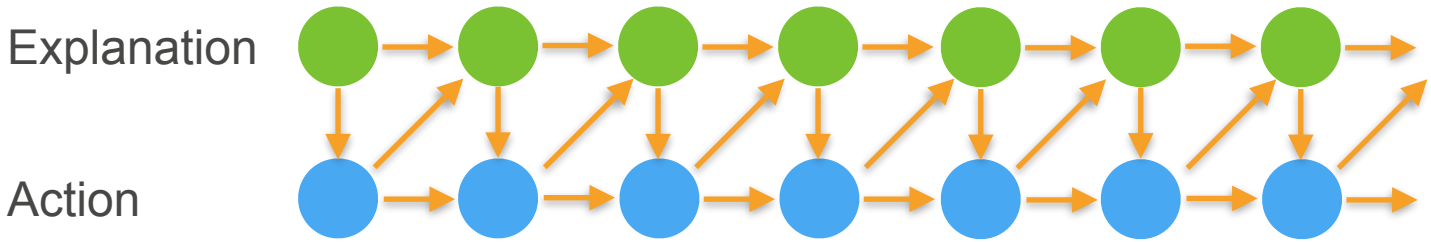
- Latent state summarizes all the relevant information about the past. So we get $h_t = f(x_1, \ldots x_{t-1}) = f(h_{t-1}, x_{t-1})$

$p(h_t \mid h_{t-1}, x_{t-1})$ and $p(x_t \mid h_t, x_{t-1})$



Latent Variable Models (classical treatment)

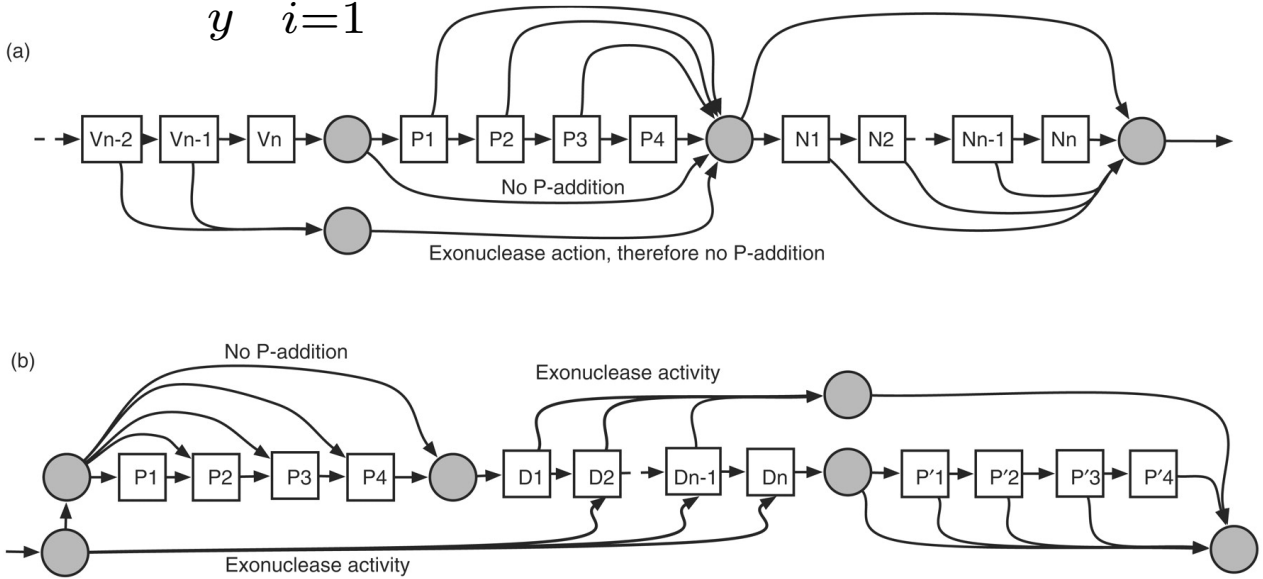
- **Temporal sequence of observations**
  Purchases, likes, app use, e-mails, ad clicks, queries, ratings
- **Latent state to explain behavior**
  - Clusters (navigational, informational queries in search)
  - Topics (interest distributions for users over time)
  - Kalman Filter (trajectory and location modeling)



Temporal Clustering aka Hidden Markov Models

- Clusters with sequential dependence

$$p(x) = \sum_{y} \prod_{i=1}^{n} p(y_i|y_{i-1})p(x_i|y_i)$$

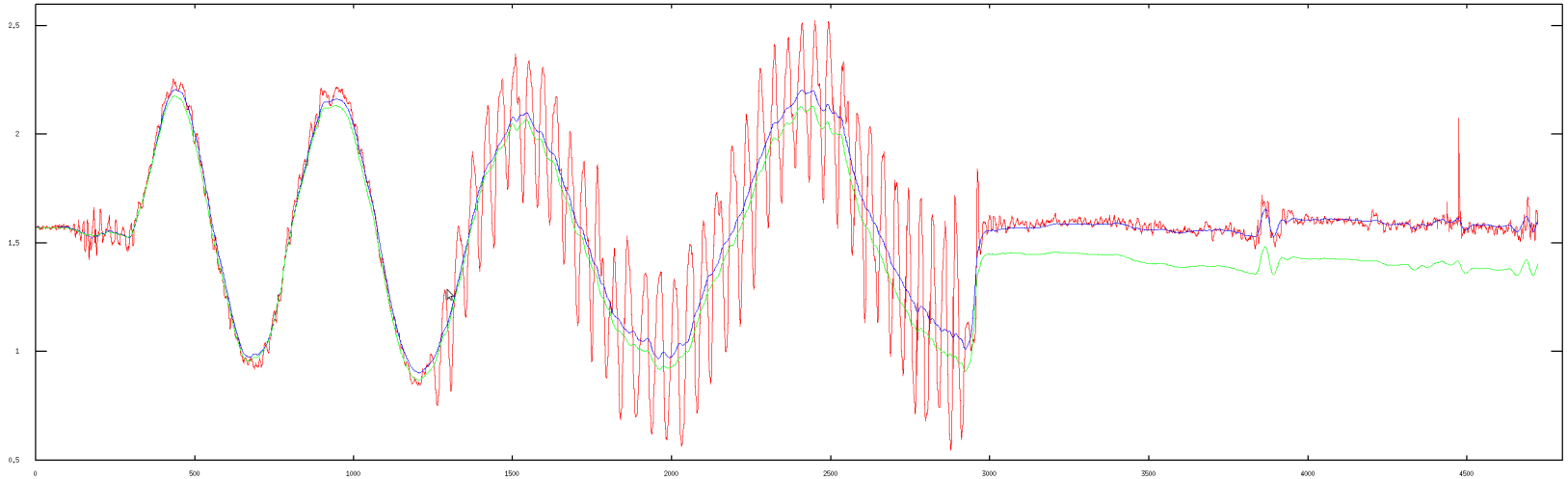


Temporal PCA aka Kalman Filter

- Latent factor variable
- Simple sequential factorial structure

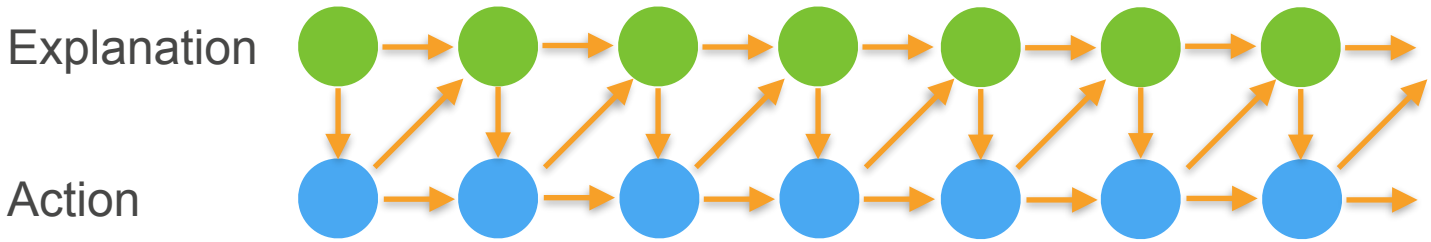
$$x_t \sim \mathcal{N}(Ay_t + \mu, K)$$

$$y_t \sim \mathcal{N}(By_{t-1} + \nu, L)$$



Sequence Models

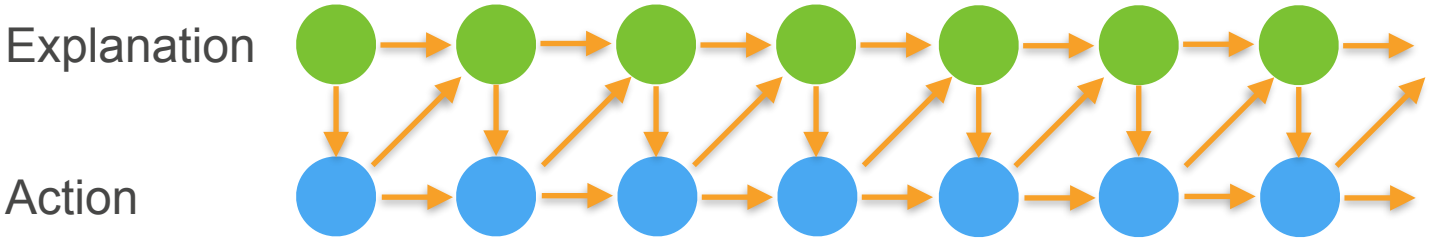
- **Temporal sequence of observations**
  Purchases, likes, app use, e-mails, ad clicks, queries, ratings
- **Latent state to explain behavior**
  - Clusters (navigational, informational queries in search)
  - Topics (interest distributions for users over time)
  - Kalman Filter (trajectory and location modeling)



Sequence Models

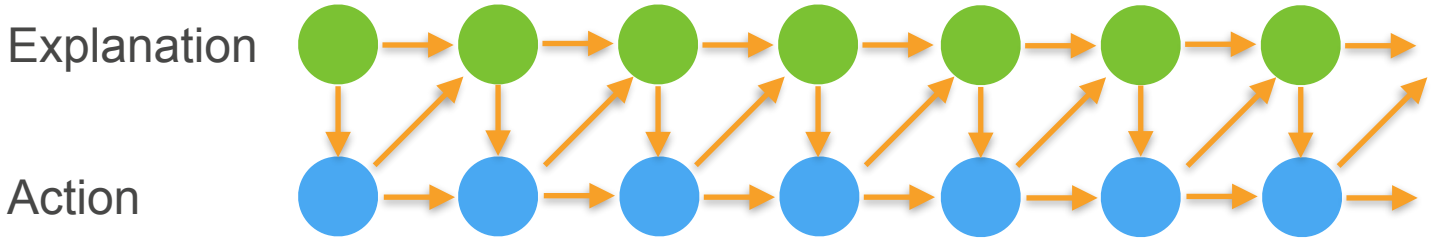
- **Temporal sequence of observations**
  Purchases, likes, app use, e-mails, ad clicks, queries, ratings
- **Latent state to explain behavior**

Are the parametric models really true?



Sequence Models

- **Temporal sequence of observations**
  Purchases, likes, app use, e-mails, ad clicks, queries, ratings
- **State space**
  Variable depth / variable representations / variable types
  (we know more about some users, queries than others)
- **Temporal resolution**
  Data doesn't arrive at quantized intervals



Plan A - Markov Assumption

Markov Assumption

- Next observation only depends on the past few terms

$$\hat{x}_t = f(x_{t-\tau}, \ldots x_{t-1})$$

- Train regression model
- Use it to predict the next step and iterate

Demo time

Language Models

Modeling Language 101

- Tokens not real values (domain is countably finite)

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_1, \ldots, w_{t-1})$$

$$\begin{aligned} & p(\text{Statistics}, \text{is}, \text{fun}, .) \\ = \; & p(\text{Statistics})\, p(\text{is} \mid \text{Statistics})\, p(\text{fun} \mid \text{Statistics}, \text{is})\, p(. \mid \text{Statistics}, \text{is}, \text{fun}) \end{aligned}$$

- Estimating it

$$\hat{p}(\text{is} \mid \text{Statistics}) = \frac{n(\text{Statistics is})}{n(\text{Statistics})}$$

N-grams (longer sequences of tokens)

- Need smoothing (long n-grams are infrequent)

$$\hat{p}(w) = \frac{n(w) + \epsilon_1/m}{n + \epsilon_1}$$

$$\hat{p}(w'|w) = \frac{n(w,w') + \epsilon_2\hat{p}(w')}{n(w) + \epsilon_2}$$

$$\hat{p}(w''|w',w) = \frac{n(w,w',w'') + \epsilon_3\hat{p}(w',w'')}{n(w,w') + \epsilon_3}$$

Hack

Talk to your friendly Bayesian if you want to do it right!

Let’s look at actual language statistics

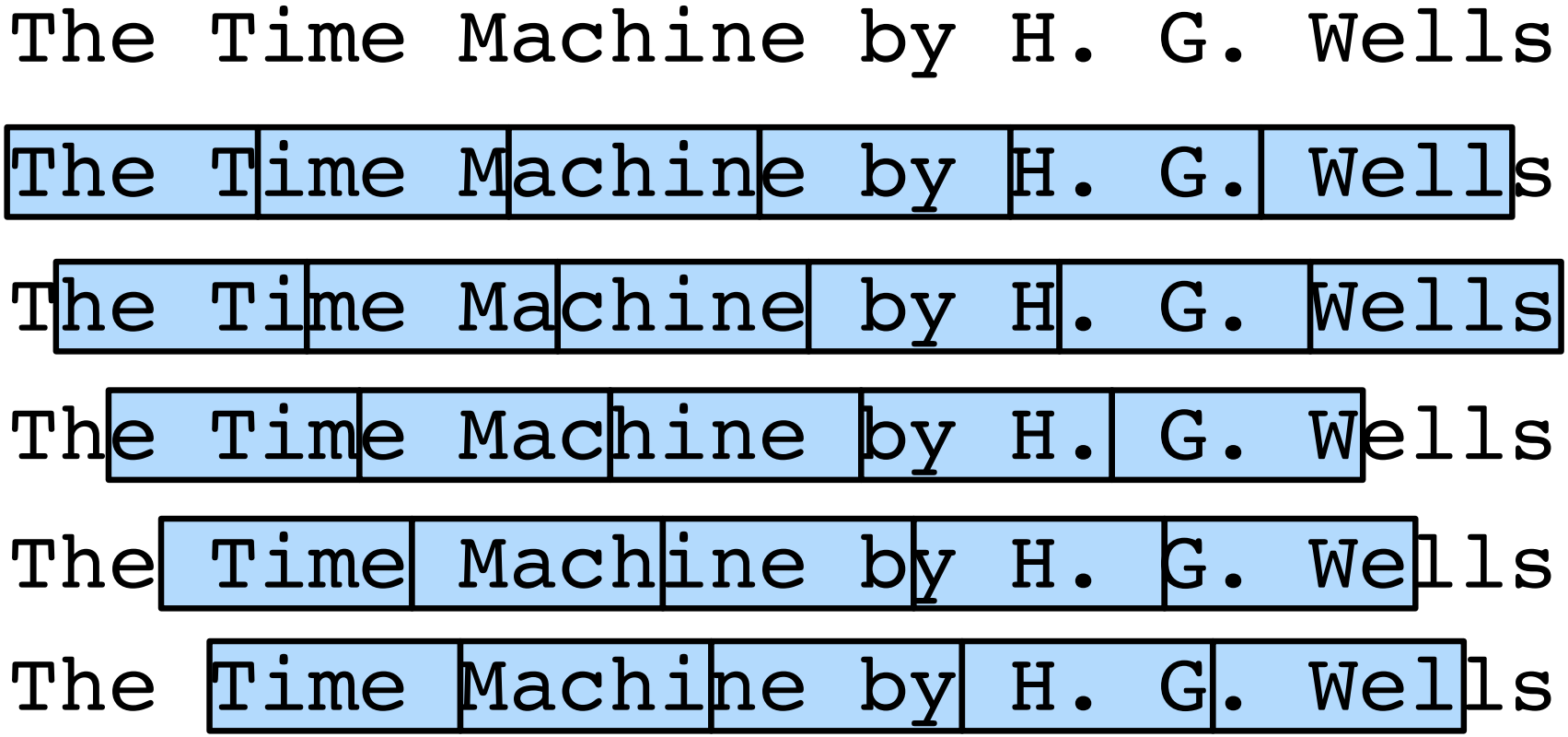
Text Preprocessing

Tokenization

`The Time Machine by H. G. Wells`

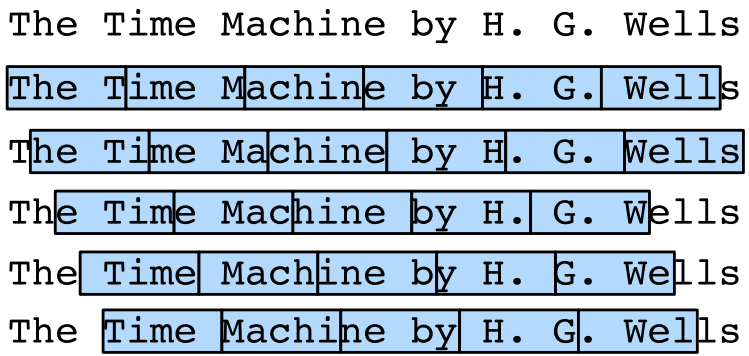
- Basic Idea - map text into sequence of IDs
- **Character Encoding** (each character has one ID)
  - Small vocabulary
  - Doesn’t work so well (DNN needs to learn spelling)
- **Word Encoding** (each word has one ID)
  - Accurate spelling
  - Doesn’t work so well (huge vocabulary = costly multinomial)
- **Byte Pair Encoding** (Goldilocks zone)
  - Frequent subsequences (like syllables)

Minibatch Generation



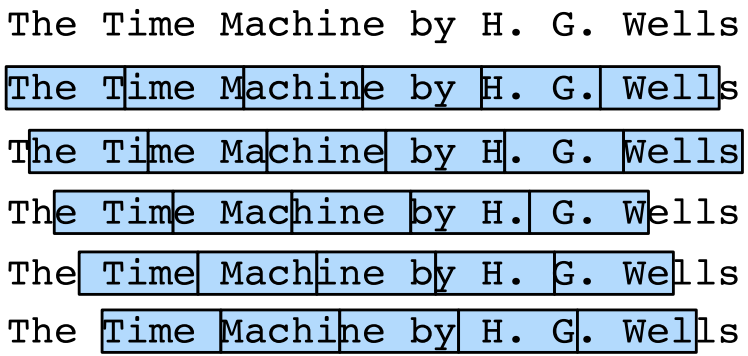
Minibatch Generation

- **Random partitioning**
  - Pick random offset
  - Distribute sequences at random over mini batches
  - Independent-ish samples
  - Need to reset hidden state



Minibatch Generation

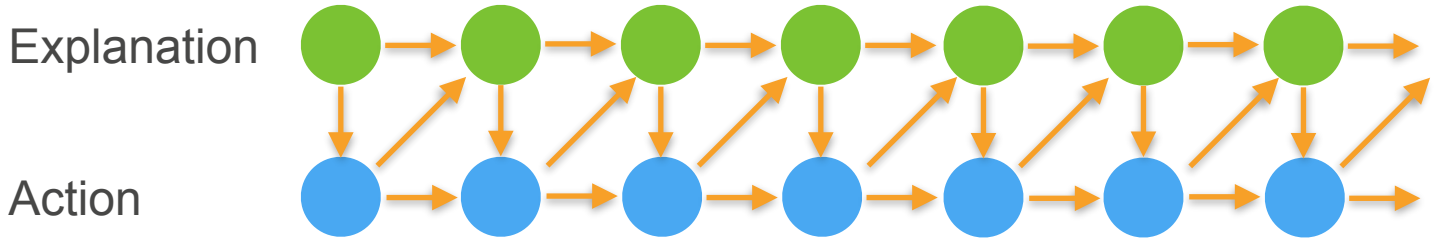
- **Sequential partitioning**
  - Pick random offset
  - Distribute sequences in sequence over mini batches
  - Dependent samples
  - Keep hidden state across mini batches (much better)



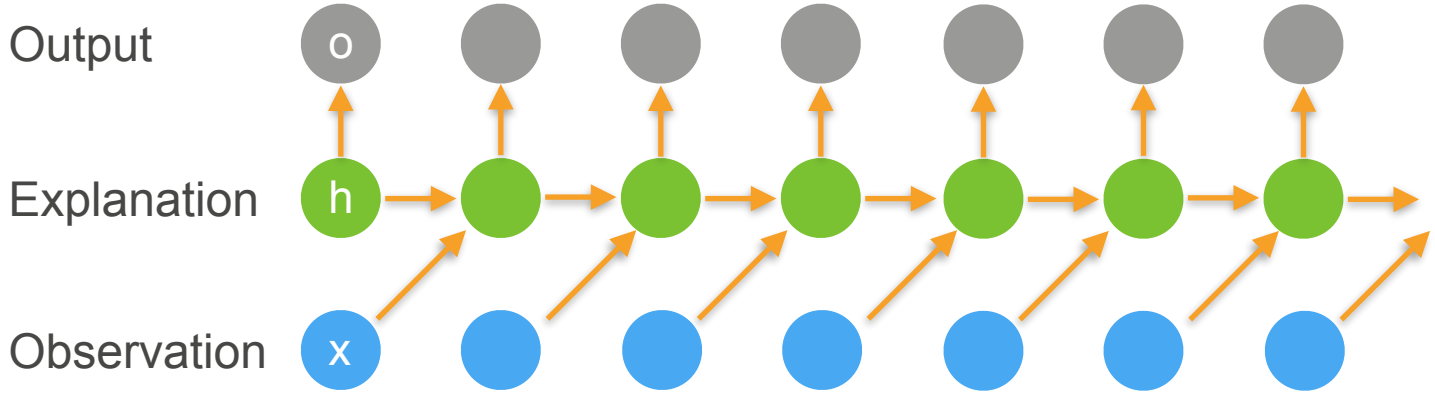
Code …

Plan B - Recurrent Neural Networks

Recurrent Neural Networks (with hidden state)



Recurrent Neural Networks (with hidden state)



- Hidden State update

$$\mathbf{h}_t = \phi(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{hx}\mathbf{x}_{t-1} + \mathbf{b}_h)$$

- Observation update

$$\mathbf{o}_t = \phi(\mathbf{W}_{ho}\mathbf{h}_t + \mathbf{b}_o)$$

Code …