FEATURE SPACES II

Prof. Alexander Huth

10/17/2017

REMINDER

* Homework 1 due TODAY! Please email it to huth@cs.utexas.edu by midnight

SYSTEM IDENTIFICATION

$$Y = f(X)$$

* What kind of a function is f?

SYSTEM IDENTIFICATION

* **Linearized model**

$$Y = \mathbb{L}(X)\beta$$

Let's invent some **L**'s

LINEARIZING TRANSFORMATIONS

* **L** should be a function that:
  * ingests stimuli
  * emits feature vectors

LINEARIZING TRANSFORMATIONS

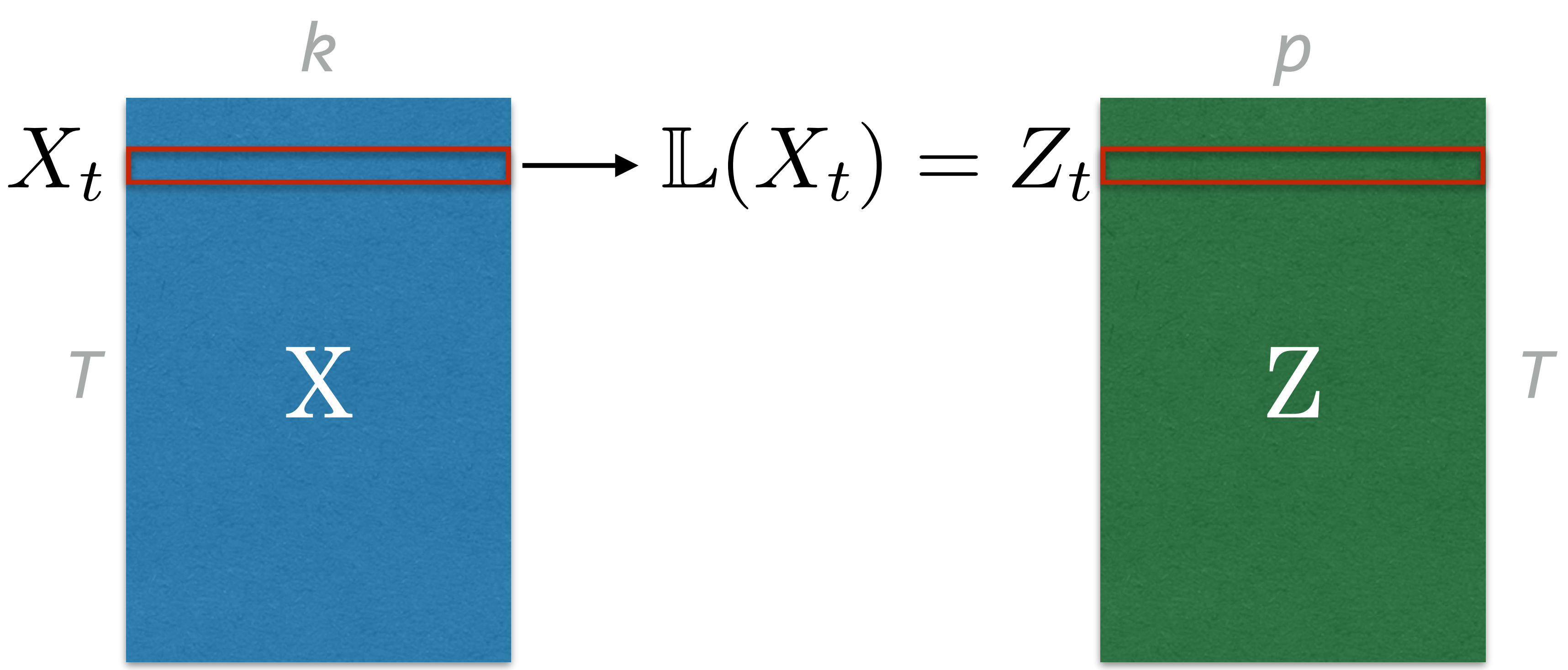
* Simplest version: time-invariant **L**

$$X_t \in \mathbb{R}^k$$ *k-dim vector*

$$\mathbb{L}(X_t) = Z_t \in \mathbb{R}^p$$ *p-dim vector*

LINEARIZING TRANSFORMATIONS

* Simplest version: time-invariant **L**



LINEARIZING TRANSFORMATIONS

* Simplest version: time-invariant **L**

* **Example:** X is an image with k pixels, Z is a binary vector saying which of p object categories are present in the image

LINEARIZING TRANSFORMATIONS

* Simplest version: time-invariant **L**

* Only suitable for some situations

LINEARIZING TRANSFORMATIONS

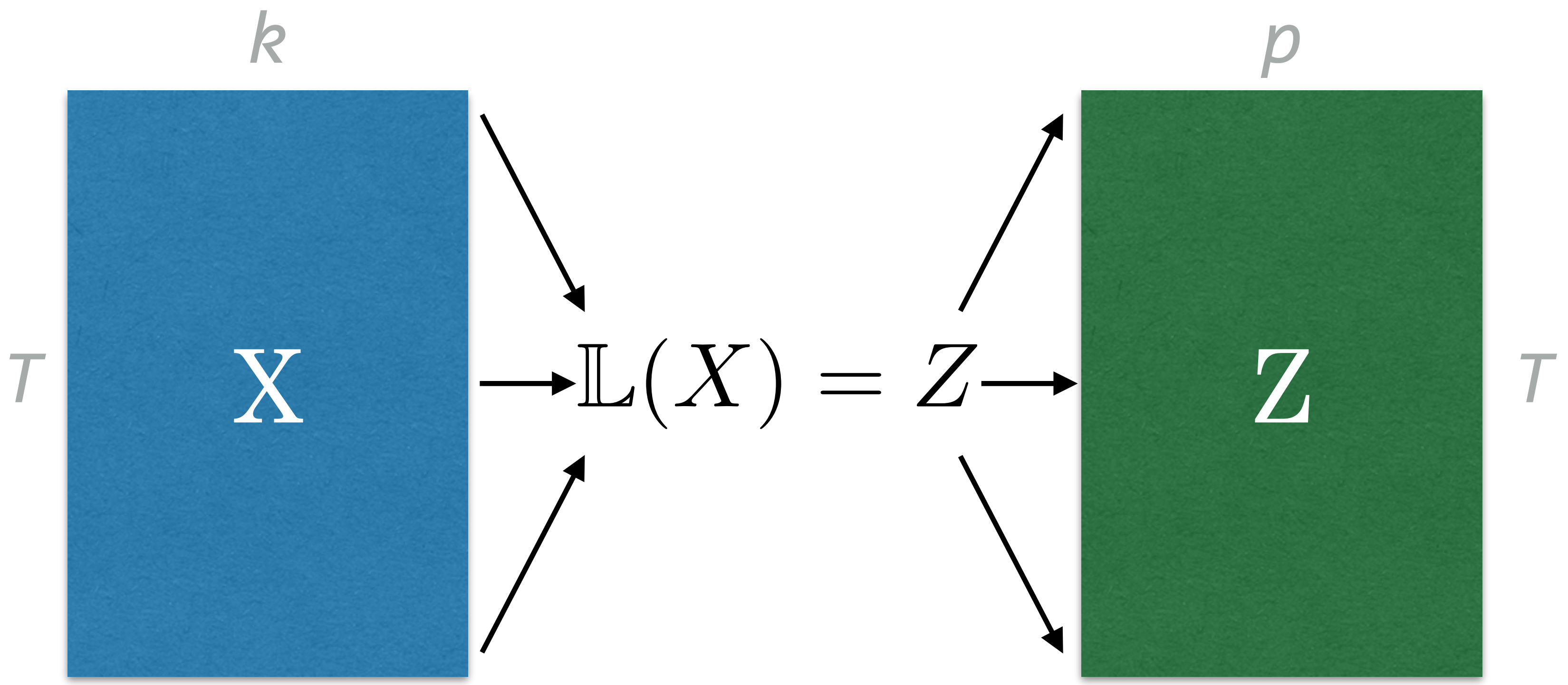
* More complex version: time-dependent **L**

$$X \in \mathbb{R}^{T \times k}$$ *Txk matrix*

$$\mathbb{L}(X) = Z \in \mathbb{R}^{T \times p}$$ *Txp matrix*

LINEARIZING TRANSFORMATIONS

* More complex version: time-dependent **L**



LINEARIZING TRANSFORMATIONS

* More complex version: time-dependent **L**

* **Example:** X_t represents which word was presented as a 1-hot vector. Z_t is the part-of-speech for that word. Time-dependence is necessary because PoS depends on word context.

  e.g. "I refuse" vs. "Refuse bin"

SYNTAX -
PART OF SPEECH

"Now this is a story all about how my life

{adv} {pn} {v} {dt} {n} {adj} {prep} {adv} {pn} {n}

got flipped-turned upside down…"

{v-p} {v-p} {v-p} {prep phrase}

X'

time

words

now	1	0	0
this	0	1	0
is	0	0	1
a	0	0	0
story	0	0	0

… ⋮

→

Z'

time

parts of speech

adv	1	0	0	0	0	0	1	0	0
pn	0	1	0	0	0	0	0	1	0
v	0	0	1	0	0	0	0	0	0
dt	0	0	0	1	0	0	0	0	0
n	0	0	0	0	1	0	0	0	1

… ⋮

SYNTAX - HMM

* **Hidden Markov model (HMM)**

* Example of a **language model**: a function that, given the previous words, returns a probability distribution over the next word.

$$P(w_i | w_1 \ldots w_{i-1}; \theta, \phi)$$

next word *HMM parameters*

previous words

SYNTAX - HMM

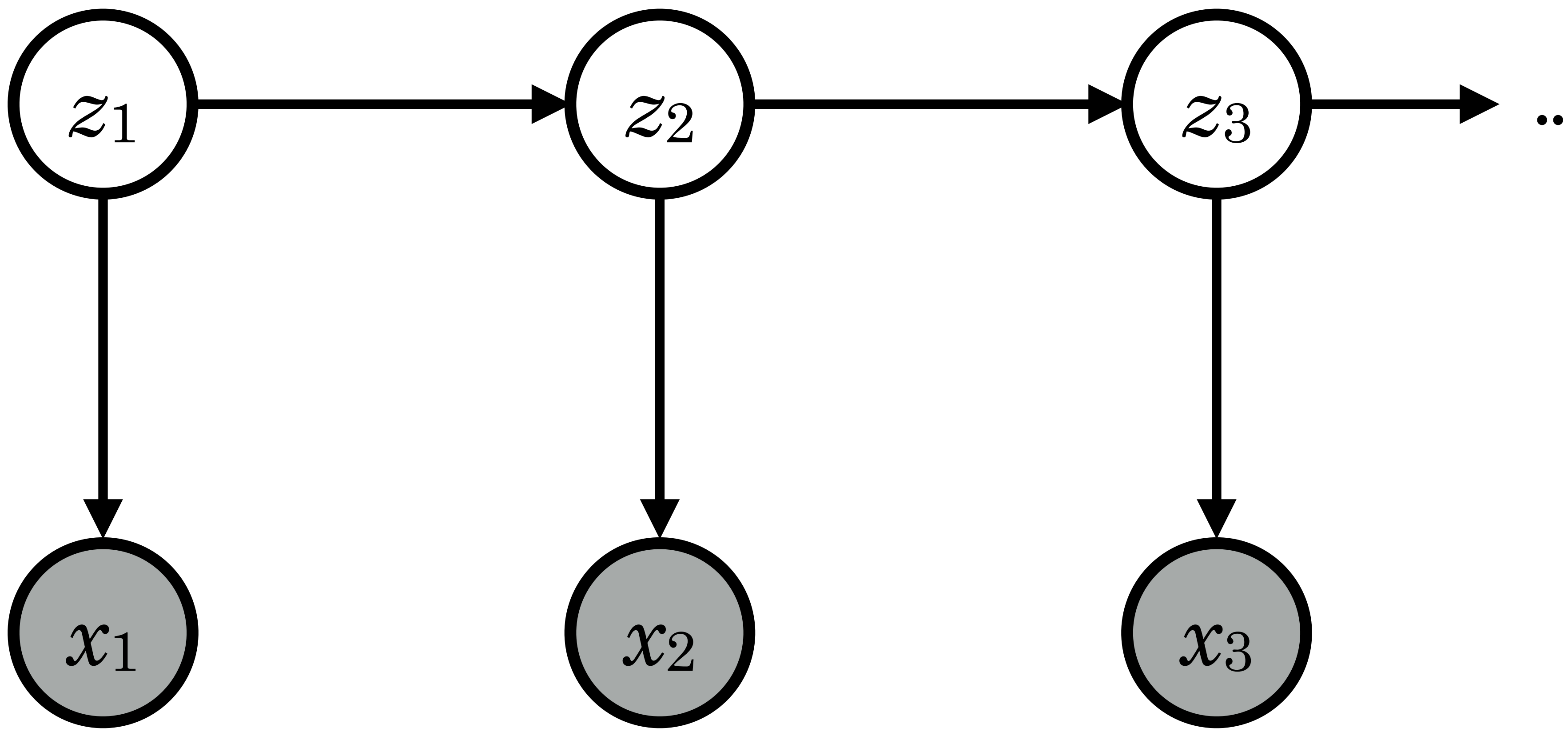
* *We get:* a sequence of observed symbols
  ["now", "this", "is", "a", "story", …]

* *We think:* there are hidden, underlying states
  [a, b, c, d, e, …]

SYNTAX - HMM

$P(z_2|z_1)$ multinomial distribution

Hidden states



Observed symbols

$P(x_1|z_1)$ multinomial distribution

SYNTAX - HMM

X'

time

words			
now	1	0	0
this	0	1	0
is	0	0	1
a	0	0	0
story	0	0	0

…

⋮

Z'

time

hidden state						
a	0.7	0.3	0.0	0.1	0.2	0.1
b	0.1	0.5	0.4	0.0	0.0	0.2
c	0.1	0.1	0.6	0.3	0.2	0.1
d	0.1	0.1	0.0	0.4	0.1	0.1
e	0.0	0.0	0.0	0.2	0.5	0.5

…

⋮

SYNTAX - HMM

* **Learning time:** we know x, what are theta and phi?
* Objective: find theta and phi that *maximize probability* of observed x
* Hyperparameters: number of hidden states (number of possible Z values), priors on theta and phi

SYNTAX - HMM

* **Learning time:** (The easy way) Markov chain Monte Carlo (MCMC) w/ ***Gibbs sampling***
  * (Init.) Guess random Z
  * Update theta & phi based on current X, Z
  * For each Z_t, update based on Z_{t-1}, X_t, phi, & theta
  * Repeat, repeat, repeat, repeat, repeat…

SYNTAX - HMM

* **Inference time:** we know x, we know theta, we know phi; what is $P(z \mid x)$?
* Finally, use inferred state probabilities as features in a linearized model!

SYNTAX - HMM

X'

words	time			
now	1	0	0	…
this	0	1	0	
is	0	0	1	
a	0	0	0	
story	0	0	0	
⋮				

$$X' \longrightarrow Z'$$

Z'

hidden state	time						
a	0.7	0.3	0.0	0.1	0.2	0.1	…
b	0.1	0.5	0.4	0.0	0.0	0.2	
c	0.1	0.1	0.6	0.3	0.2	0.1	
d	0.1	0.1	0.0	0.4	0.1	0.1	
e	0.0	0.0	0.0	0.2	0.5	0.5	
⋮							

LEXICAL SEMANTICS

* Let's create an **L** that captures word-level semantic information

* Unlike the awful syntax models, this model will be *time-invariant*

LEXICAL SEMANTICS

1-hot vector — *p-dim vector*

$$\mathbb{L}_{semantic}(X_t) = Z_t \in \mathbb{R}^p$$

X'

time

words			
now	1	0	0
this	0	1	0
is	0	0	1
a	0	0	0
story	0	0	0
⋮			

…

$\longrightarrow$

Z'

time

features

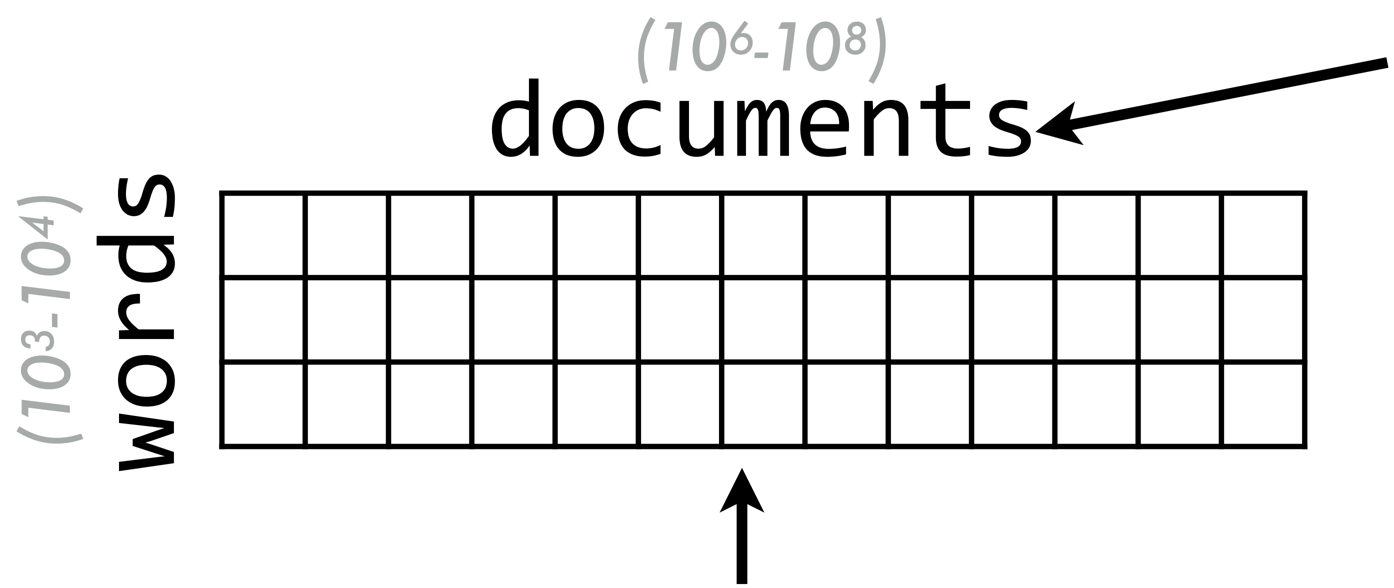
somehow each column captures something about the meaning of the corresponding word

…

⋮

LEXICAL SEM. - LSA

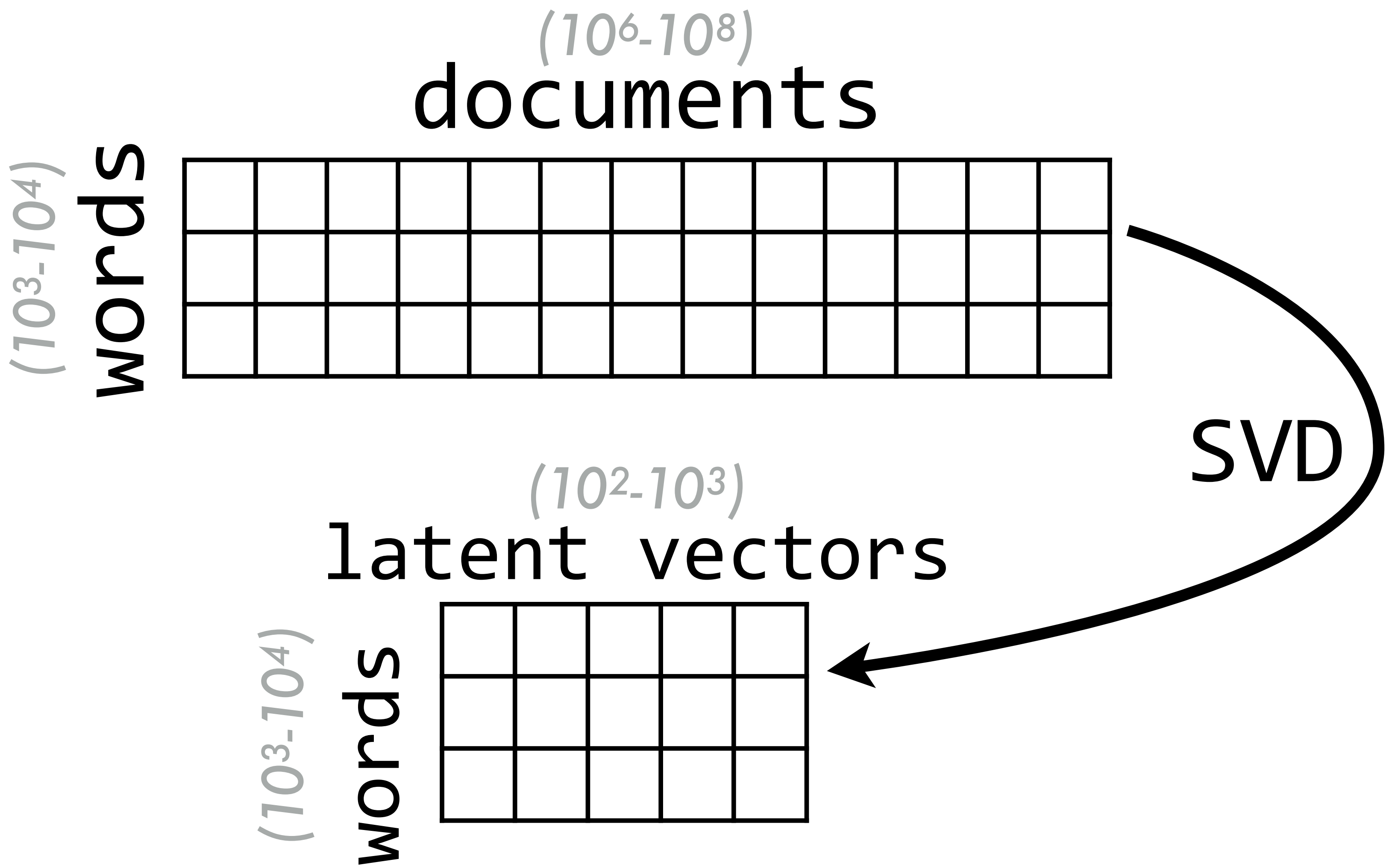
* Latent Semantic Analysis (LSA)



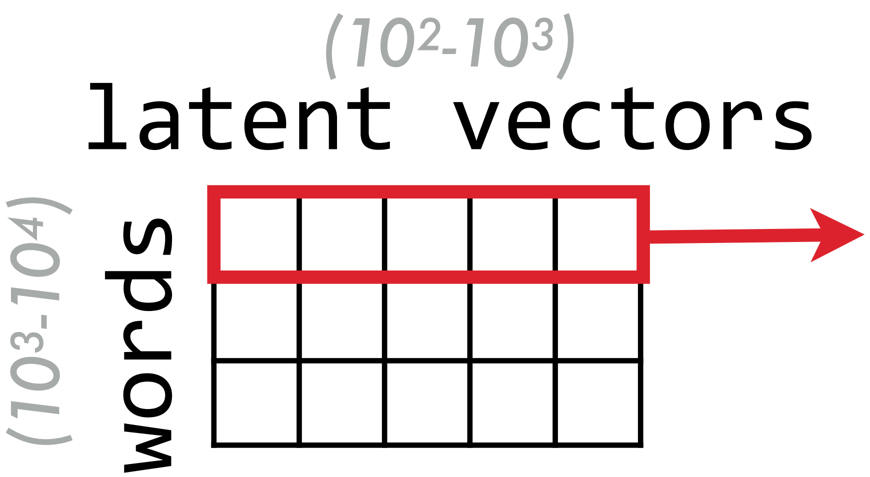
Deerwester et al. (1988)

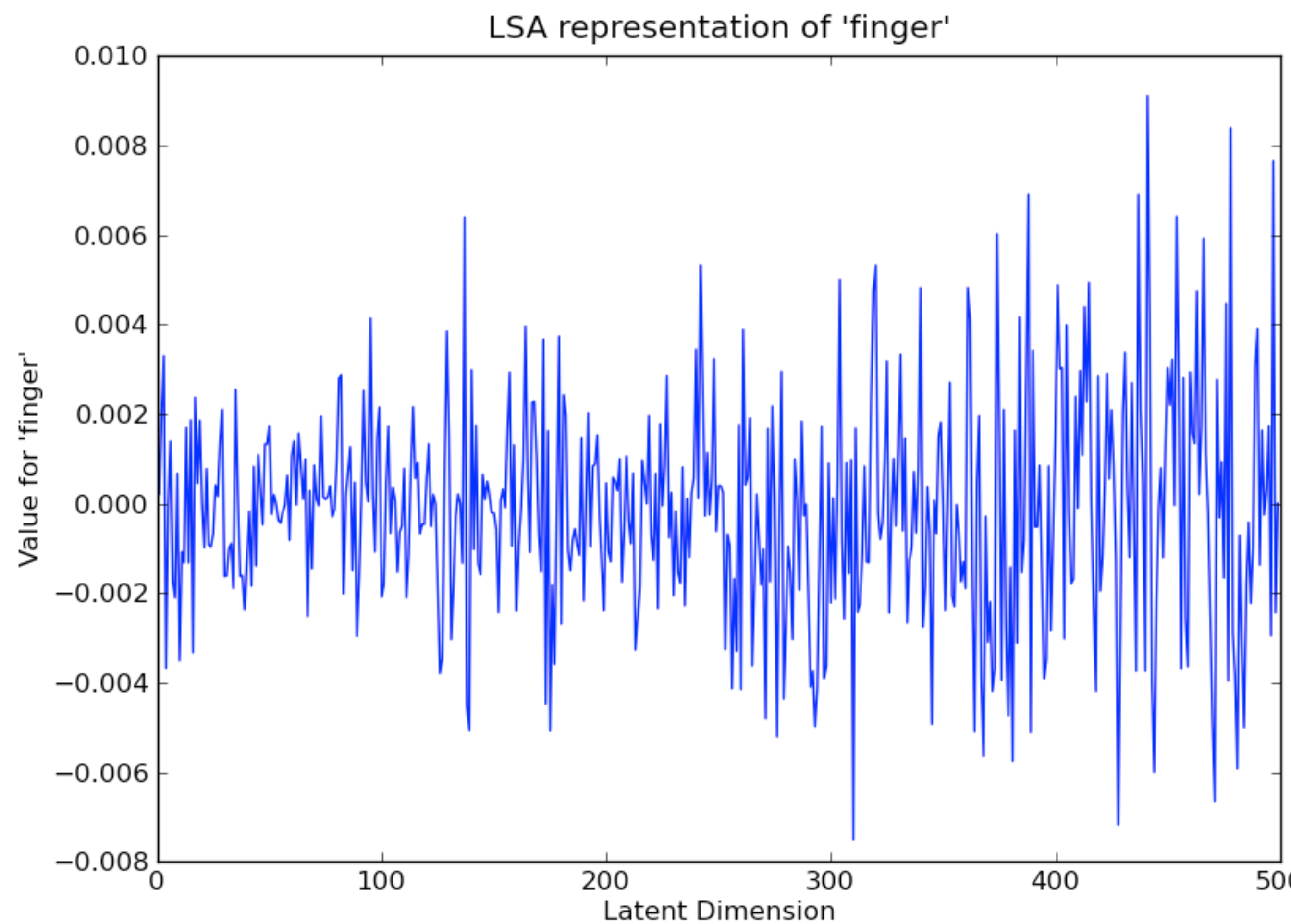
LEXICAL SEM. - LSA

* Latent Semantic Analysis (LSA)

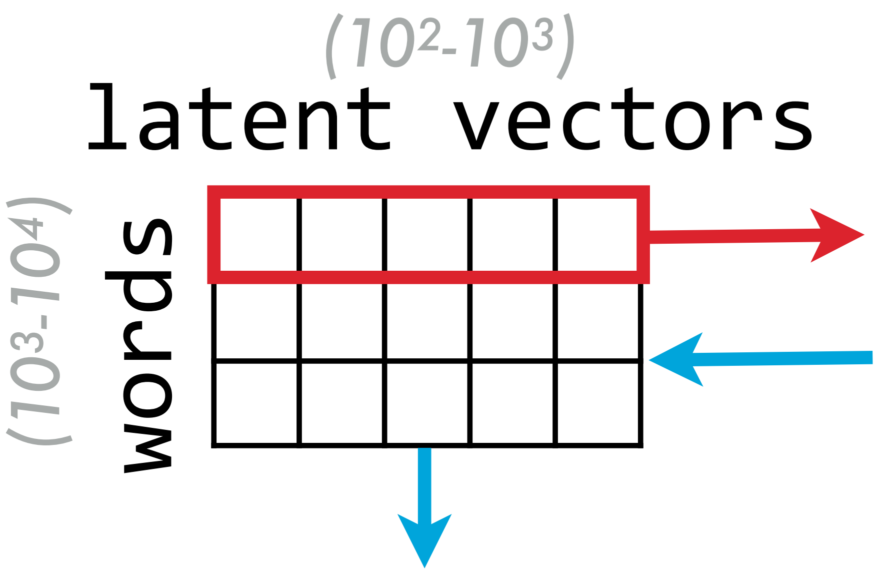


LEXICAL SEM. - LSA

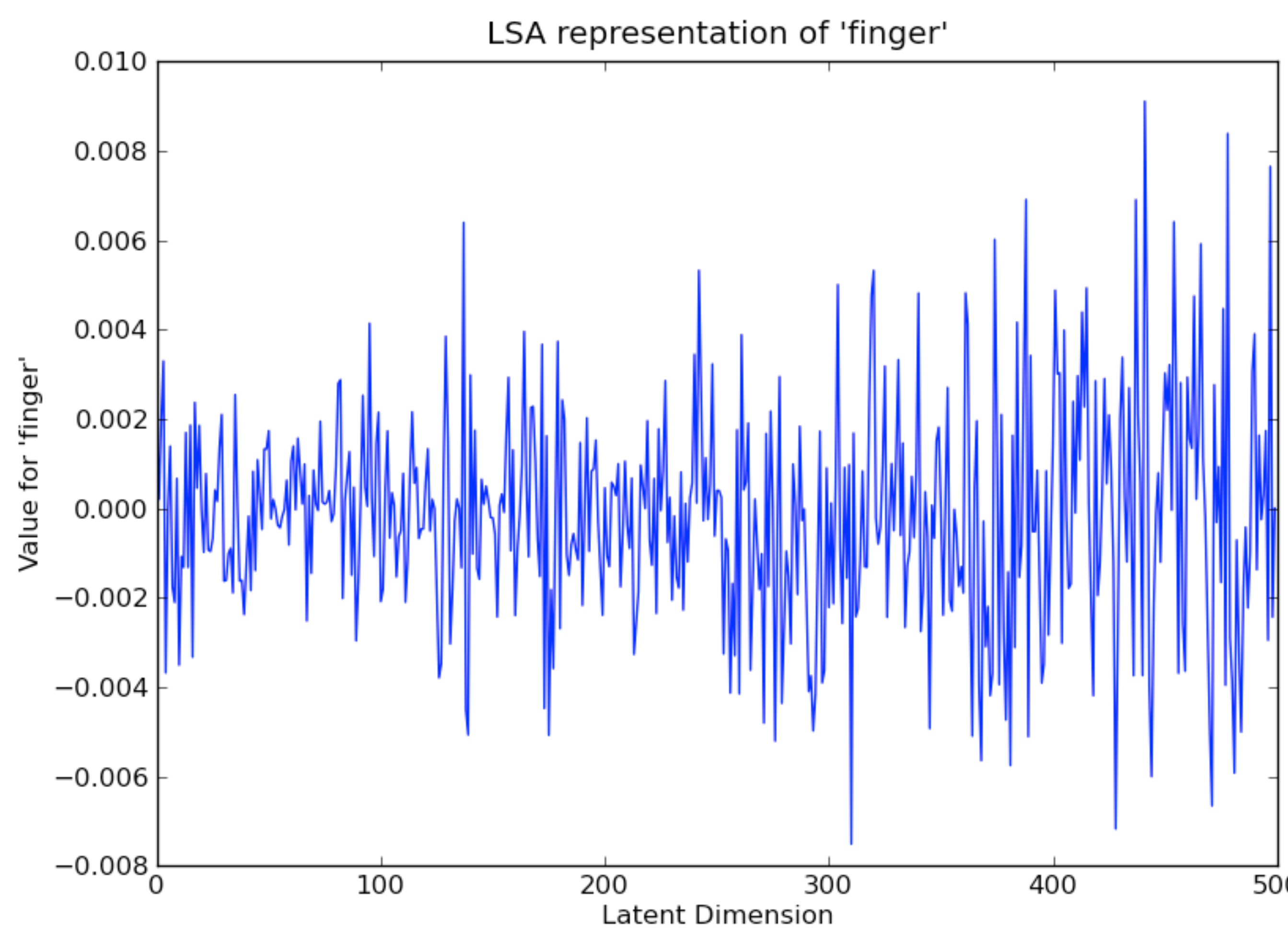




LEXICAL SEM. - LSA



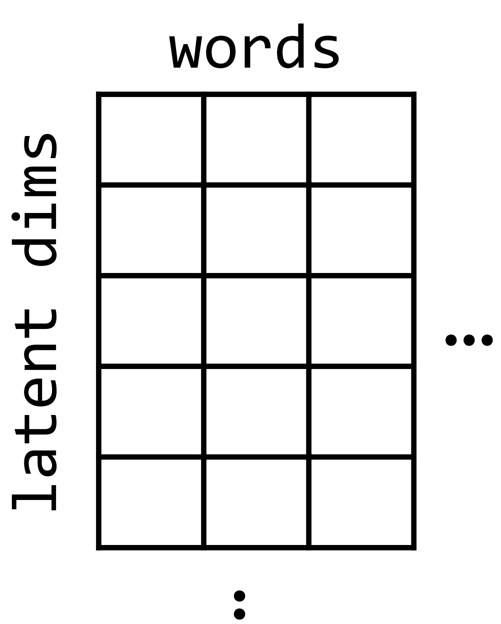
0.61- thumb
0.52- fingers
0.51- forefinger
0.49- tip
0.47- wrist
0.41- elbow
0.41- throat
0.41- lips
0.40- string



LEXICAL SEM. - LSA

$$E \cdot X' = Z'$$

embedding matrix' * word matrix' = semantic stimulus matrix'



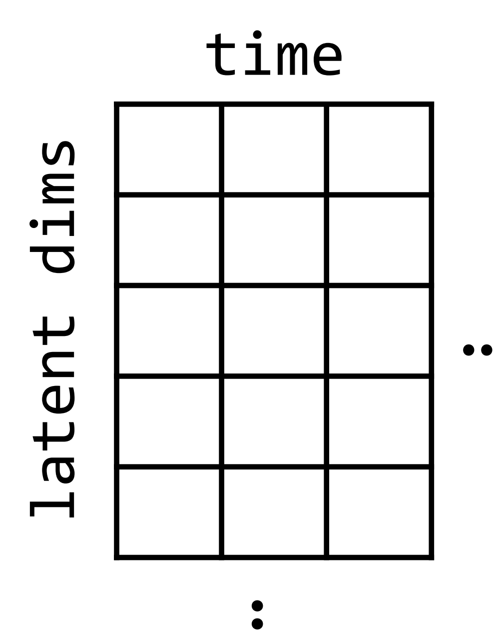
time

words

now	1	0	0
this	0	1	0
is	0	0	1
a	0	0	0
story	0	0	0

...

⋮



REMINDER FROM A FEW WEEKS AGO…

TIKHONOV REGRESSION

* this is equivalent to **TIKHONOV REGRESSION** on the **WORDS** with a prior determined by the **WORD EMBEDDING**

$$\underbrace{\frac{1}{\sigma^2}\Sigma_\beta}_{\text{PRIOR COVARIANCE}} = \underbrace{(C^T C)^{-1}}_{\text{INVERSE OF PENALTY INNER PRODUCT}} = \underbrace{E^T E}_{\text{EMBEDDING INNER PRODUCT}}$$

* **i.e. the prior covariance between two words' weights is equal to the dot product of their embedding vectors**

NEXT TIME

* MORE SEMANTICS!!