

Topic Modeling

CSCI 1460: Computational Linguistics
Lecture 7

Ellie Pavlick
Fall 2023

Announcements

- Assignment 2 will be out tomorrow (due Sunday Oct 16)
 - Grading will still begin on Monday
- NLP talks in the department!
 - **Oct 7th, 2pm Jennifer Hu** A Targeted Evaluation of Human-like Linguistic Knowledge in Neural Language Models
 - **Oct 13th, 12pm Panel/Discussion** Strong vs. Weak Compositionality in Humans and Machines!

Topics

- Lecture 6 Followup: Word Embeddings from SVD
- What is a topic model
- Latent Semantic Analysis (LSA)
- Latent Dirichelet Allocation (LDA)
 - “Generative Stories”
 - Graphical Model Notation
 - Training and Evaluation

Topics

- **Lecture 6 Followup: Word Embeddings from SVD**
- What is a topic model
- Latent Semantic Analysis (LSA)
- Latent Dirichelet Allocation (LDA)
 - “Generative Stories”
 - Graphical Model Notation
 - Training and Evaluation

Singular Value Decomposition

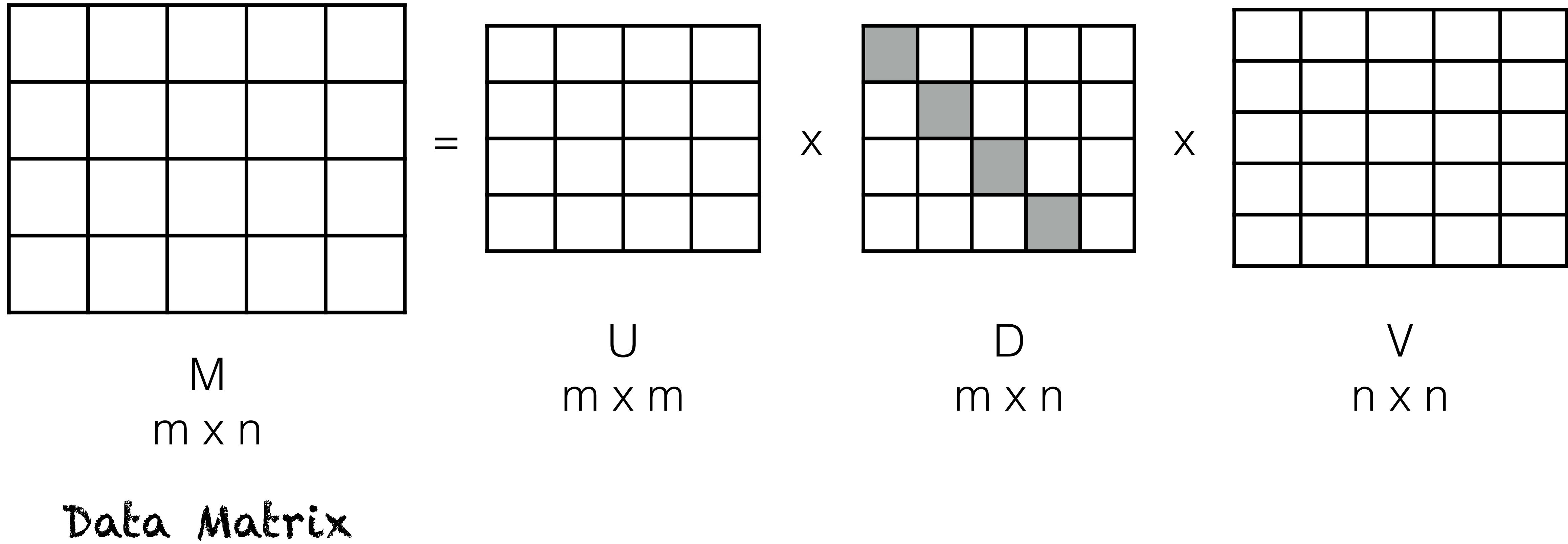
$$M_{m \times n} = U_{m \times m} \times D_{m \times n} \times V_{n \times n}$$

The diagram illustrates the Singular Value Decomposition (SVD) of a matrix M . It shows M as a product of three matrices: U , D , and V . The matrices are represented by grids:

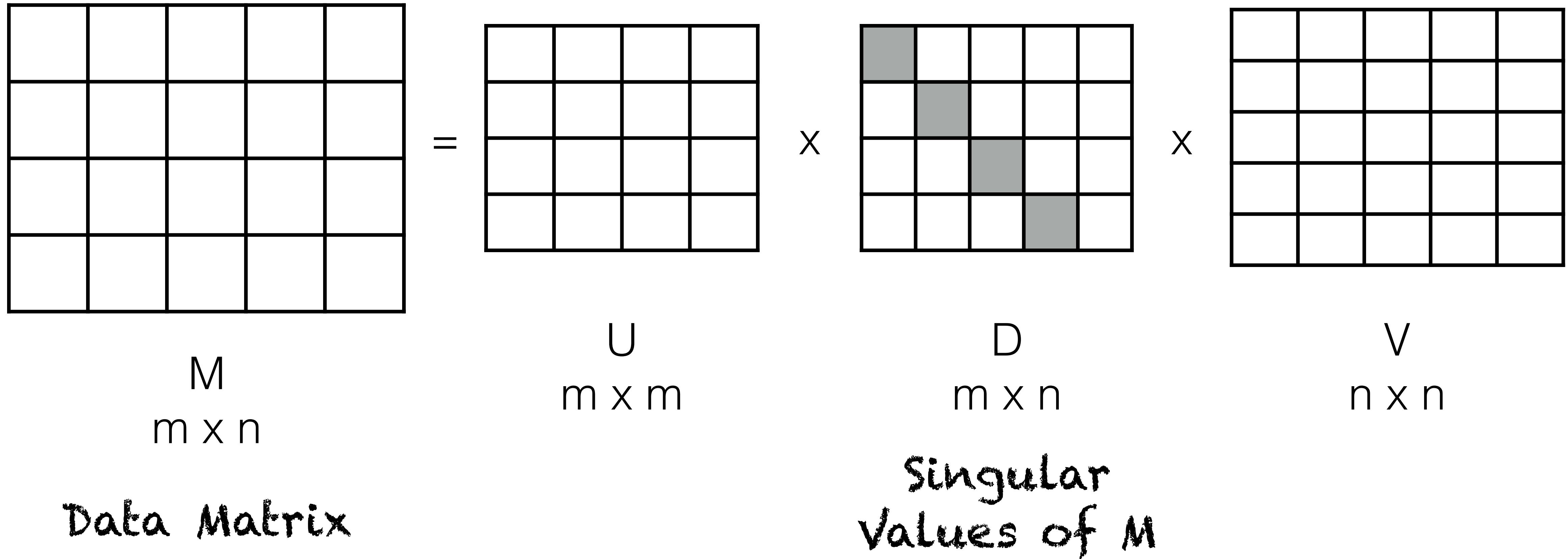
- M is a $m \times n$ grid.
- U is an $m \times m$ grid.
- D is an $m \times n$ grid with shaded (gray) entries in the main diagonal.
- V is an $n \times n$ grid.

The multiplication is indicated by the equals sign ($=$) followed by a times sign (\times) between U and D , and another times sign (\times) between D and V .

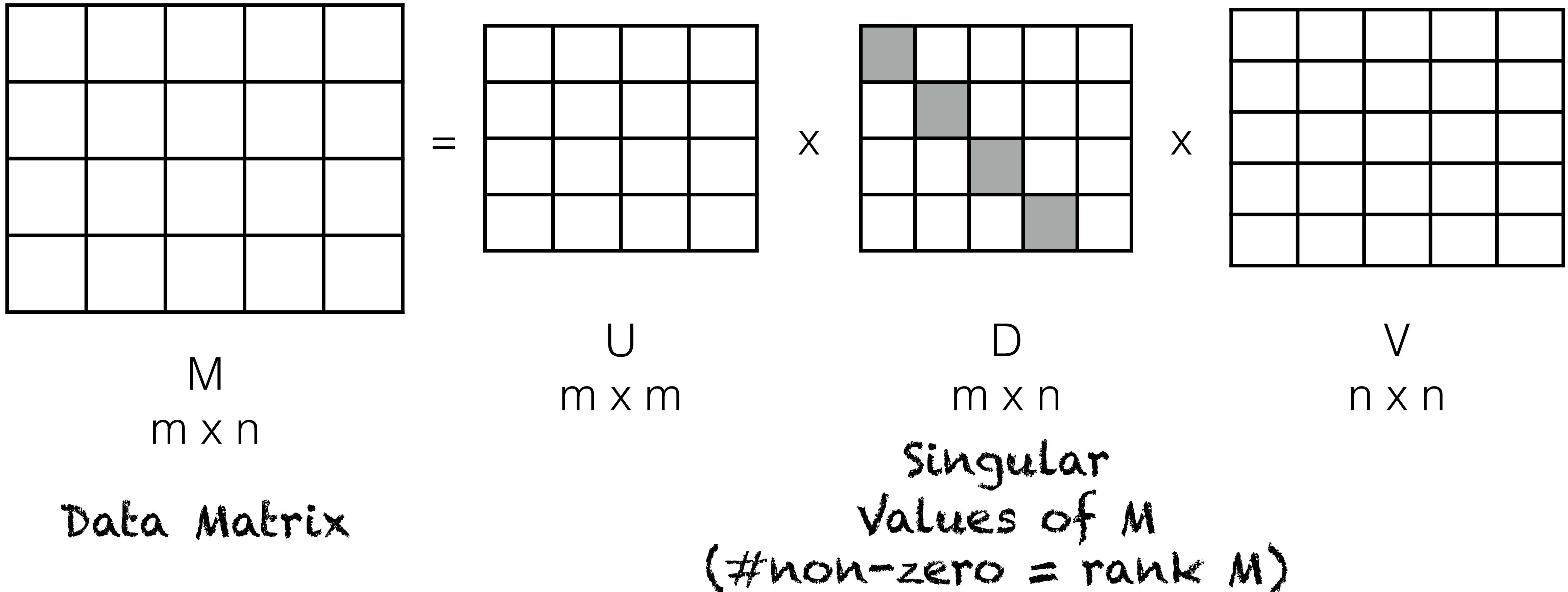
Singular Value Decomposition



Singular Value Decomposition

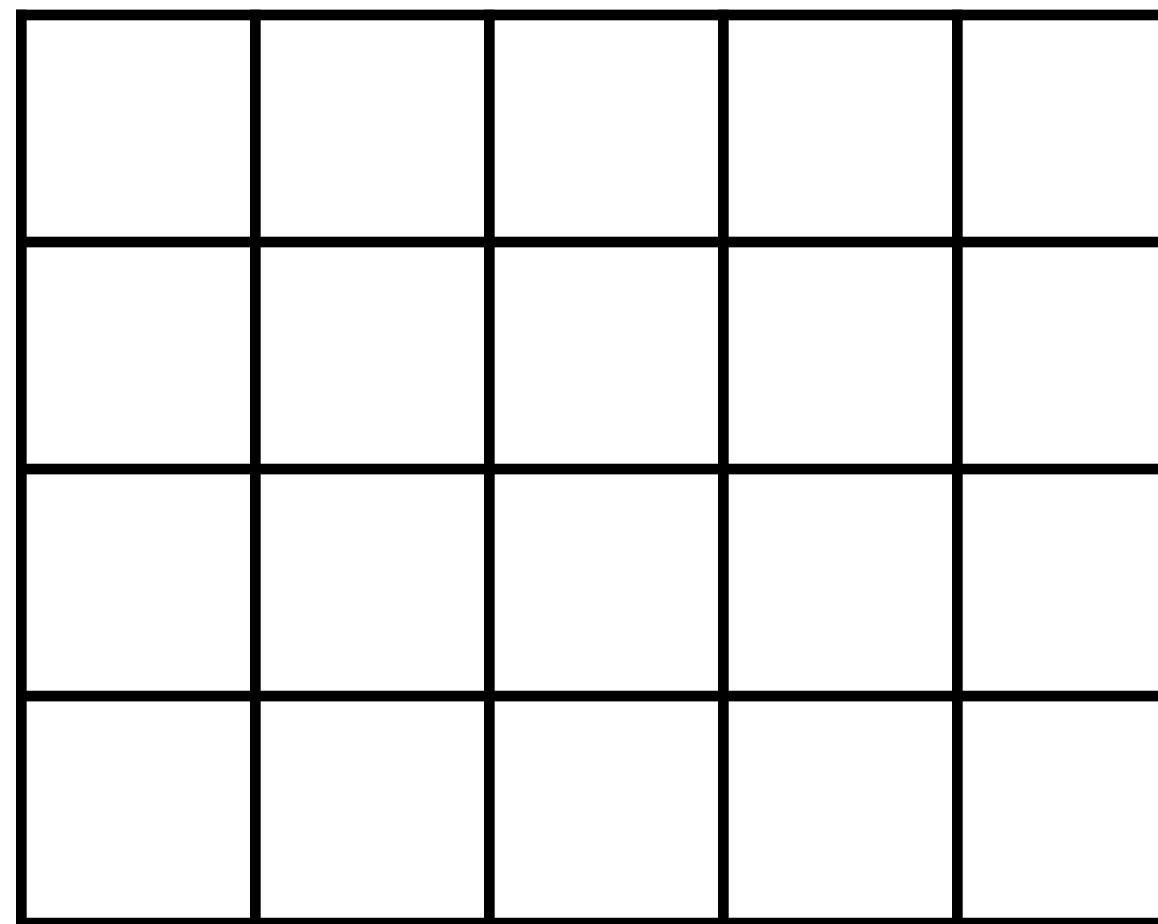


Singular Value Decomposition



Singular Value Decomposition

Representation of
rows of M in new
feature space



M
 $m \times n$

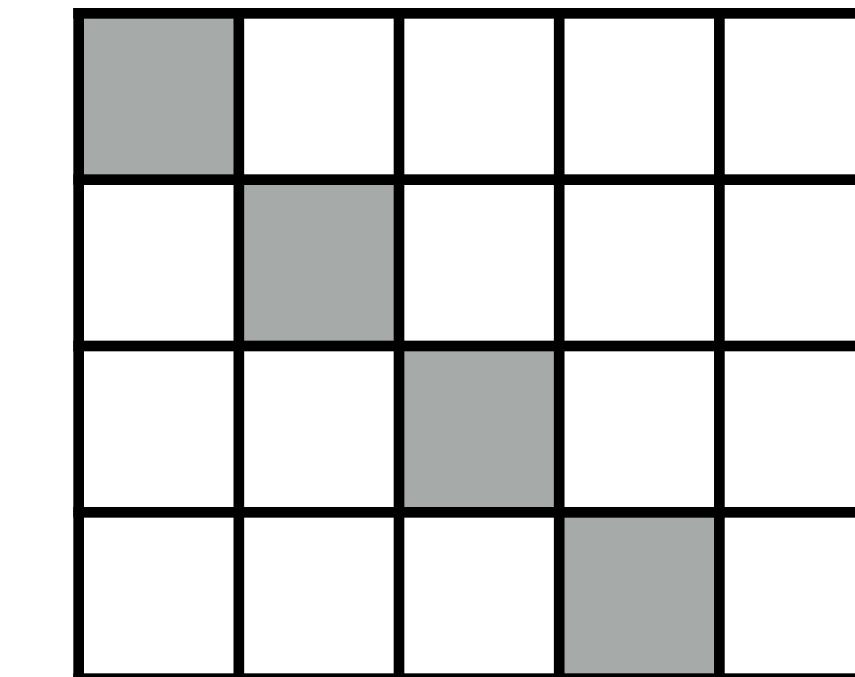
Data Matrix

=



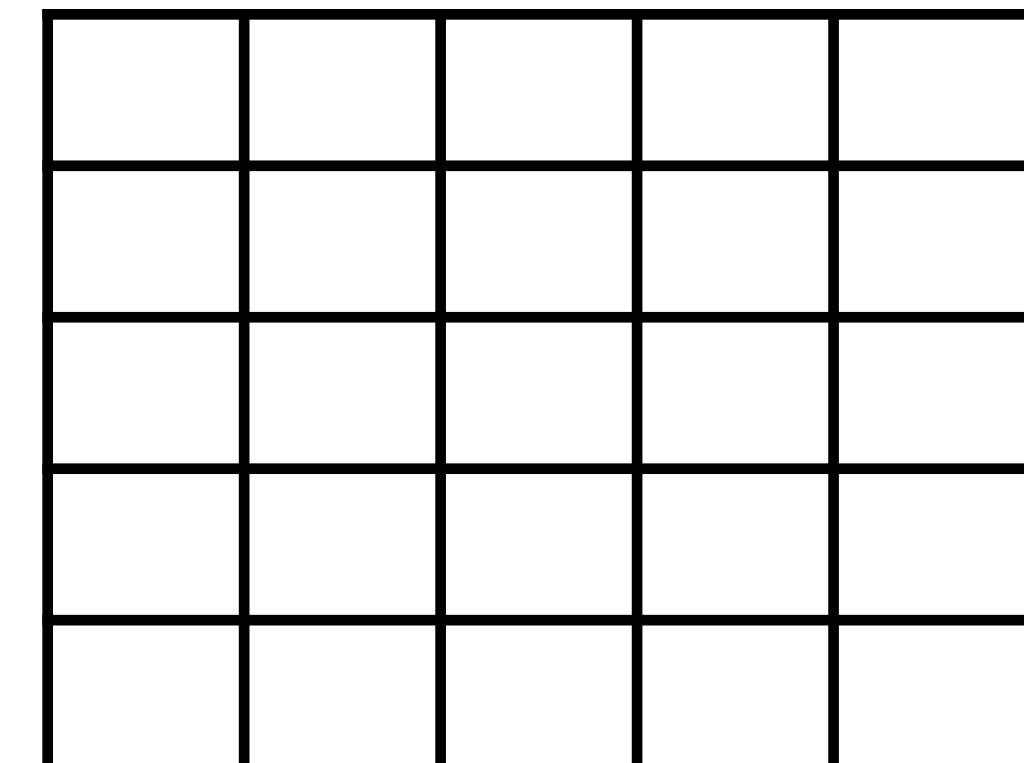
U
 $m \times m$

\times



D
 $m \times n$

\times



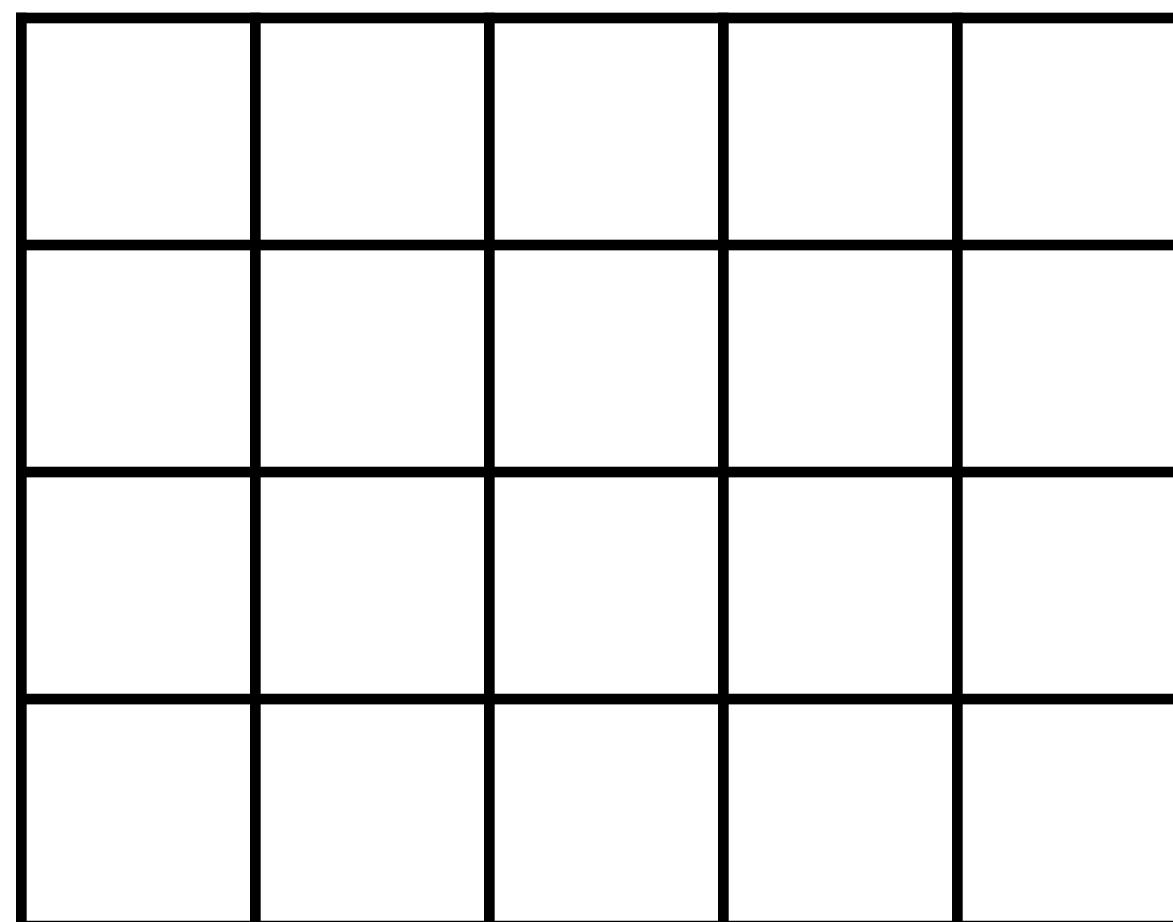
V
 $n \times n$

Singular
Values of M
(#non-zero = rank M)

Singular Value Decomposition

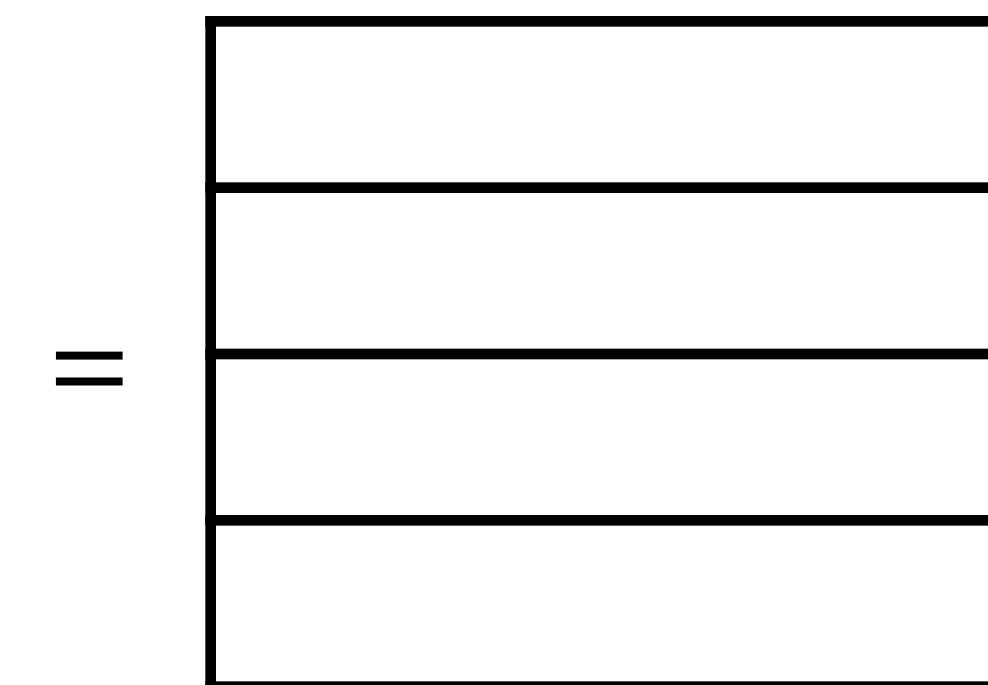
Representation of
rows of M in new
feature space

Principle
Components
(new features)

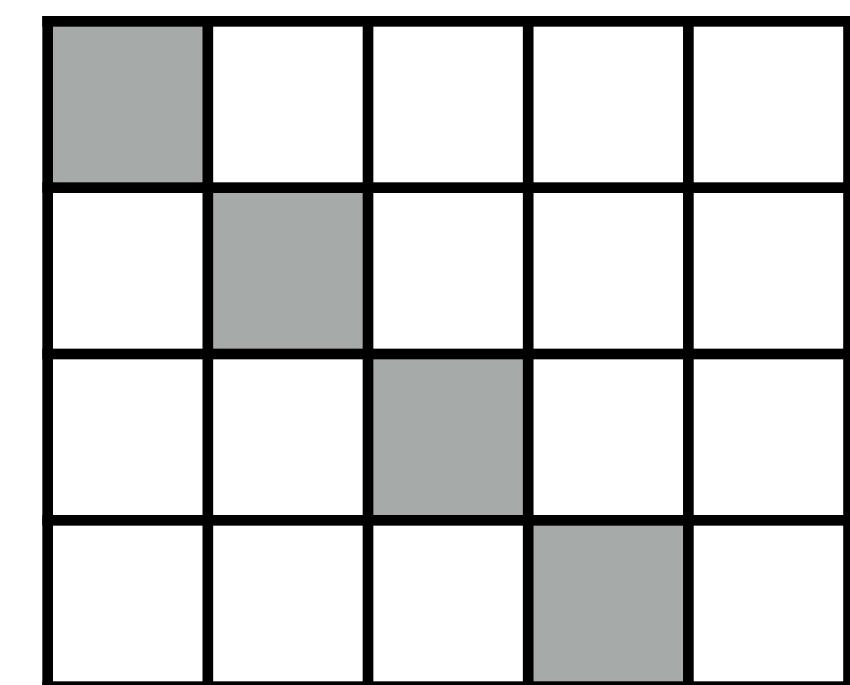


M
 $m \times n$

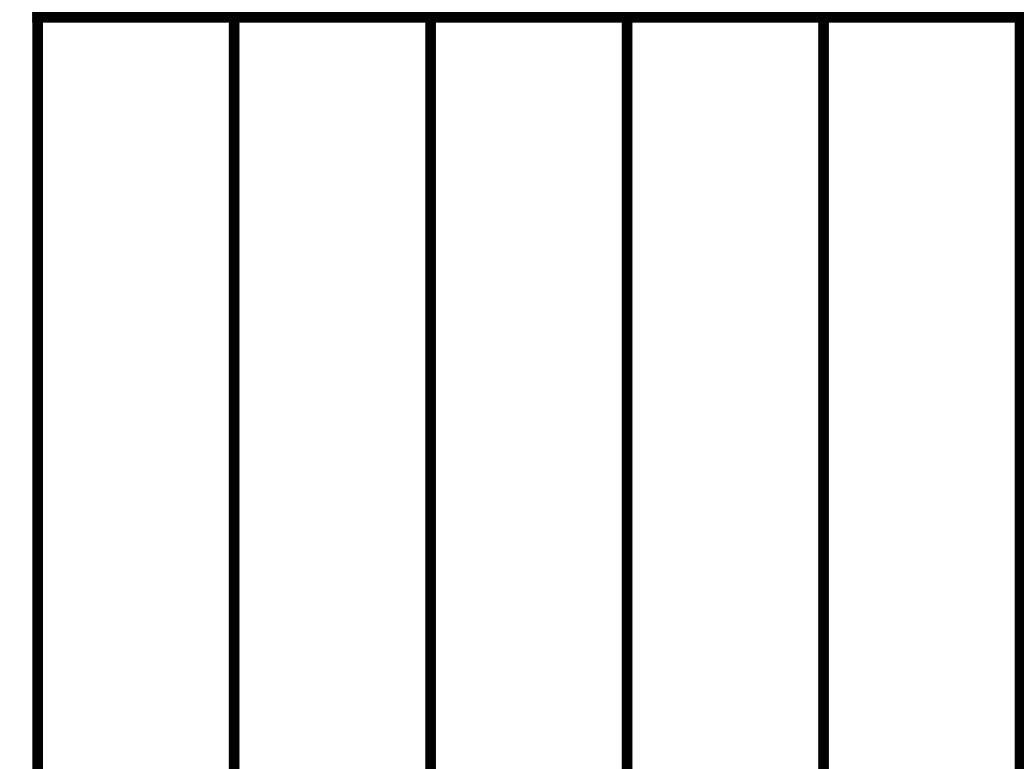
Data Matrix



U
 $m \times m$



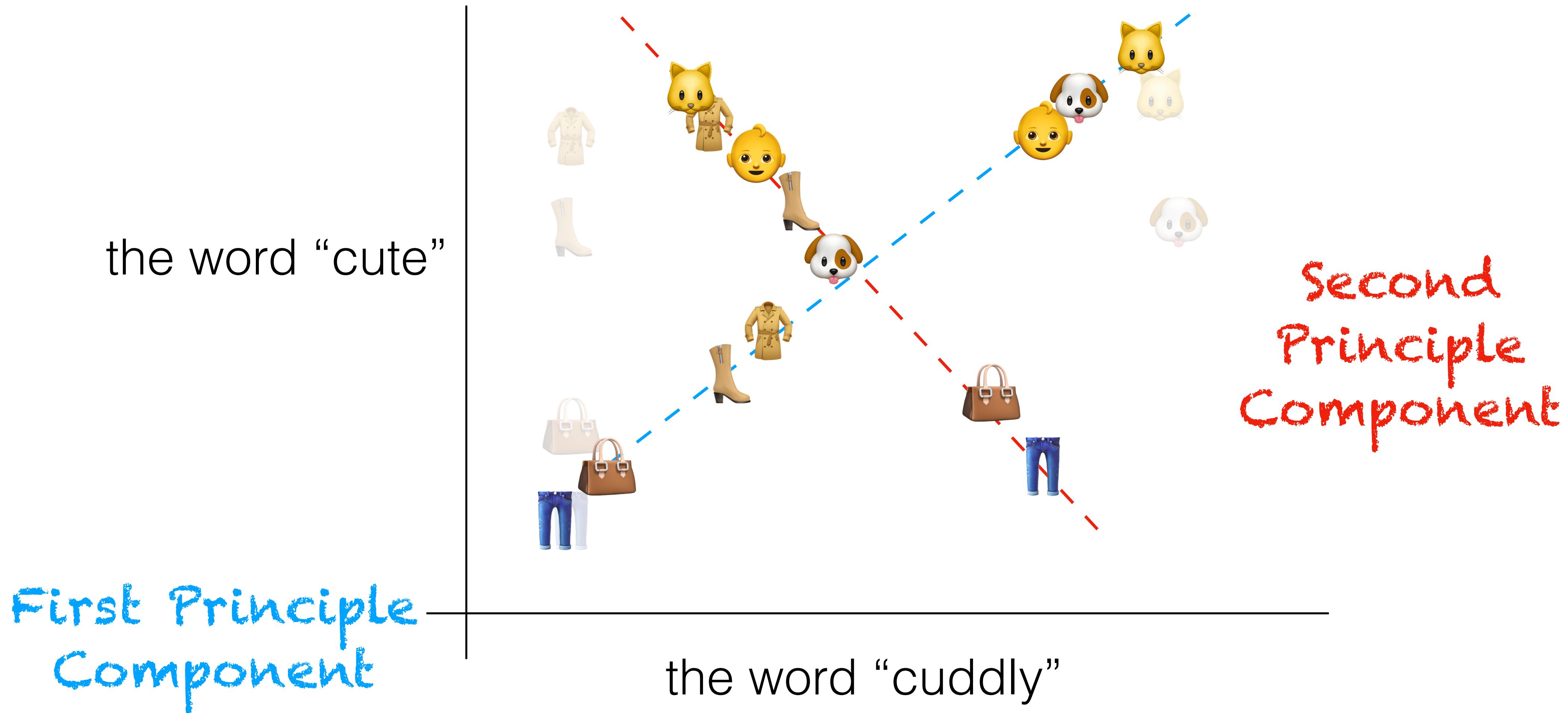
D
 $m \times n$



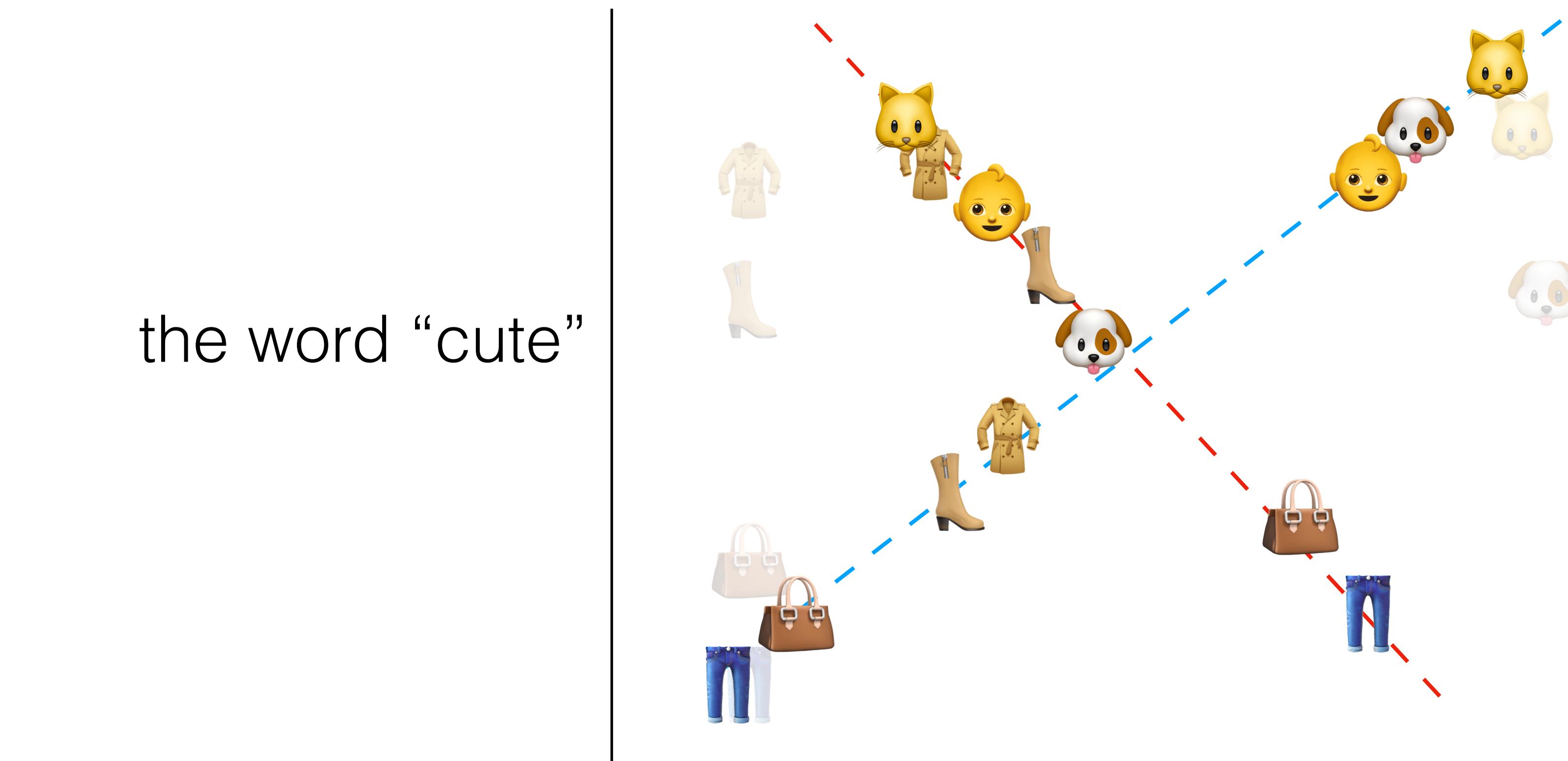
V
 $n \times n$

Singular
Values of M
(#non-zero = rank M)

Singular Value Decomposition



Singular Value Decomposition



First Principle
Component

the word “cuddly”

0.5 “cuddly”	-0.5 “cuddly”
0.5 “cute”	0.5 “cute”

V

$n \times n$

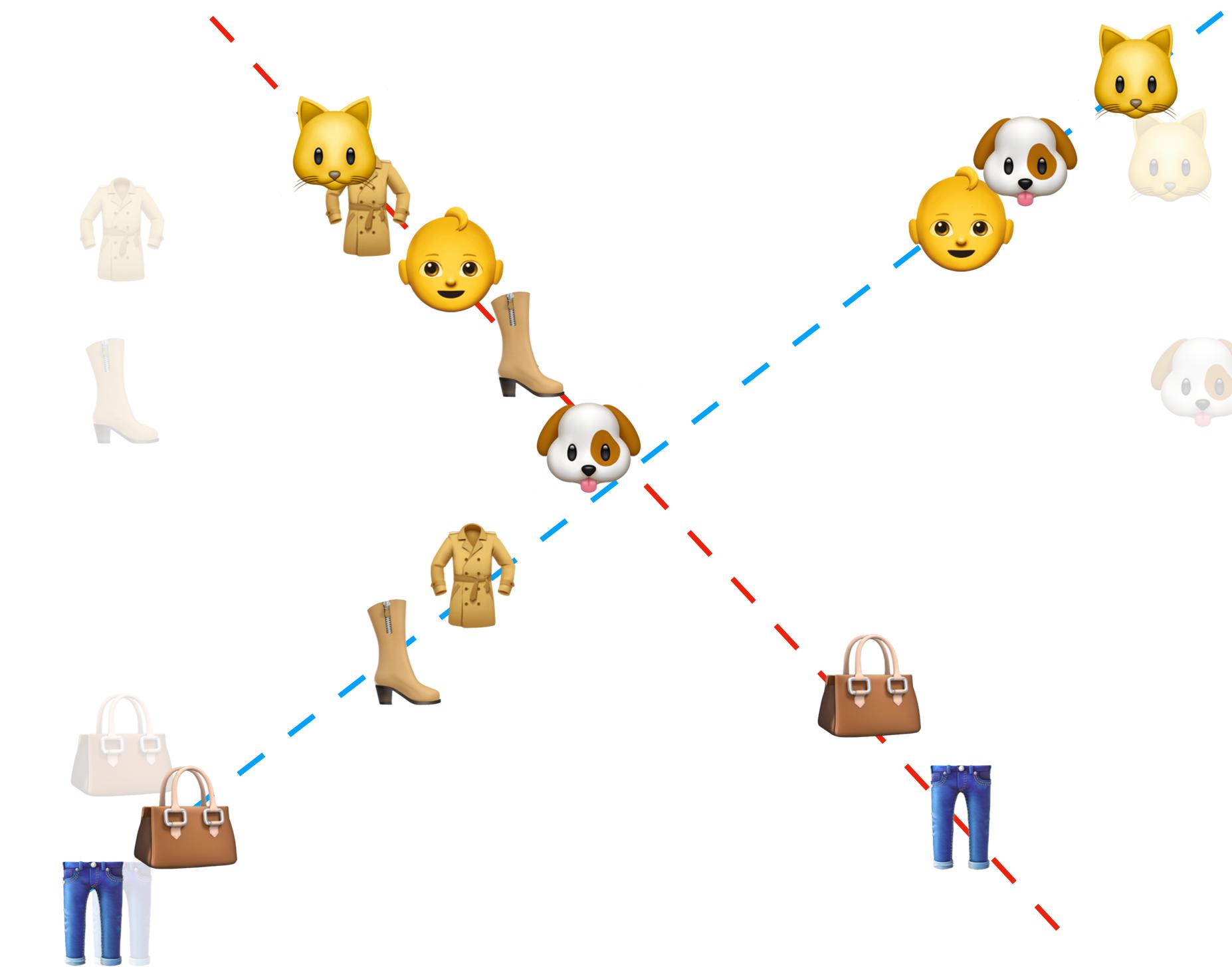
Second
Principle
Component

*warning: numbers are made up (not necessarily to scale)

Singular Value Decomposition

the word “cute”

the word “cuddly”



	3	-3
	2.5	-0.5
	-2	1.5
	-2.5	2

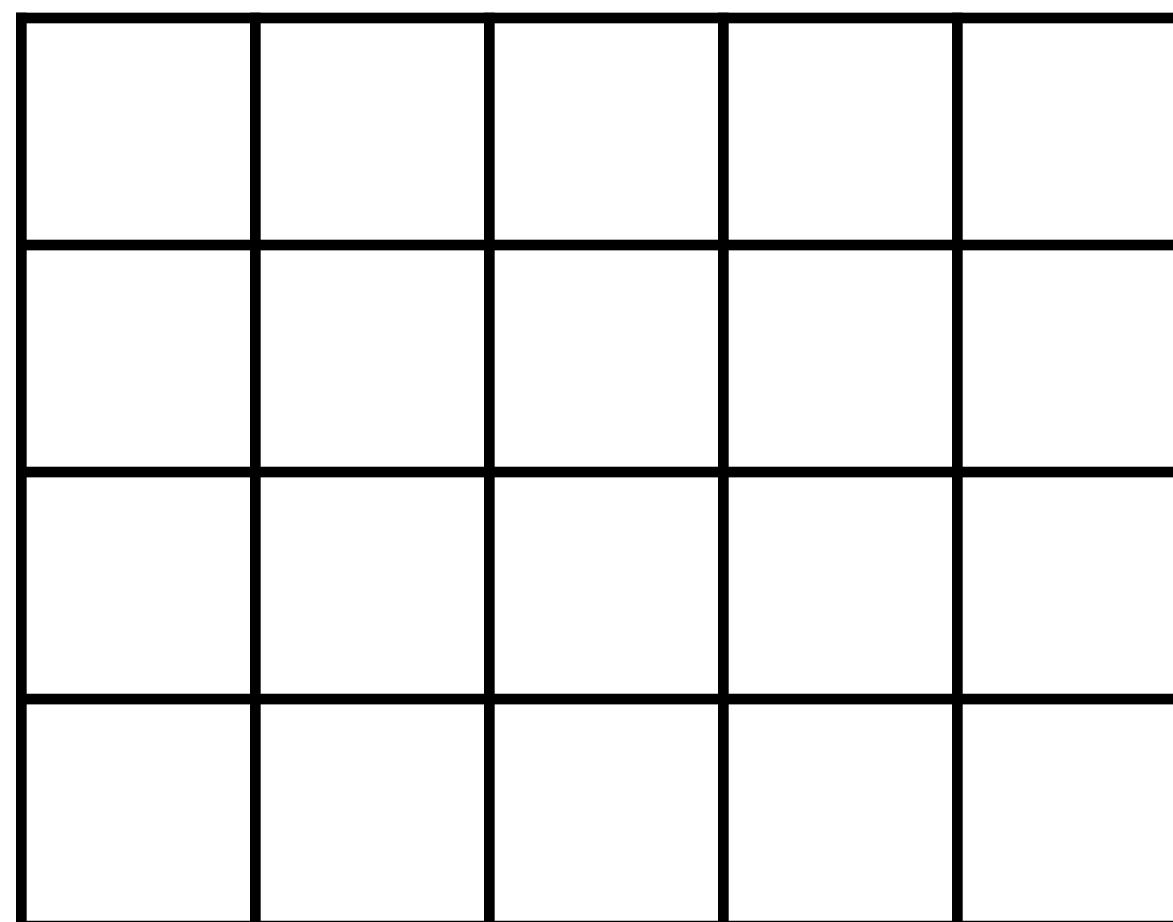
U
 $m \times m$

*warning: numbers are made up (not necessarily to scale)

Singular Value Decomposition

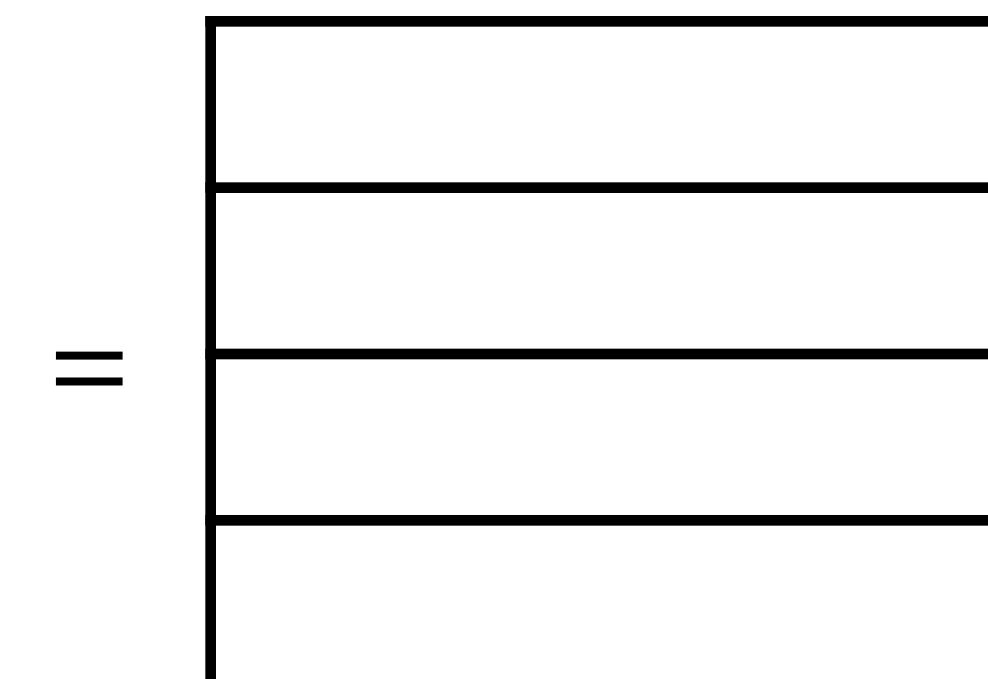
Representation of
rows of M in new
feature space

Principle
Components
(new features)

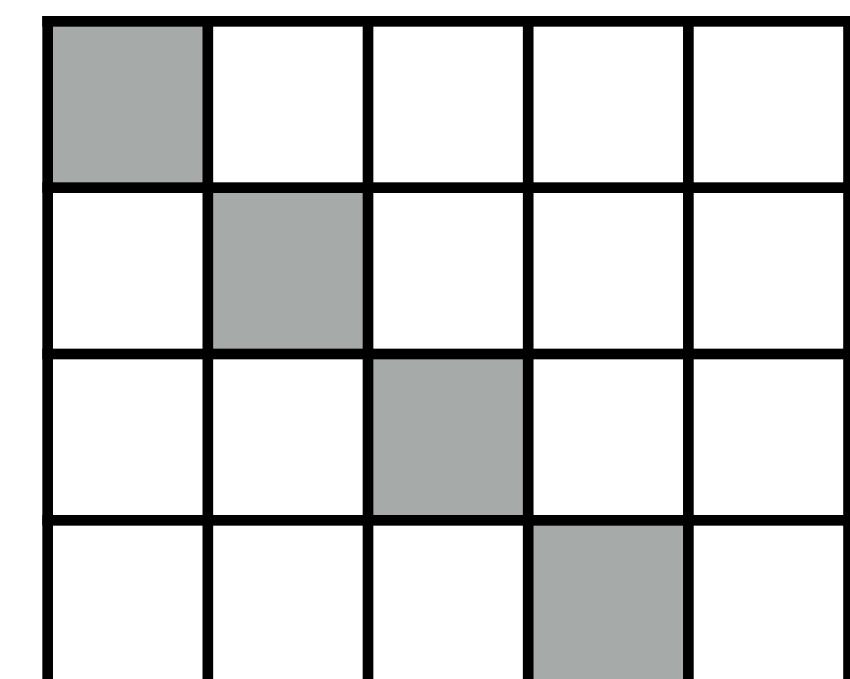


M
 $m \times n$

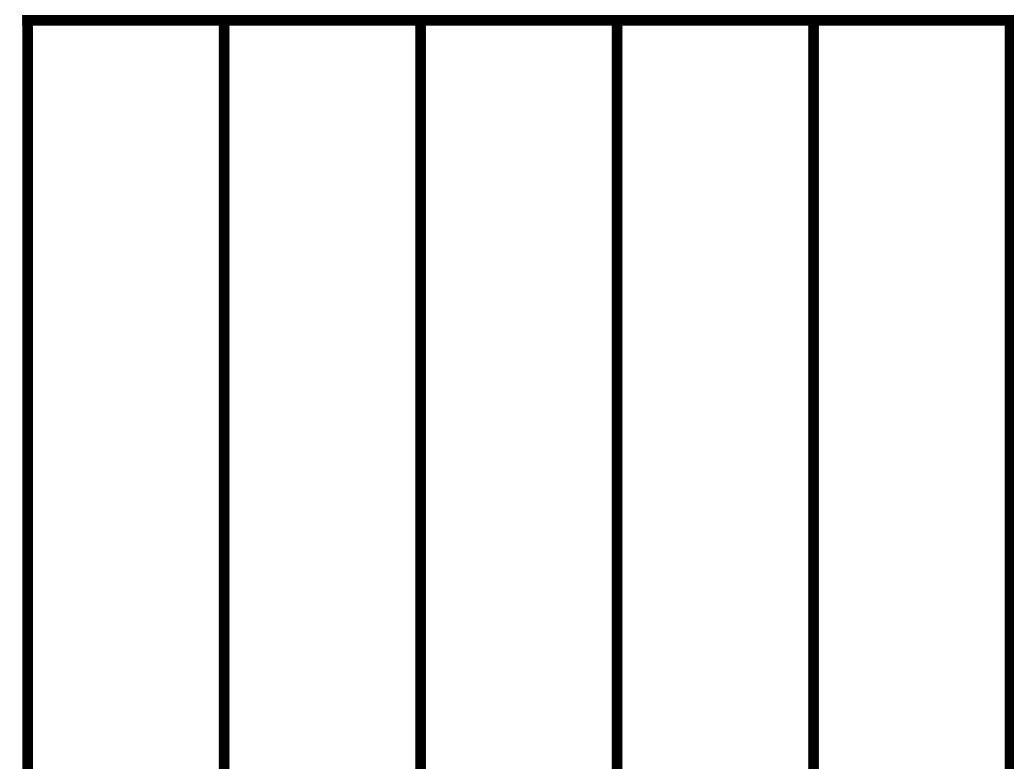
Data Matrix



U
 $m \times m$



D
 $m \times n$

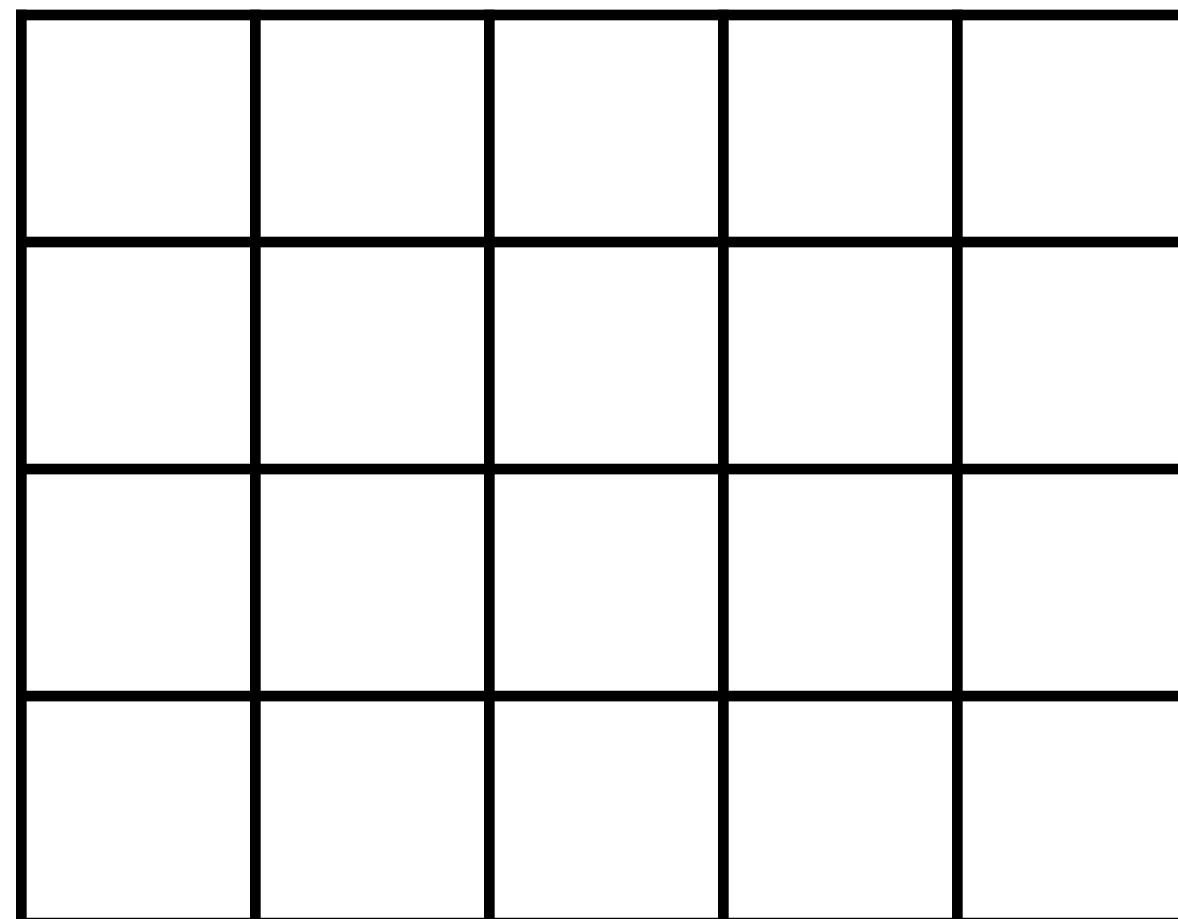


V
 $n \times n$

Singular
Values of M
(#non-zero = rank M)

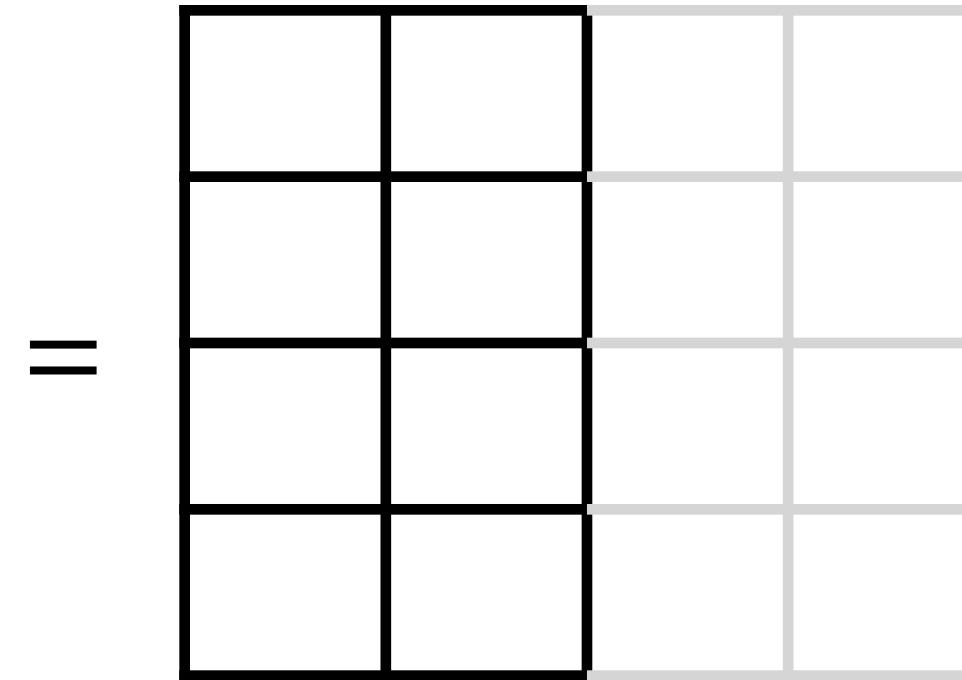
Truncated SVD

keep only first L components

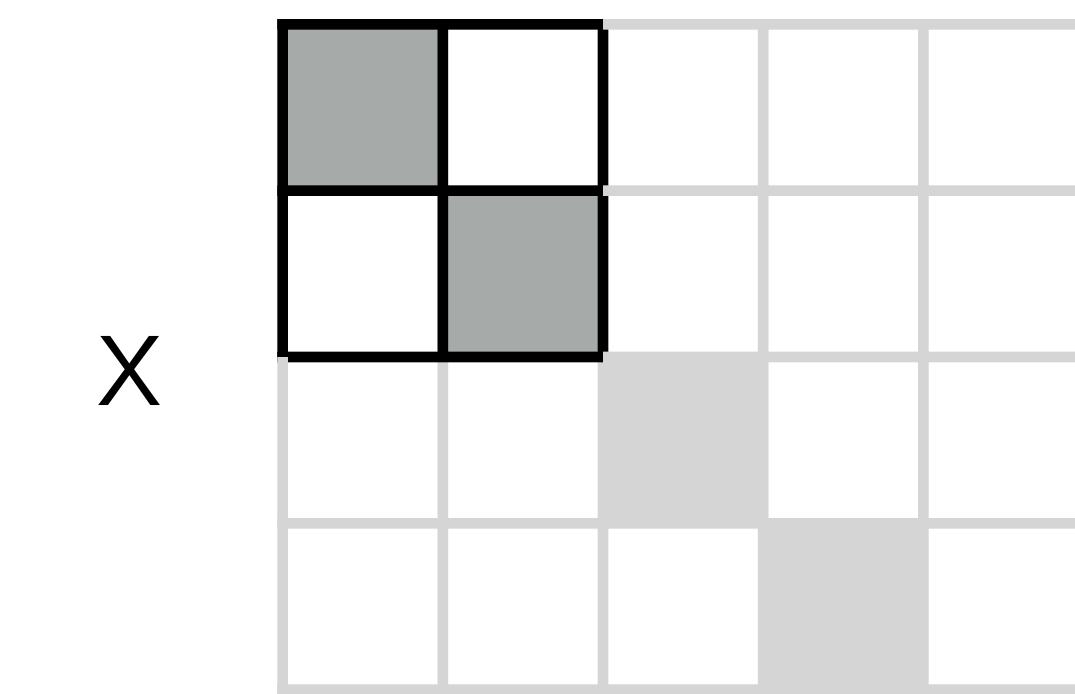


M
 $m \times n$

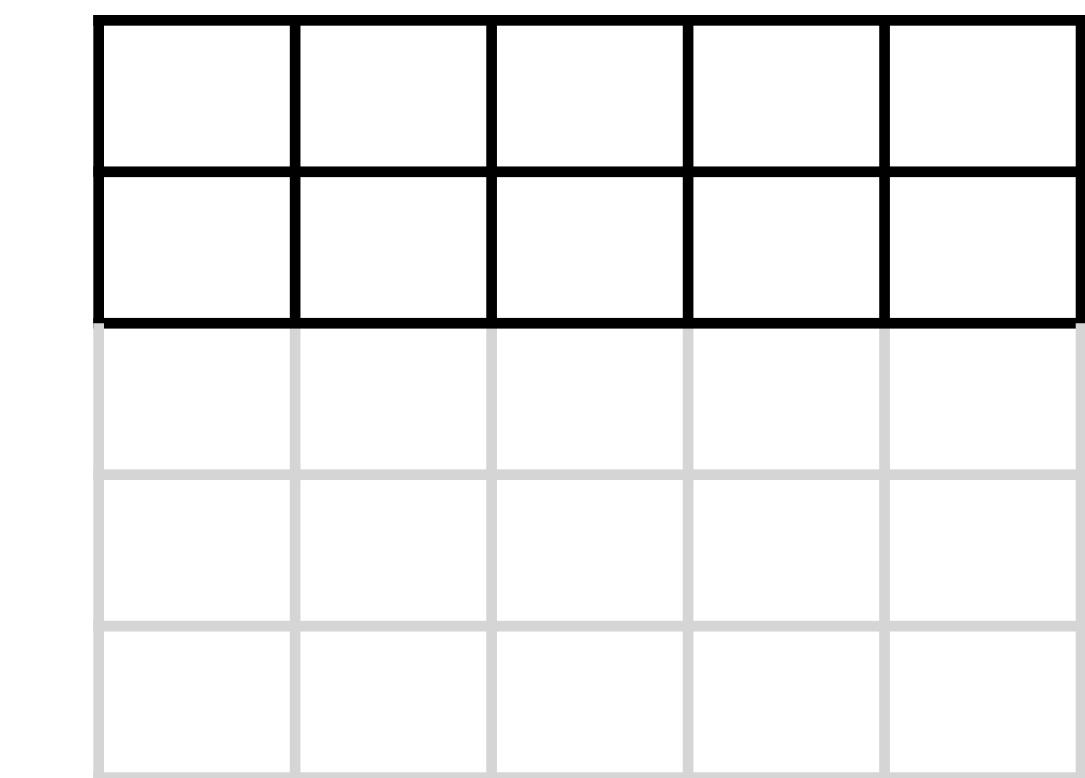
Data Matrix



U
 $m \times l$



D
 $l \times l$

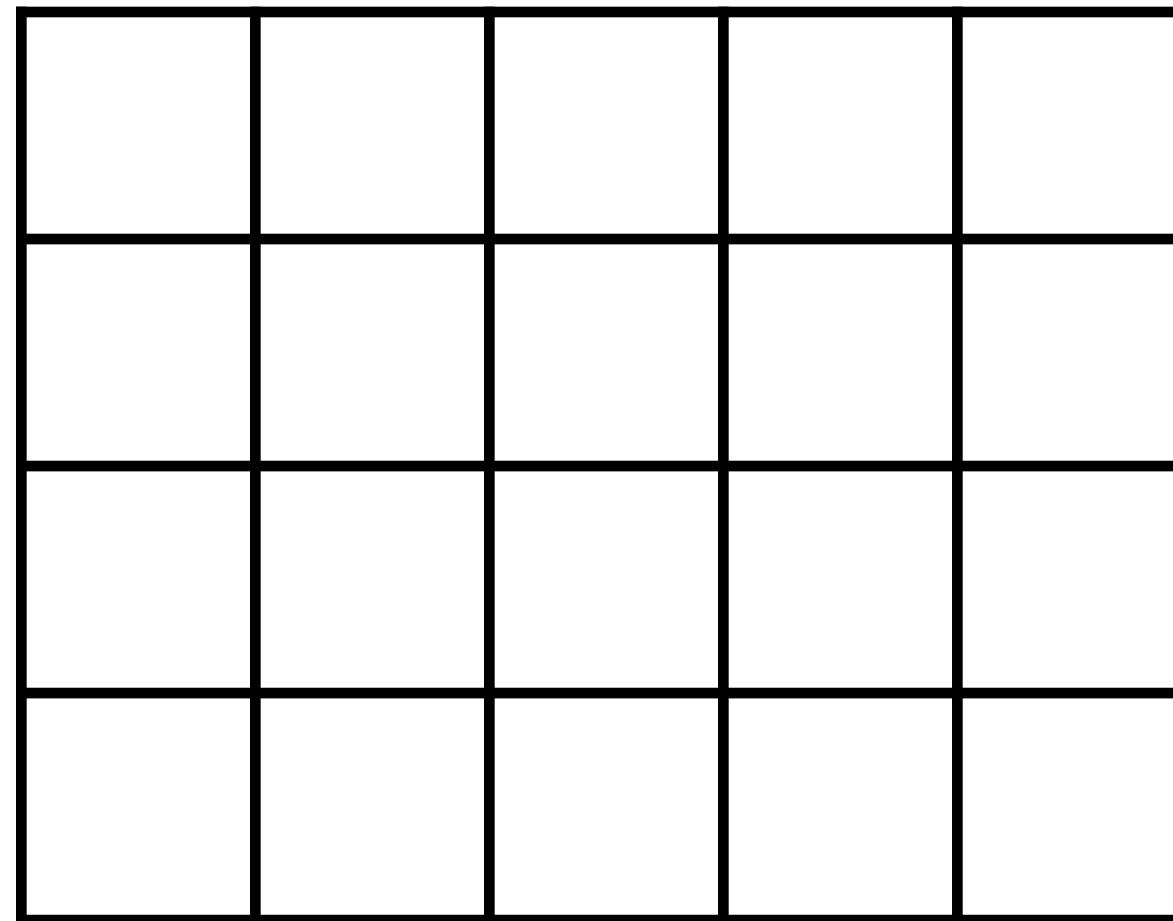


V
 $l \times n$

Truncated SVD

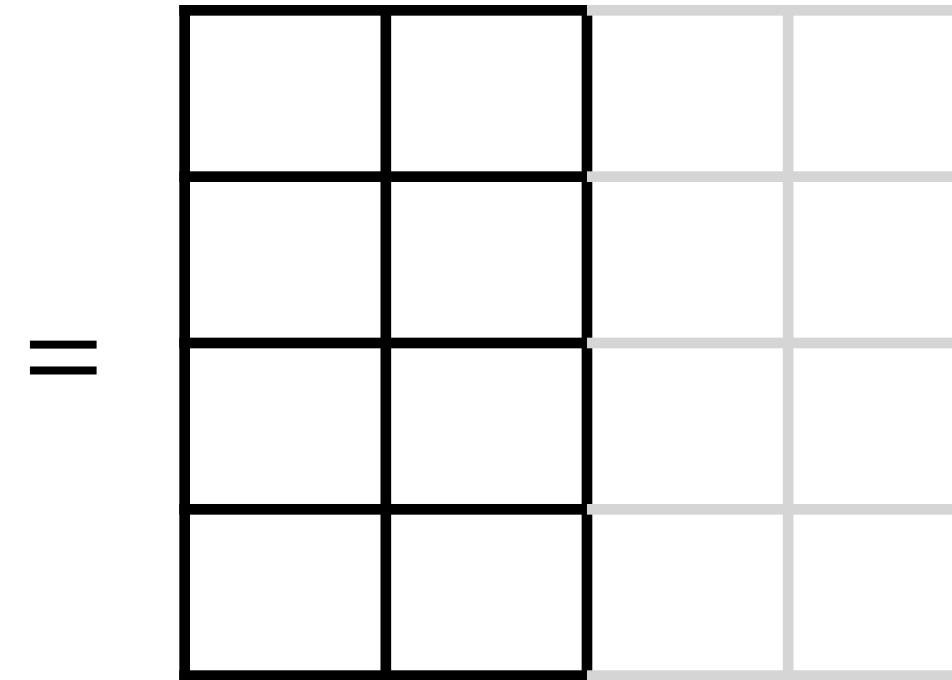
keep only first L components

"best L-rank approximation of M"

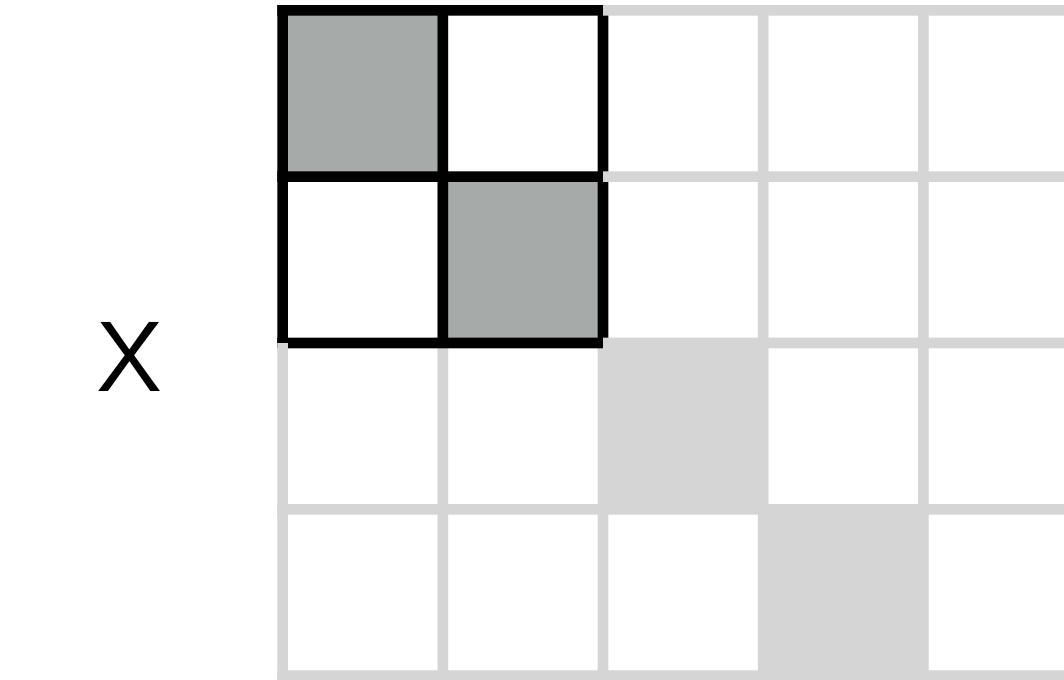


M
 $m \times n$

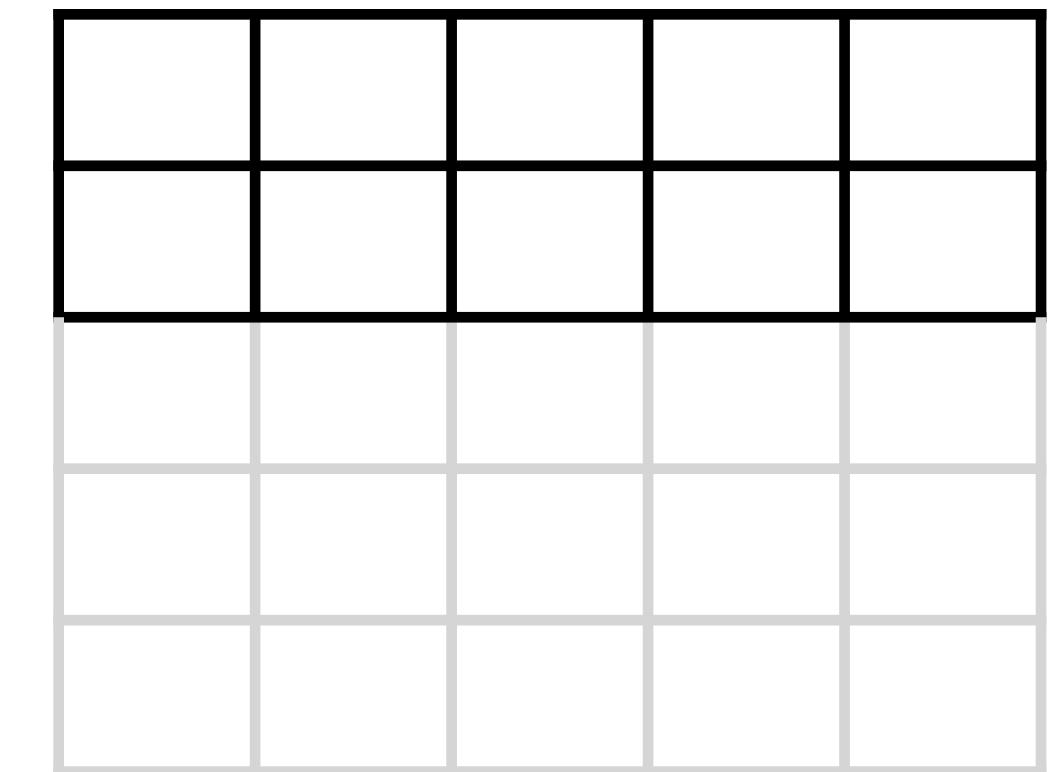
Data Matrix



U
 $m \times l$



D
 $l \times l$



V
 $l \times n$

$\|M - UDV\|^2$ as small as possible

Truncated SVD

	-0.60	-0.39	0.70	0.00
cat	-0.48	0.50	-0.12	-0.71
kitten	-0.43	-0.58	-0.69	0.00
cute	-0.48	0.50	-0.12	0.71

	cat	kitten	cute	adorable
cat	1	1	1	1
kitten	1	0	1	0
cute	1	1	0	1
adorable	1	0	1	1

	cat	kitten	cute	adorable
cat	-0.65	-0.34	-0.51	-0.34
kitten	0.02	-0.54	0.34	-0.54
cute	-0.42	0.02	0.79	0.02
adorable	-0.63	0.27	0.00	0.37

U

D

V

M

Truncated SVD

Word-Context
Matrix



	cat	kitten	cute	adorable
cat	1	1	1	1
kitten	1	0	1	0
cute	1	1	0	1
adorable	1	0	1	1

M

cat	-0.60	-0.39	0.70	0.00
kitten	-0.48	0.50	-0.12	-0.71
cute	-0.43	-0.58	-0.69	0.00
adorable	-0.48	0.50	-0.12	0.71

3.06	0.00	0.00	0.00
0.00	1.81	0.00	0.00
0.00	0.00	0.57	0.00
0.00	0.00	0.00	0.00

D

-0.65	-0.34	-0.51	-0.34
0.02	-0.54	0.34	-0.54
-0.42	0.02	0.79	0.02
-0.63	0.27	0.00	0.37
-0.04	0.73	0.00	-0.68

V

U

Truncated SVD

Word-Context
Matrix

Word Embeddings



cat kitten cute adorable

	cat	kitten	cute	adorable
cat	1	1	1	1
kitten	1	0	1	0
cute	1	1	0	1
adorable	1	0	1	1

M

cat

-0.60	-0.39	0.70	0.00
-0.48	0.50	-0.12	-0.71
-0.43	-0.58	-0.69	0.00
-0.48	0.50	-0.12	0.71

kitten

3.06	0.00	0.00	0.00
0.00	1.81	0.00	0.00
0.00	0.00	0.57	0.00
0.00	0.00	0.00	0.00

cute

-0.65	-0.34	-0.51	-0.34
0.02	-0.54	0.34	-0.54
-0.42	0.02	0.79	0.02
-0.63	0.27	0.00	0.37
-0.04	0.73	0.00	-0.68

adorable

D

U

V

Truncated SVD

Word-Context
Matrix

Word Embeddings



cat kitten cute adorable

	cat	kitten	cute	adorable
cat	1	1	1	1
kitten	1	0	1	0
cute	1	1	0	1
adorable	1	0	1	1

M New Features
("Topics")
↓

cat kitten cute adorable

-0.65	-0.34	-0.51	-0.34
0.02	-0.54	0.34	-0.54
-0.42	0.02	0.79	0.02
-0.63	0.27	0.00	0.37
-0.04	0.73	0.00	-0.68

cat	-0.60	-0.39	0.70	0.00
kitten	-0.48	0.50	-0.12	-0.71
cute	-0.43	-0.58	-0.69	0.00
adorable	-0.48	0.50	-0.12	0.71

3.06	0.00	0.00	0.00
0.00	1.81	0.00	0.00
0.00	0.00	0.57	0.00
0.00	0.00	0.00	0.00

U

D

V

Word Embeddings

What's the point?

- Computational Reasons
 - Lower dimensional = less computationally intensive
- Better Generalization
 - Lower dimensional forces abstraction: two columns that capture effectively the same information can be combined
 - Lower dimensional removes noise: throw away columns that don't improve predictive power on held-out data
- Representational Richness
 - Dimensionality reduction can capture “second order” effects
 - E.g., w1 occurs with c1, w2 occurs with c2, c1 and c2 are similar. Thus, w1 and w2 are similar.

Word Embeddings

What's the point?

***Don't count, predict!* A systematic comparison of
context-counting vs. context-predicting semantic vectors**

Marco Baroni and Georgiana Dinu and Germán Kruszewski

Center for Mind/Brain Sciences (University of Trento, Italy)

(marco.baroni|georgiana.dinu|german.kruszewski)@unitn.it

Word Embeddings

What's the point?

Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors

Marco Baroni and Georgiana Dinu and Germán Kruszewski

, Italy)
ki)@unitn.it

name	task	measure	source	soa
rg	relatedness	Pearson	Rubenstein and Goodenough (1965)	Hassan and Mihalcea (2011)
ws	relatedness	Spearman	Finkelstein et al. (2002)	Halawi et al. (2012)
wss	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
wsr	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
men	relatedness	Spearman	Bruni et al. (2013)	Bruni et al. (2013)
toefl	synonyms	accuracy	Landauer and Dumais (1997)	Bullinaria and Levy (2012)
ap	categorization	purity	Almuhareb (2006)	Rothenhäusler and Schütze (2009)
esslli	categorization	purity	Baroni et al. (2008)	Katrenko and Adriaans (2008)
battig	categorization	purity	Baroni et al. (2010)	Baroni and Lenci (2010)
up	sel pref	Spearman	Padó (2007)	Herdağdelen and Baroni (2009)
mcrae	sel pref	Spearman	McRae et al. (1998)	Baroni and Lenci (2010)
an	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)
ansyn	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013a)
ansem	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)

Word Embeddings

What's the point?

Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors

Marco Baroni and Georgiana Dinu and Germán Kruszewski

Center for Mind/Brain Sciences (University of Trento, Italy)

(marco.baroni|georgiana.dinu|german.kruszewski)@unitn.it

	rg	ws	wss	wsr	men	toefl	ap	esslli	battig	up	mcrae	an	ansyn	ansem
<i>best setup on each task</i>														
cnt	74	62	70	59	72	76	66	84	98	41	27	49	43	60
pre	84	75	80	70	80	91	75	86	99	41	28	68	71	66
<i>best setup across tasks</i>														
cnt	70	62	70	57	72	76	64	84	98	37	27	43	41	44
pre	83	73	78	68	80	86	71	77	98	41	26	67	69	64
<i>worst setup across tasks</i>														
cnt	11	16	23	4	21	49	24	43	38	-6	-10	1	0	1
pre	74	60	73	48	68	71	65	82	88	33	20	27	40	10
<i>best setup on rg</i>														
cnt	(74)	59	66	52	71	64	64	84	98	37	20	35	42	26
pre	(84)	71	76	64	79	85	72	84	98	39	25	66	70	61

Word Embeddings

What's the point?

Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors

Marco Baroni and Georgiana Dinu and Germán Kruszewski

Center for Mind/Brain Sciences (University of Trento, Italy)

(marco.baroni|georgiana.dinu|german.kruszewski)@unitn.it

	rg	ws	wss	wsr	men	toefl	ap	esslli	battig	up	mcrae	an	ansyn	ansem
<i>best setup on each task</i>														
cnt	74	62	70	59	72	76	66	84	98	41	27	49	43	60
pre	84	75	80	70	80	91	75	86	99	41	28	68	71	66
<i>best setup across tasks</i>														
cnt	70	62	70	57	72	76	64	84	98	37	27	43	41	44
pre	83	73	78	68	80	86	71	77	98	41	26	67	69	64
<i>worst setup across tasks</i>														
cnt	11	16	23	4	21	49	24	43	38	-6	-10	1	0	1
pre	74	60	73	48	68	71	65	82	88	33	20	27	40	10
<i>best setup on rg</i>														
cnt	(74)	59	66	52	71	64	64	84	98	37	20	35	42	26
pre	(84)	71	76	64	79	85	72	84	98	39	25	66	70	61



Topics

- Lecture 6 Followup: Word Embeddings from SVD
- **What is a topic model**
- Latent Semantic Analysis (LSA)
- Latent Dirichelet Allocation (LDA)
 - “Generative Stories”
 - Graphical Model Notation
 - Training and Evaluation

What is a topic model?

Example

- Imagine building a classifier to recommend articles to readers
 - Option 1: Use BOW models
 - “Since you read an article with the word ‘court-side’, you might also like this article which has the word ‘rebound’”
 - Option 2: Topics!
 - “Since you read an article about basketball, you might also like this other article about basketball”

What is a topic model?

Example

- Imagine working as a campaign strategist and analyzing public opinion in your district
 - Option 1: Use BOW models
 - “In the past year, there has been an increase in the use of the word ‘price’ and a decrease in the word ‘classroom’”
 - Option 2: Topics!
 - “In the past year, people have been talking more about the economy than about schools”

What is a topic model?

- Method for ***automatically*** organizing a collection of text
- Used for ***unlabeled*** document collections

What is a topic model?

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

Seeking Life's Bare (Genetic) Necessities

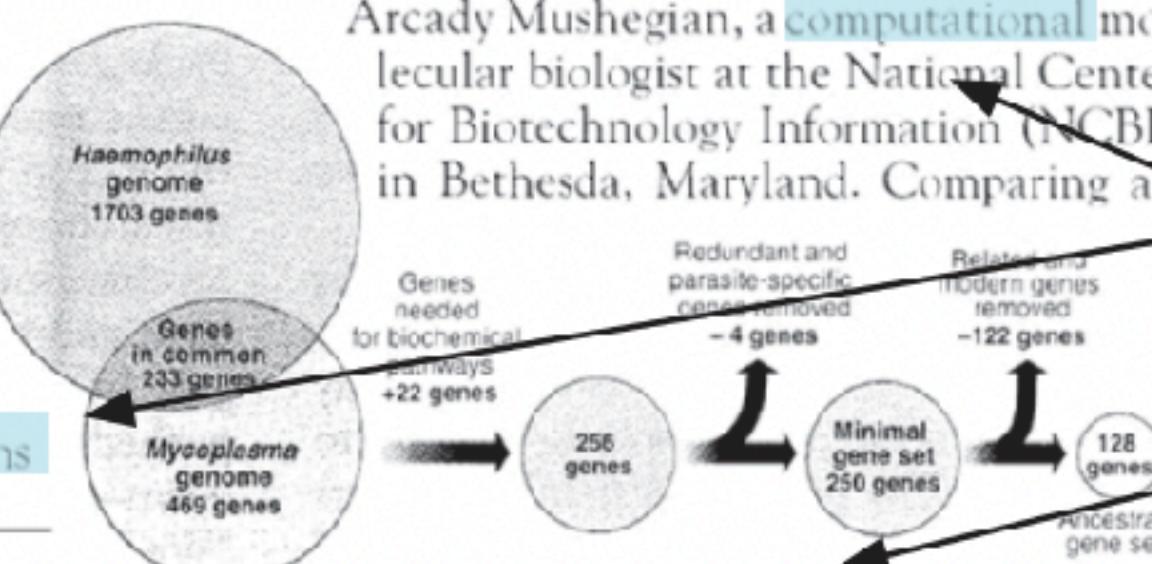
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Topic proportions and assignments

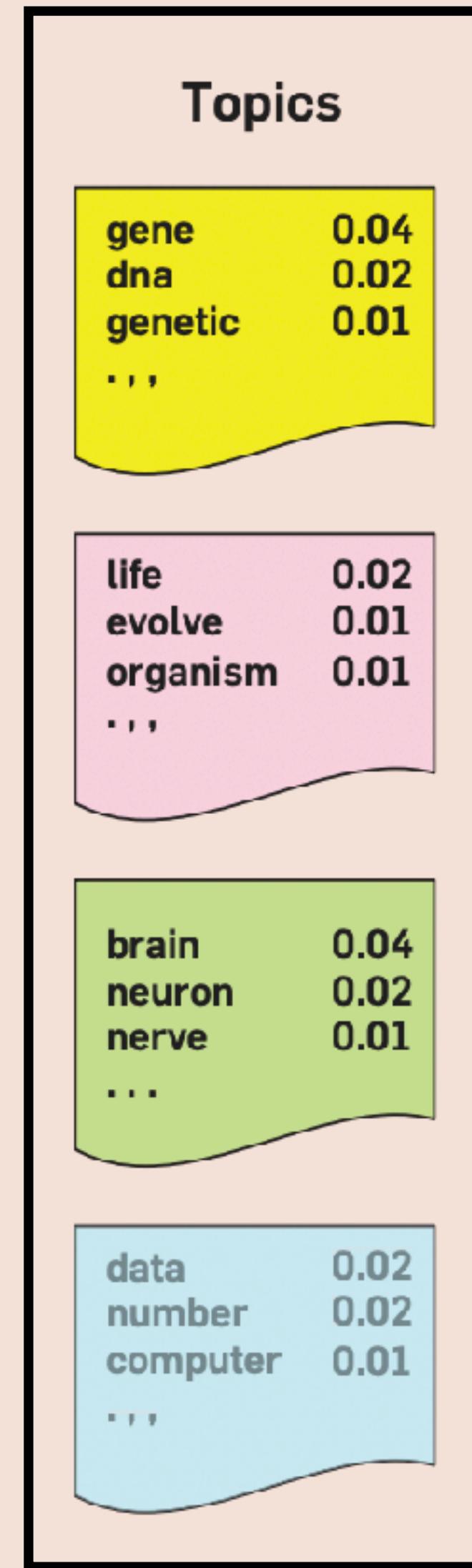
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all



What is a topic model?

Topics defined as distributions over words

Topic proportions and assignments

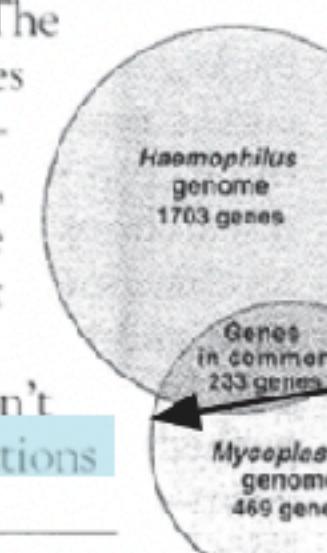


Seeking Life's Bare (Genetic) Necessities

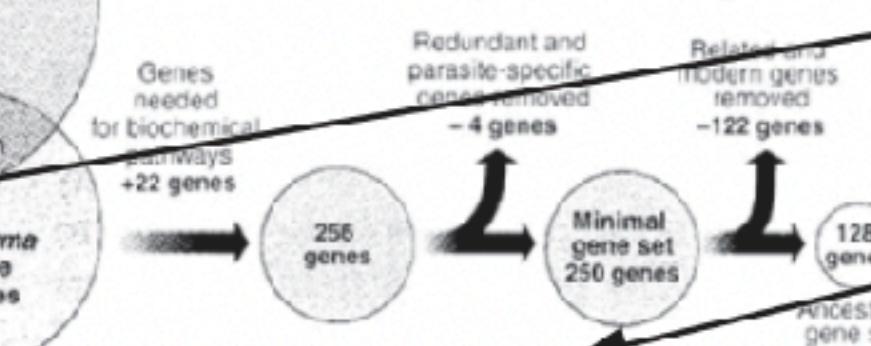
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.



* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

What is a topic model?

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

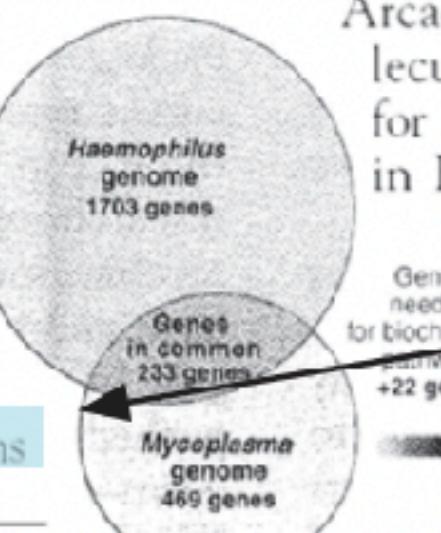
Documents explained as distributions over topics

SOURCE: Last week at a genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

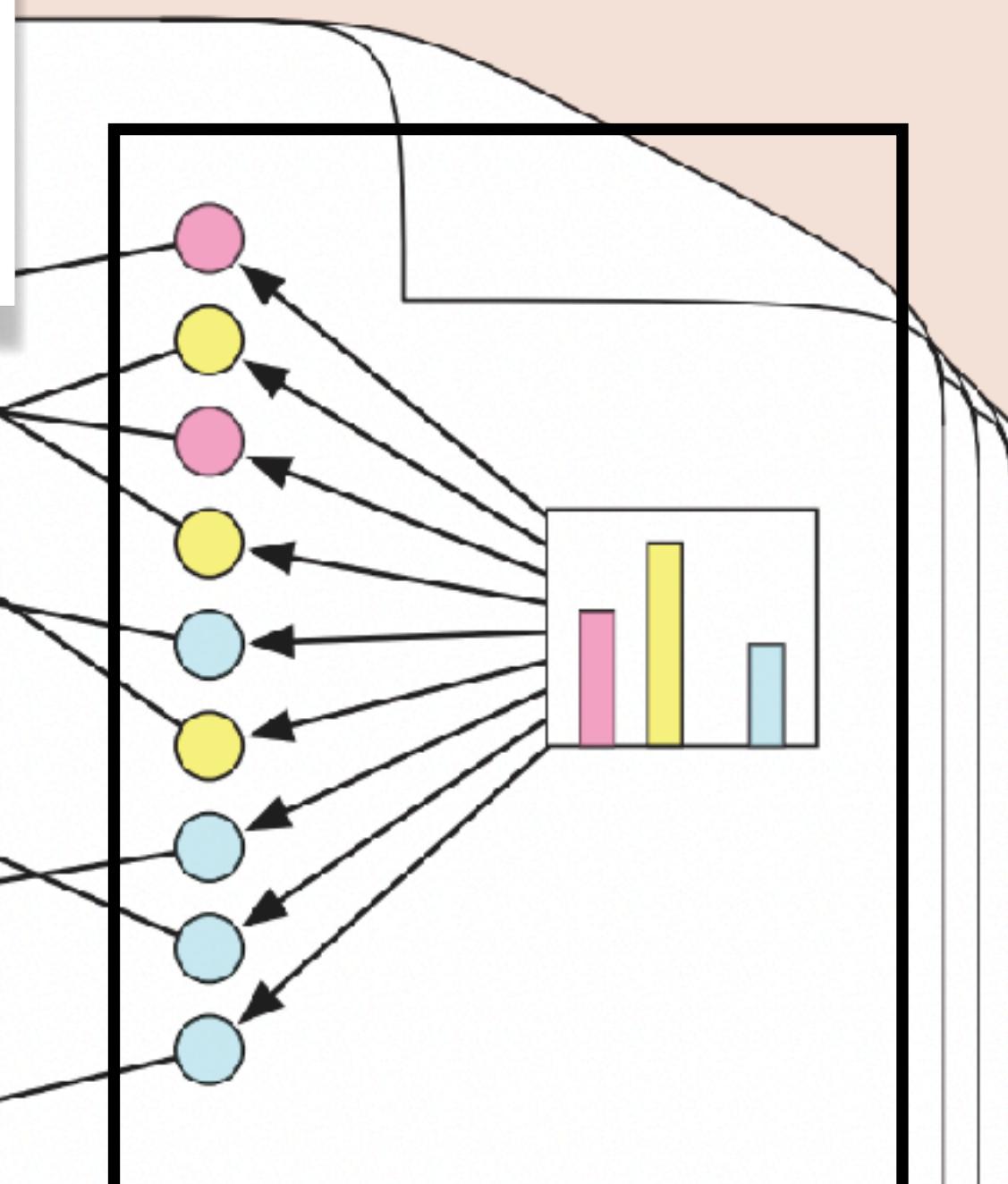
University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions and assignments

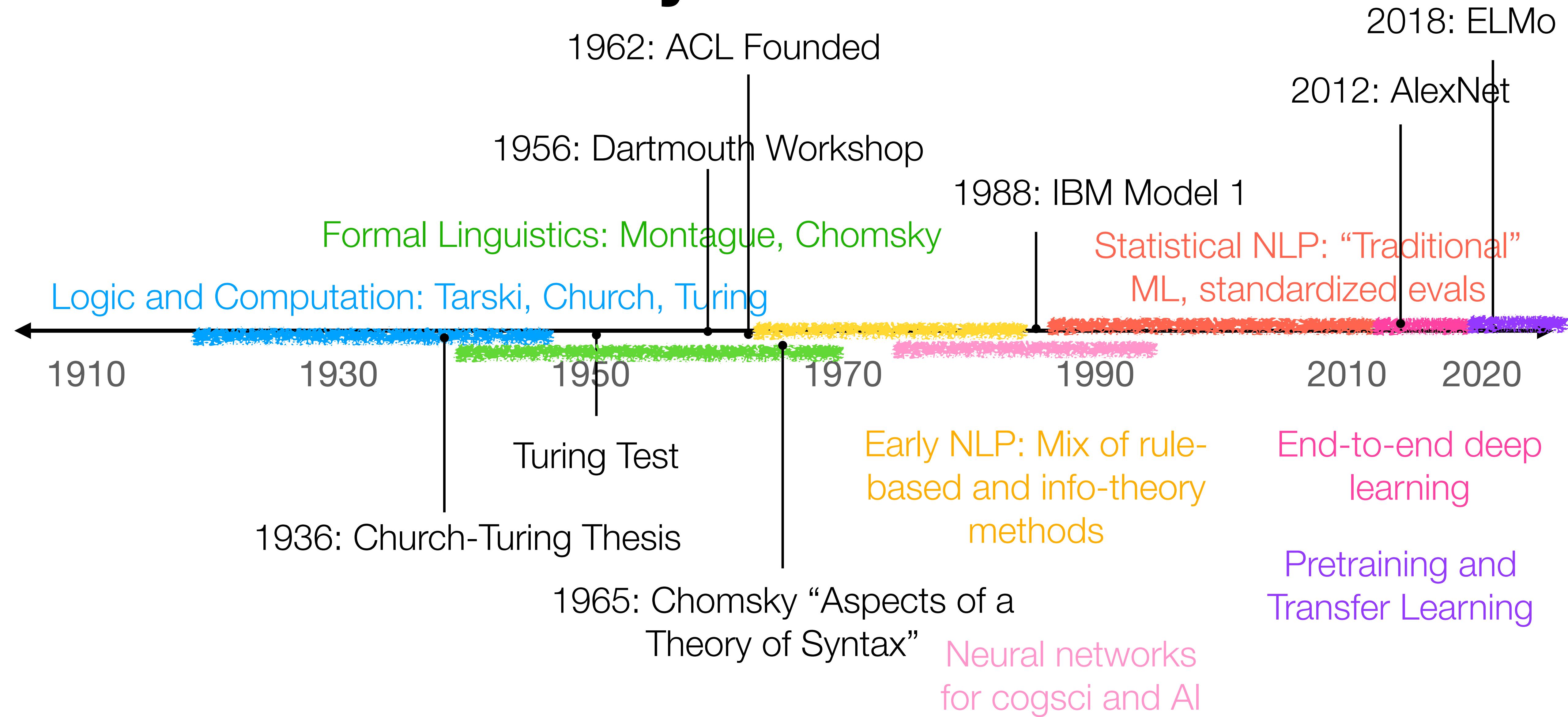


What is a topic model?

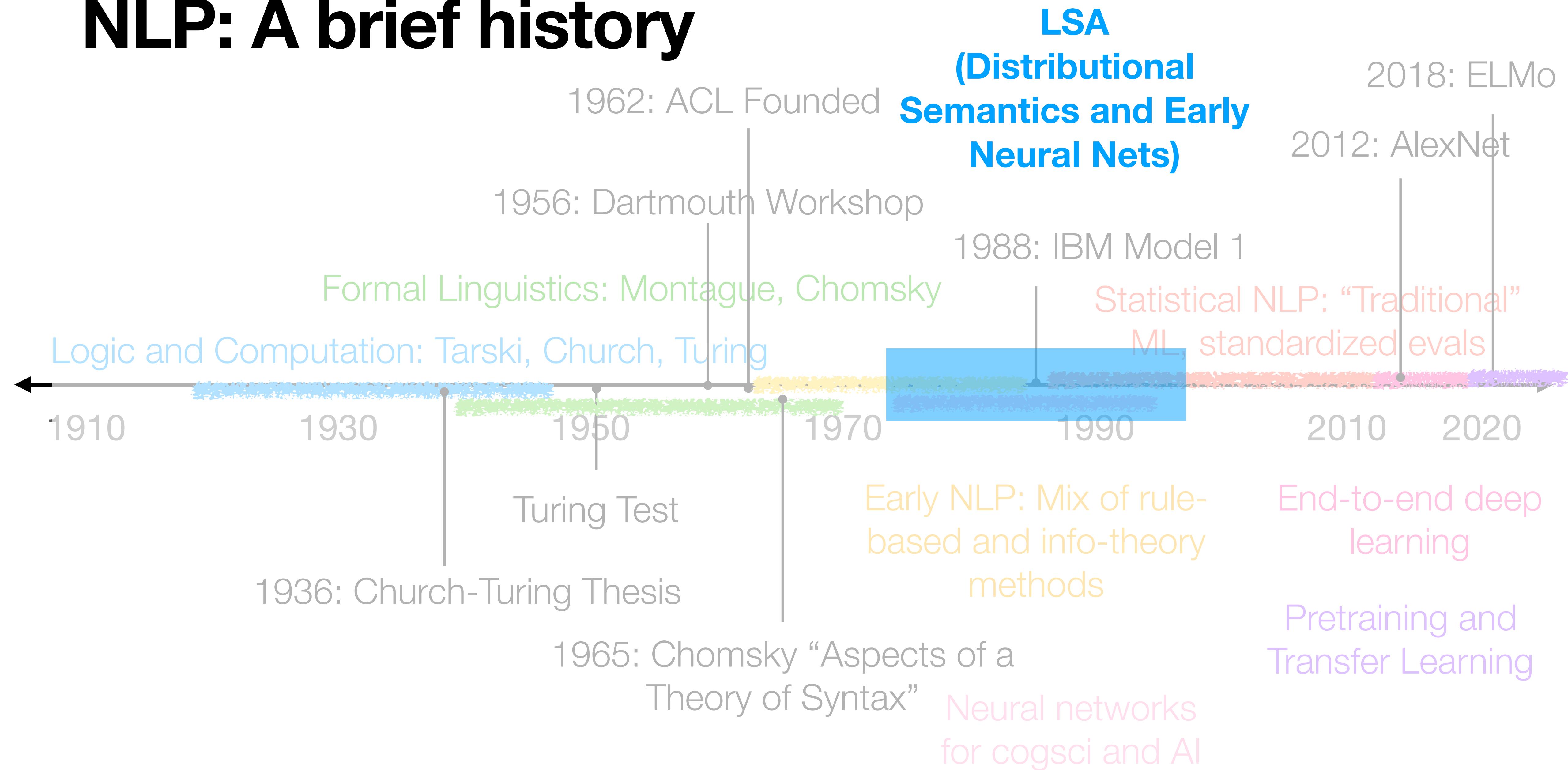
LSA vs. LDA

- Two dominant algorithms for producing topic models
 - Latent Semantic Analysis (LSA)
 - Uses dimensionality reduction/linear algebra
 - Latent Dirichelet Allocation (LDA)
 - Uses probability models/graphical models

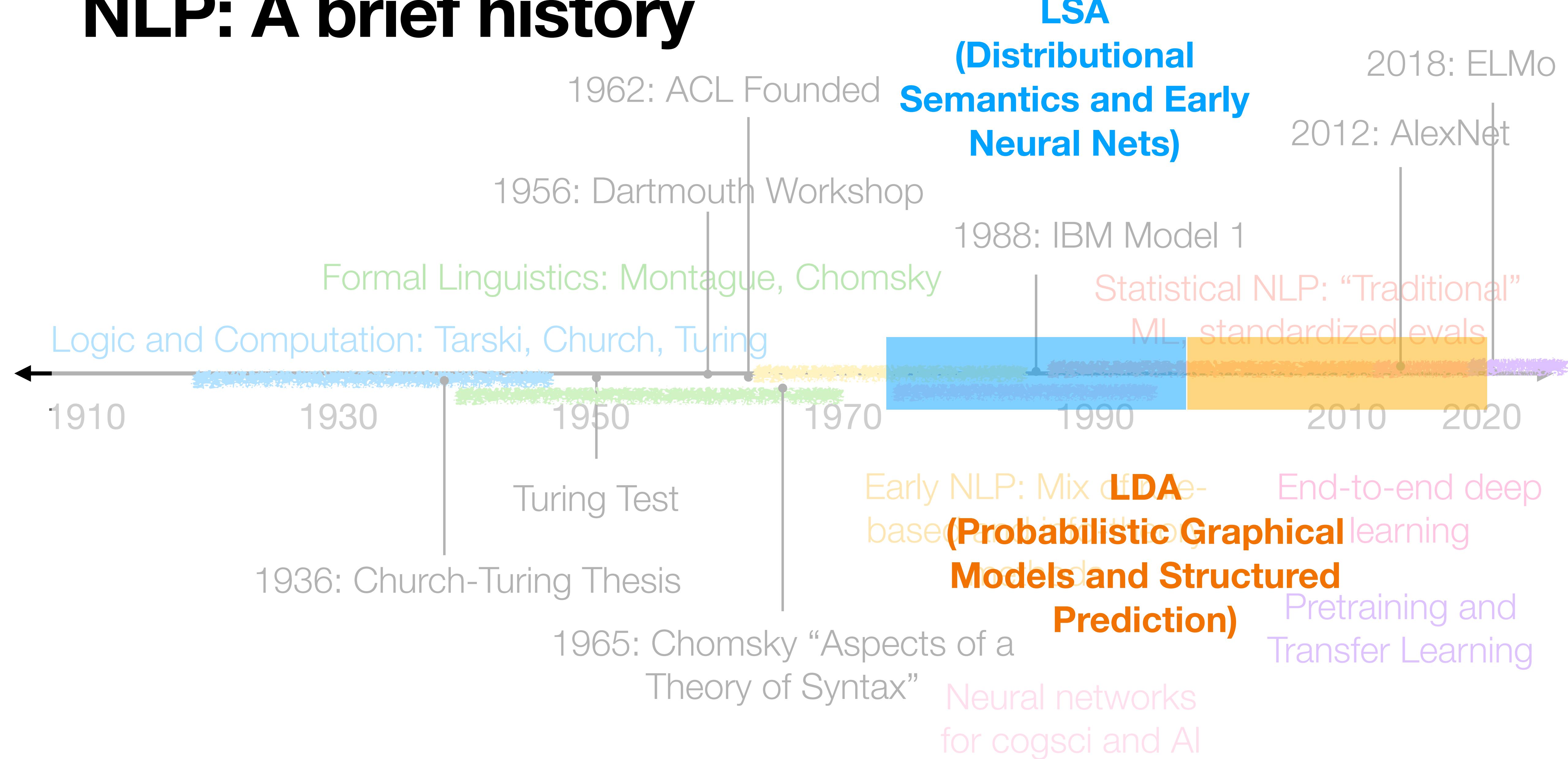
NLP: A brief history



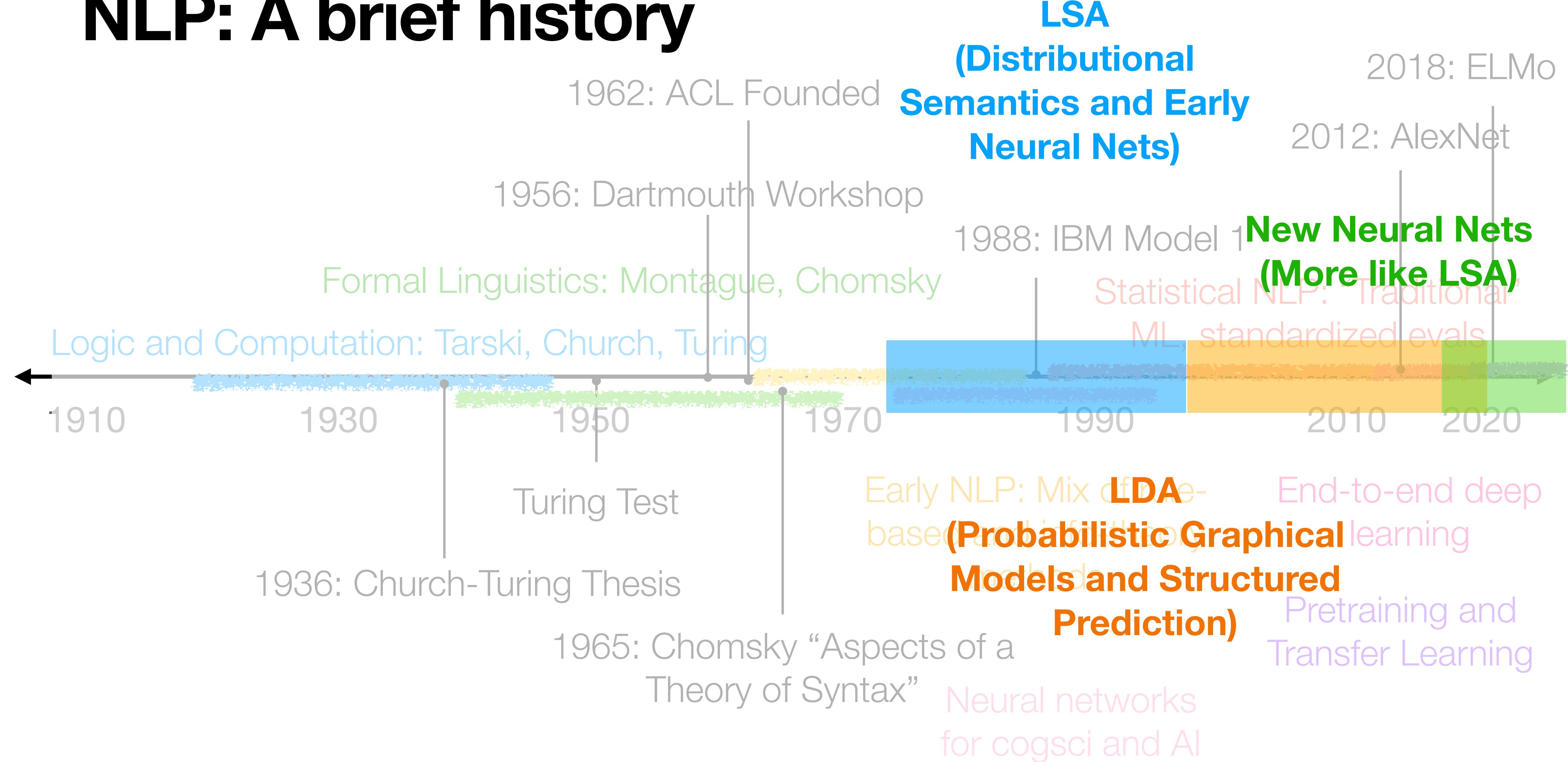
NLP: A brief history



NLP: A brief history



NLP: A brief history



Topics

- Lecture 6 Followup: Word Embeddings from SVD
- What is a topic model
- **Latent Semantic Analysis (LSA)**
- Latent Dirichelet Allocation (LDA)
 - “Generative Stories”
 - Graphical Model Notation
 - Training and Evaluation

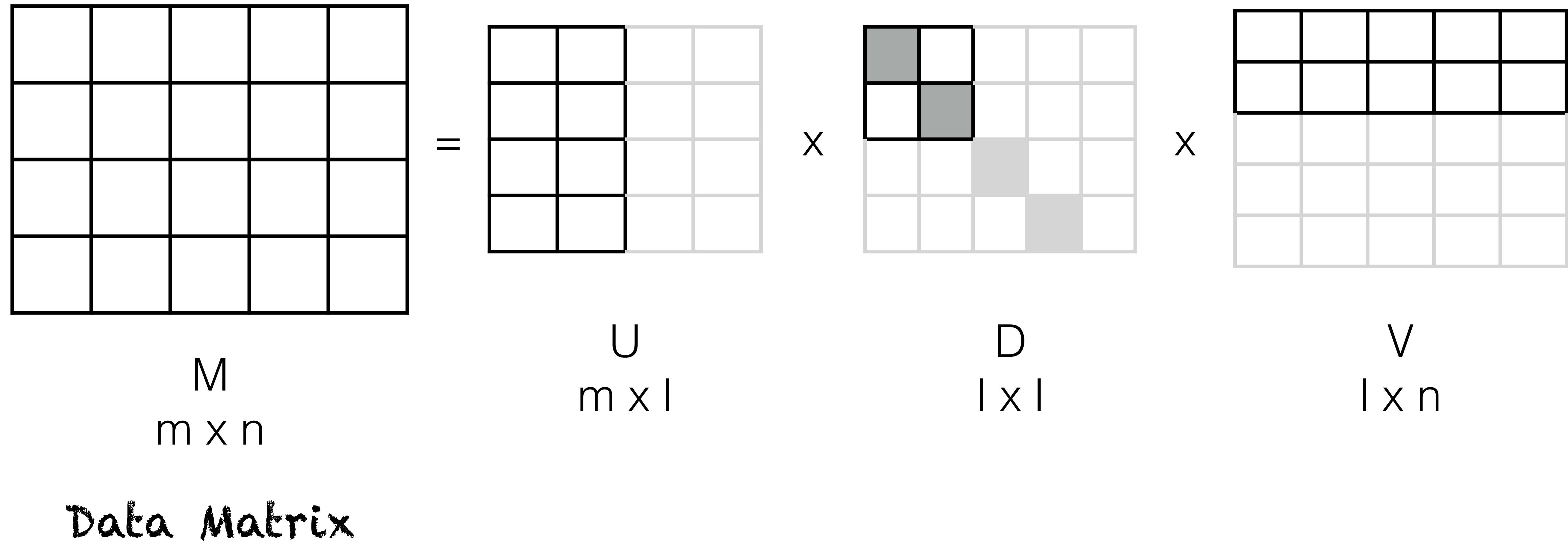
Latent Semantic Analysis (LSA)

Overview

- For a topic model, we want to find:
 - Topics = Distributions over words
 - Documents = Distributions over topics
- Dimensionality reduction gives us this via the V and U matrix, respectively

*warning: these numbers are purely made up. they may not be "to scale"

Latent Semantic Analysis (LSA)



*warning: these numbers are purely made up. they may not be "to scale"

Latent Semantic Analysis (LSA)

	cat	kitten	adorable	cute	
doc1	1	1	1	1	M
doc2	1	0	1	0	
doc3	1	1	0	1	
doc4	1	0	1	1	
U	-0.60	-0.39			V
doc1	-0.48	0.50	3.06	0.00	
doc2	-0.43	-0.58	0.00	1.81	
doc3	-0.48	0.50			
D					
doc4					

*warning: these numbers are purely made up. they may not be "to scale"

Latent Semantic Analysis (LSA)

	cat	kitten	adorable	cute
doc1	1	1	1	1
doc2	1	0	1	0
doc3	1	1	0	1
doc4	1	0	1	1

doc1	-0.60	-0.39
doc2	-0.48	0.50
doc3	-0.43	-0.58
doc4	-0.48	0.50

	3.06	0.00
	0.00	1.81

new features, i.e.,
"topics", i.e.,
weighting of
words

M

2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

D

V

*warning: these numbers are purely made up. they may not be "to scale"

Latent Semantic Analysis (LSA)

doc1 in new
feature space
(i.e., weighting of
topics)

	-0.60	-0.39
doc1	-0.48	0.50
doc2	-0.43	-0.58
doc3	-0.48	0.50

U

	cat	kitten	adorable	cute
doc1	1	1	1	1
doc2	1	0	1	0
doc3	1	1	0	1
doc4	1	0	1	1

D

3.06	0.00
0.00	1.81

M

2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

V

new features, i.e.,
"topics", i.e.,
weighting of
words

*warning: these numbers are purely made up. they may not be "to scale"

Latent Semantic Analysis (LSA)

weight of topic1
for doc1

	U	V
doc1	-0.60	-0.39
doc2	-0.48	0.50
doc3	-0.43	-0.58
doc4	-0.48	0.50

	cat	kitten	adorable	cute
doc1	1	1	1	1
doc2	1	0	1	0
doc3	1	1	0	1
doc4	1	0	1	1

3.06	0.00
0.00	1.81

M

2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

D

V

*warning: these numbers are purely made up. they may not be "to scale"

Latent Semantic Analysis (LSA)

weight of topic1
for doc1

	U	V
doc1	-0.60	-0.39
doc2	-0.48	0.50
doc3	-0.43	-0.58
doc4	-0.48	0.50

	cat	kitten	adorable	cute
doc1	1	1	1	1
doc2	1	0	1	0
doc3	1	1	0	1
doc4	1	0	1	1

3.06	0.00
0.00	1.81

weight of
word "cat" in
topic1

M

2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

D

V

Latent Semantic Analysis (LSA Example)

- Imagine building a classifier to recommend articles to readers
 - Option 1: Use BOW models
 - “Since you read an article with the word ‘court-side’, you might also like this article which has the word ‘rebound’”
 - Option 2: Topics!
 - “Since you read an article about basketball, you might also like this other article about basketball”

doc1	-0.60	-0.39
doc2	-0.48	0.50
doc3	-0.43	-0.58
doc4	-0.48	0.50

U

2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

V

Latent Semantic Analysis (LSA) Example

- Imagine building a classifier to recommend articles to readers
 - Option 1: Use BOW model
 - “Since you read an article about ‘court-side’, you might also like this other article which has the word ‘rebound’”
 - Option 2: Topics!

Use embeddings as
input to your
classifier.

doc1
doc2
doc3
doc4

-0.60	-0.39
-0.48	0.50
-0.43	-0.58
-0.48	0.50

U

2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

V

Latent Semantic Analysis (LSA)

Example

- Imagine building a classifier to recommend articles to readers
 - Option 1: Use BOW model
 - “Since you read an article about ‘court-side’, you might like another article which has the word ‘rebound’”
 - Option 2: Topics!
 - “Since you read an article about basketball, you might also like this other article about basketball”

Refer to topics to understand/explain predictions

doc1
doc2
doc3
doc4

-0.60	-0.39
-0.48	0.50
-0.43	-0.58
-0.48	0.50

U

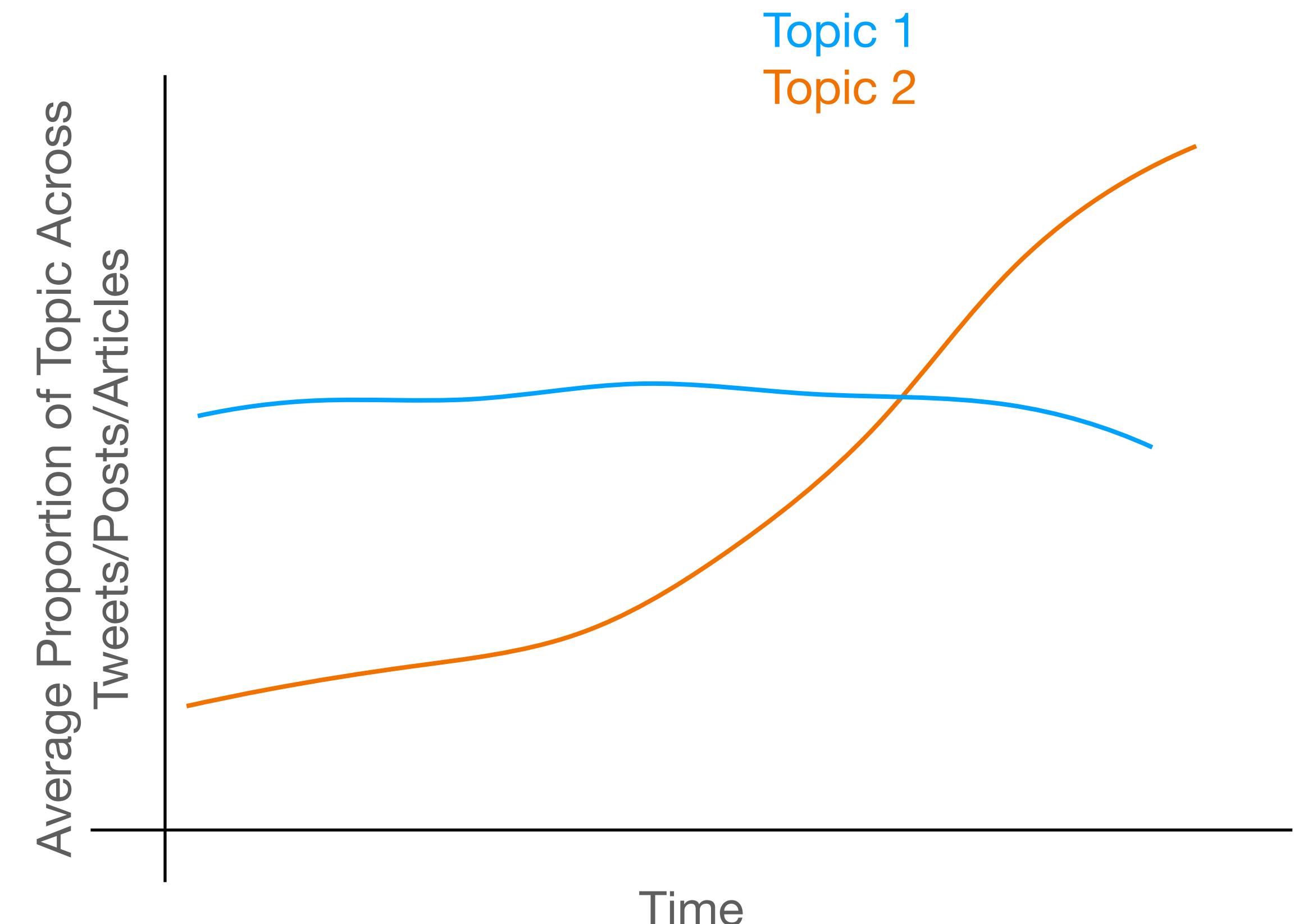
2.5	0.3
3.4	0.1
0.6	1.5
0.9	4.4

V

What is a topic model?

Example

- Imagine working as a campaign strategist and analyzing public opinion in your district
 - Option 1: Use BOW models
 - “In the past year, there has been an increase in the use of the word ‘price’ and a decrease in the word ‘classroom’”
 - Option 2: Topics!
 - “In the past year, people have been talking more about the economy than about schools”





Topics

- Lecture 6 Followup: Word Embeddings from SVD
- What is a topic model
- Latent Semantic Analysis (LSA)
- **Latent Dirichelet Allocation (LDA)**
 - “Generative Stories”
 - Graphical Model Notation
 - Training and Evaluation

Latent Dirichlet Allocation (LDA)

Overview

- Probabilistic model for finding topic models (in contrast to LSA has no notion of probability)
- We are covering LDA today because:
 - Its a good, popular method for training topic models!
 - It introduces important ideas about **generative models** which are used elsewhere in NLP
 - It introduces the notion of “**graphical models**” which are used elsewhere in NLP

Latent Dirichlet Allocation (LDA)

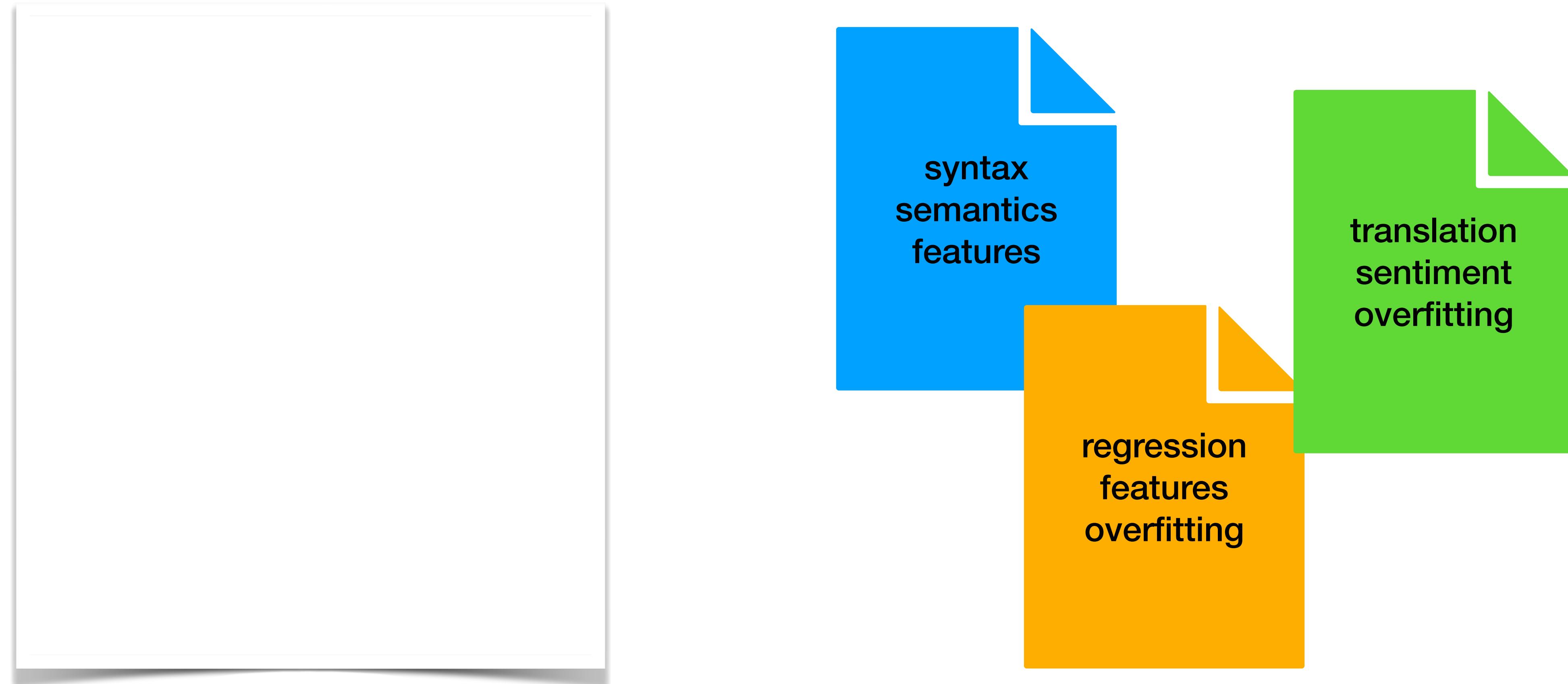
Generative Story

- Step 1 of building models is to make assumptions about the structure of the data/problem
- In many* NLP models, these assumptions are given by the “generative story”
- Story for how the observed data came to be; provides intuition behind the set of assumptions that are then encoded in the model
- We will return to this concept (Machine Translation, Sequence Tagging, ...)

*Specifically, probabilistic generative models

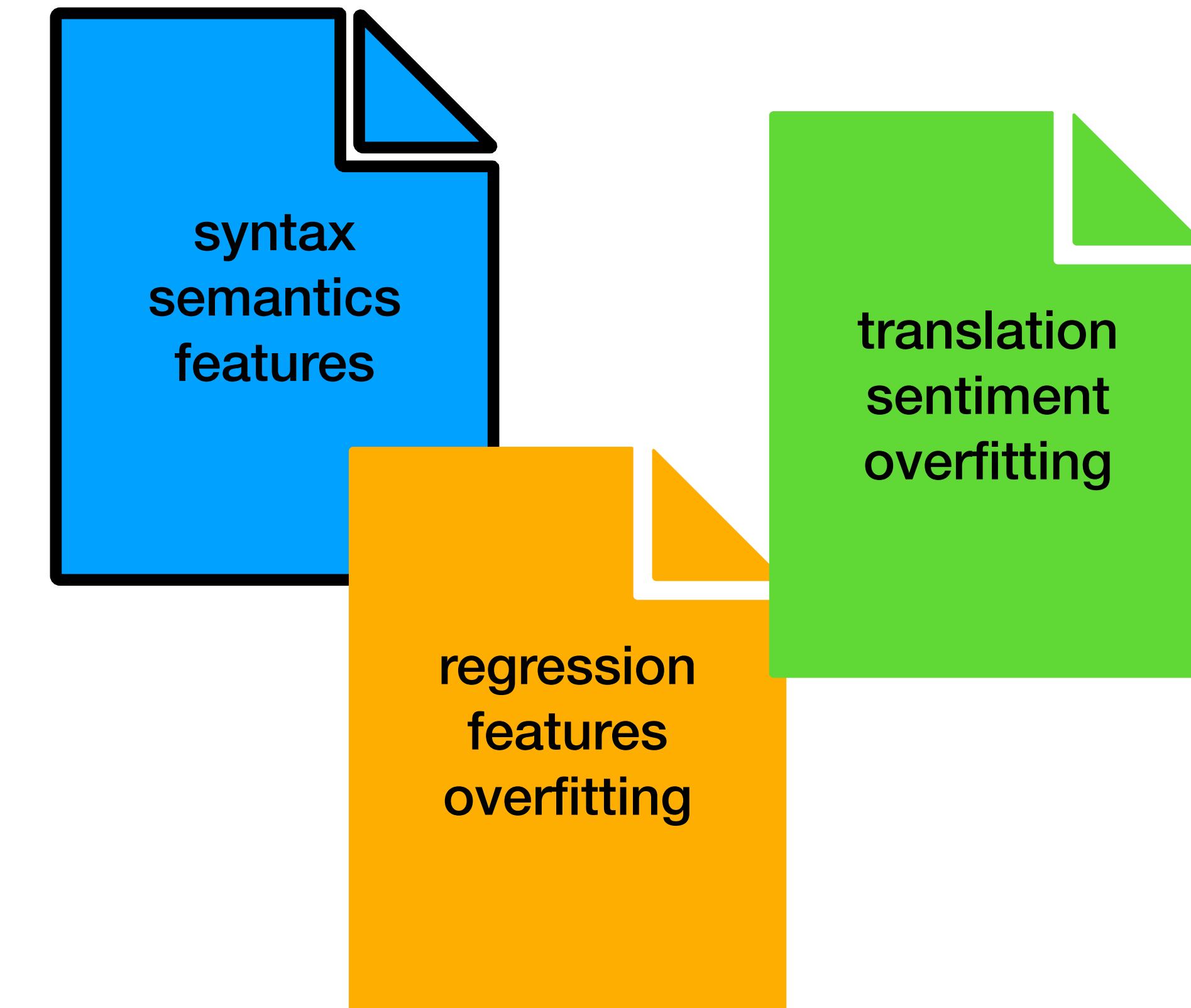
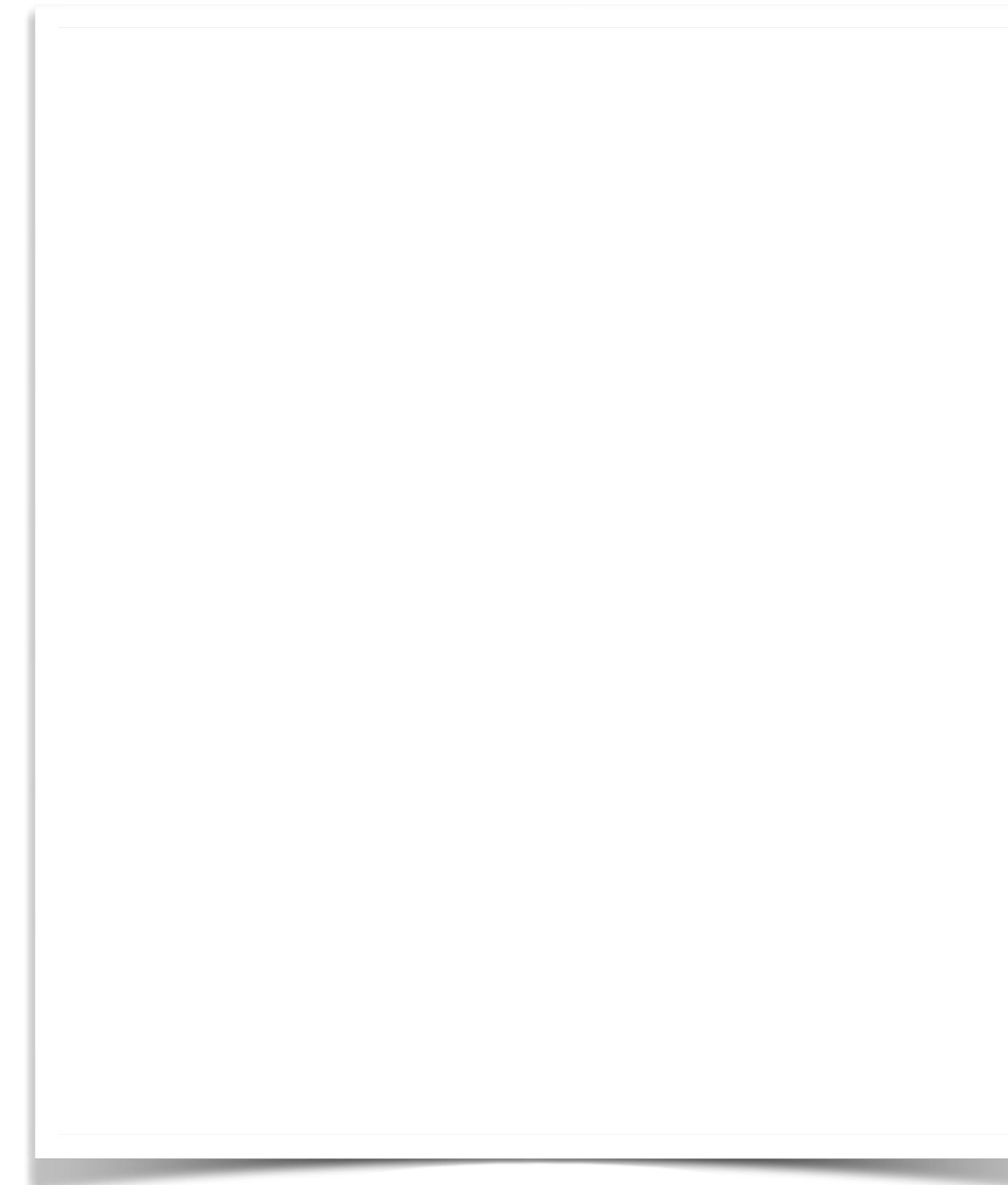
Latent Dirichlet Allocation (LDA)

Generative Story



Latent Dirichlet Allocation (LDA)

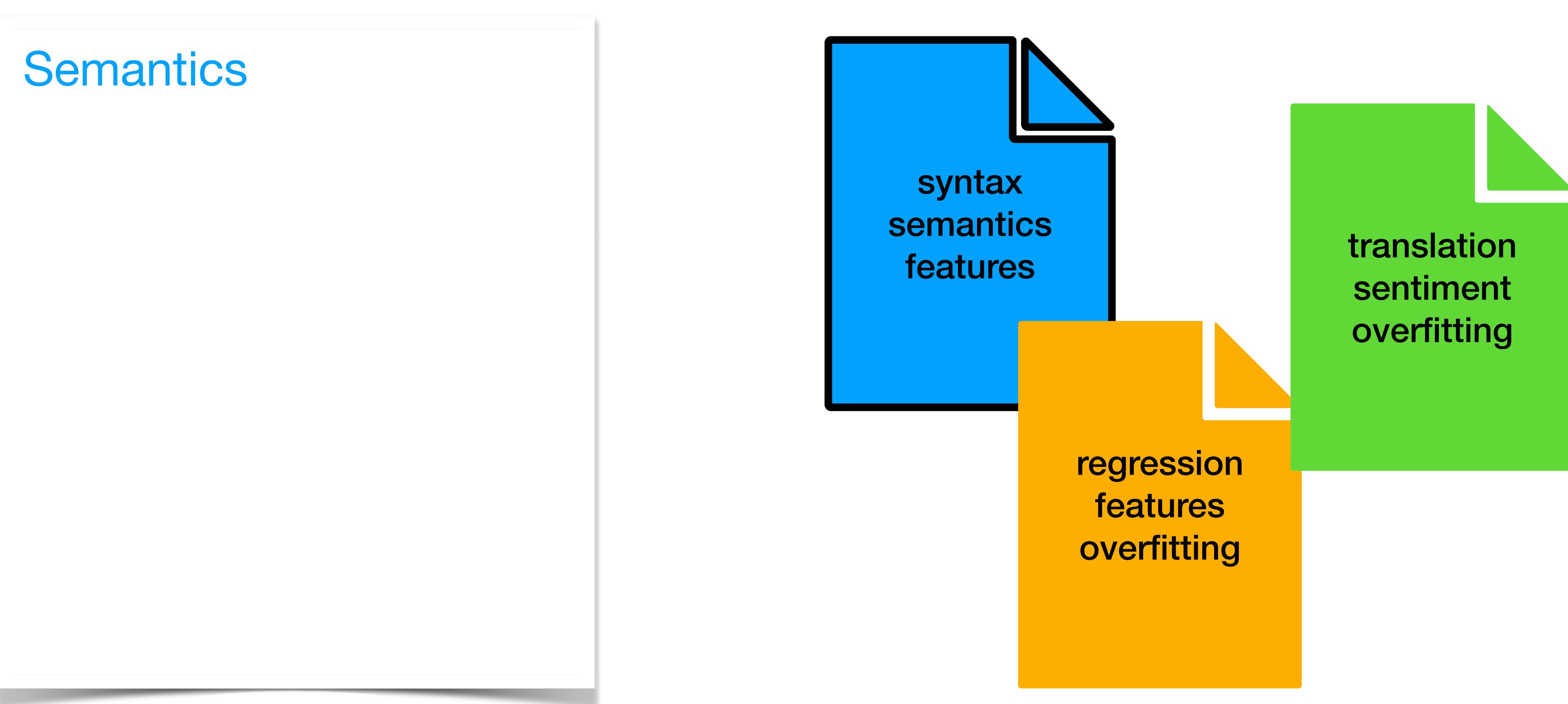
Generative Story



1. Sample a topic

Latent Dirichlet Allocation (LDA)

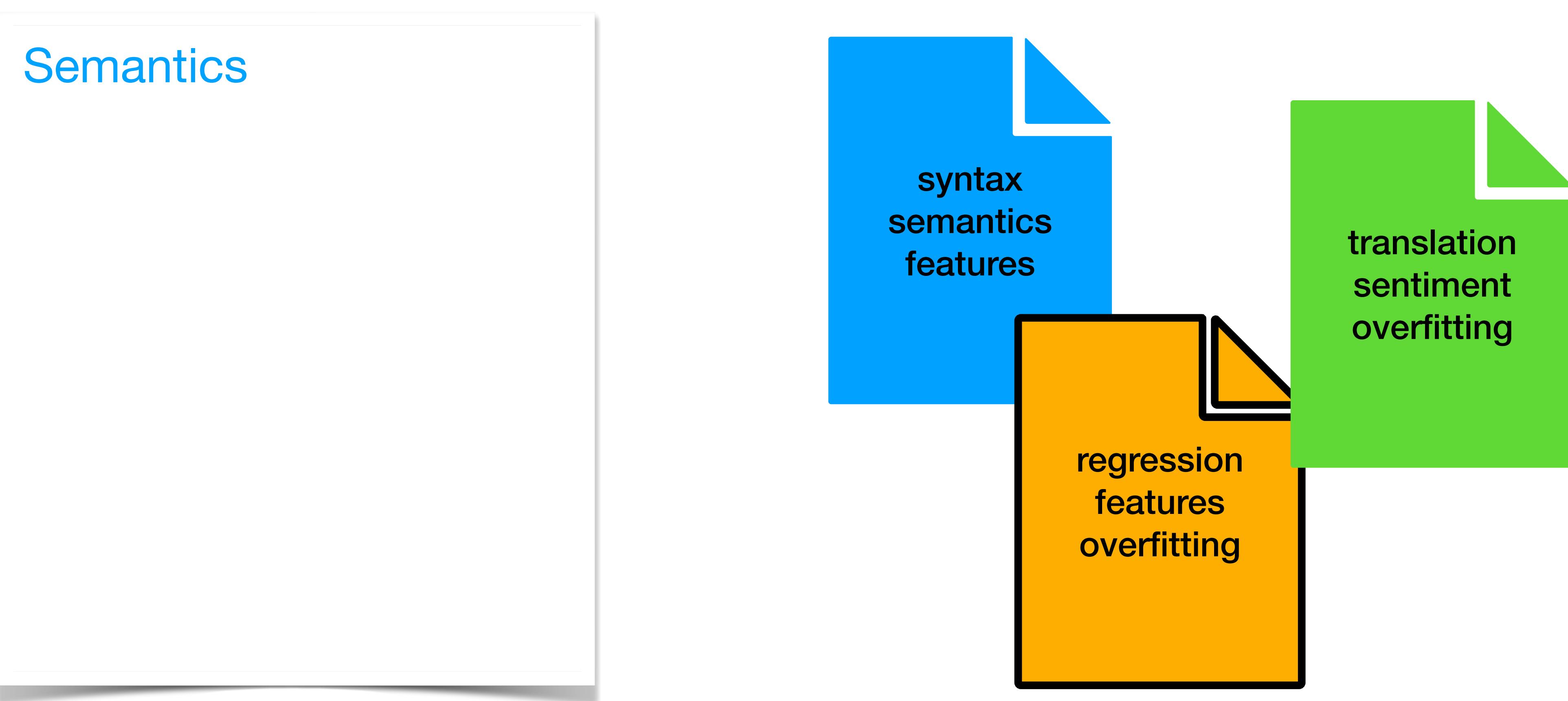
Generative Story



2. Sample a word from that topic

Latent Dirichlet Allocation (LDA)

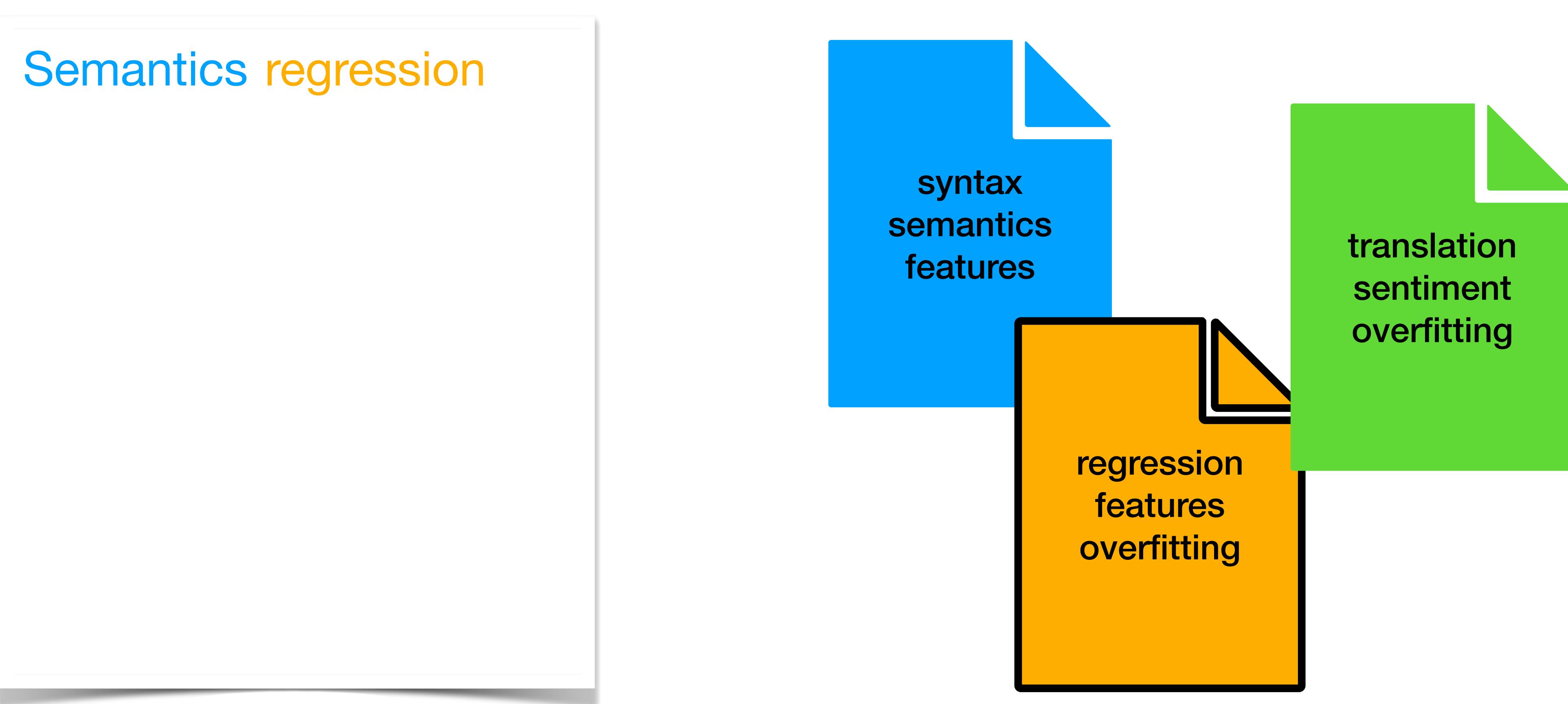
Generative Story



1. Sample a topic

Latent Dirichlet Allocation (LDA)

Generative Story

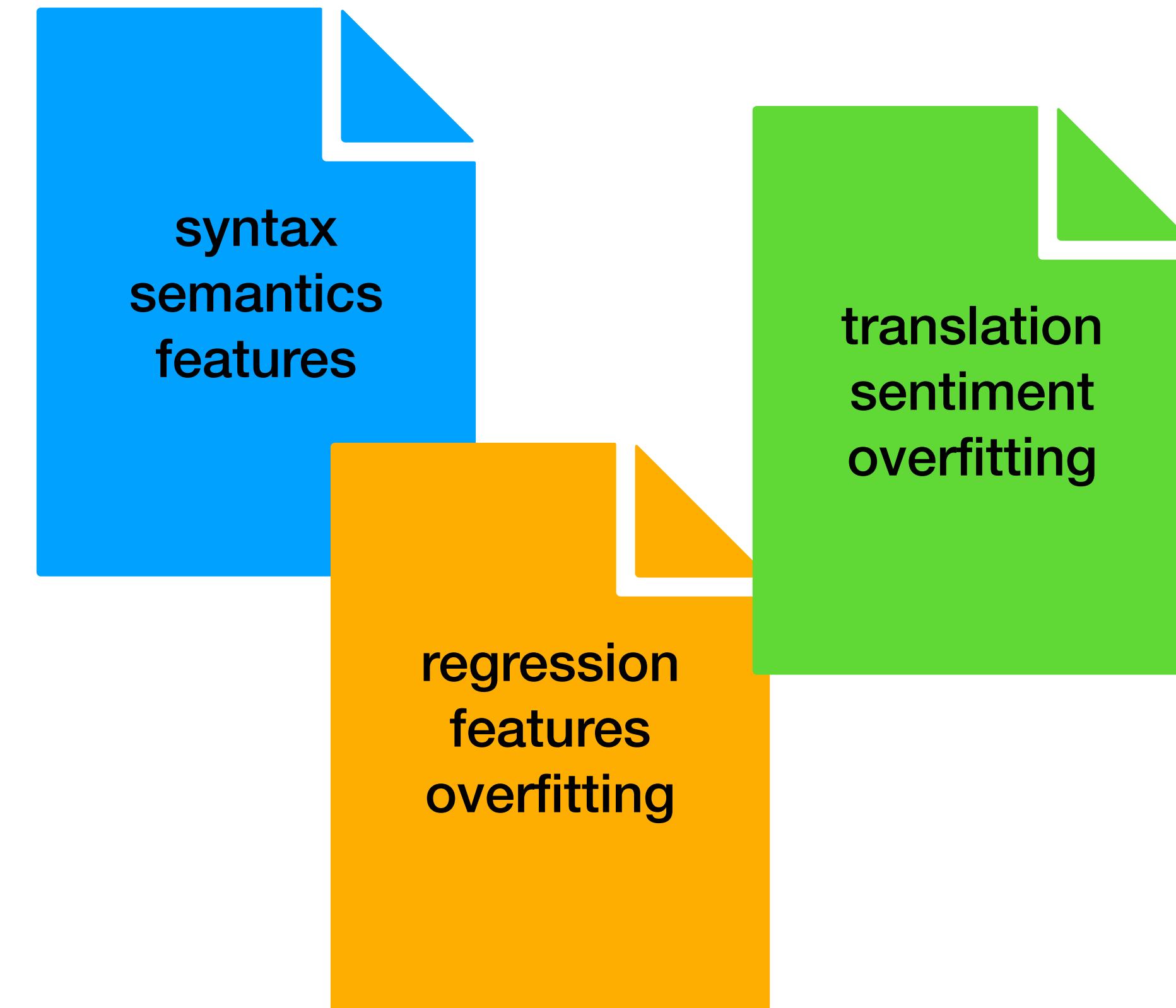


2. Sample a word from that topic

Latent Dirichlet Allocation (LDA)

Generative Story

Semantics regression
models features
evaluation difficult
overfitting parameters
loss prediction
errors multilingual
morphology



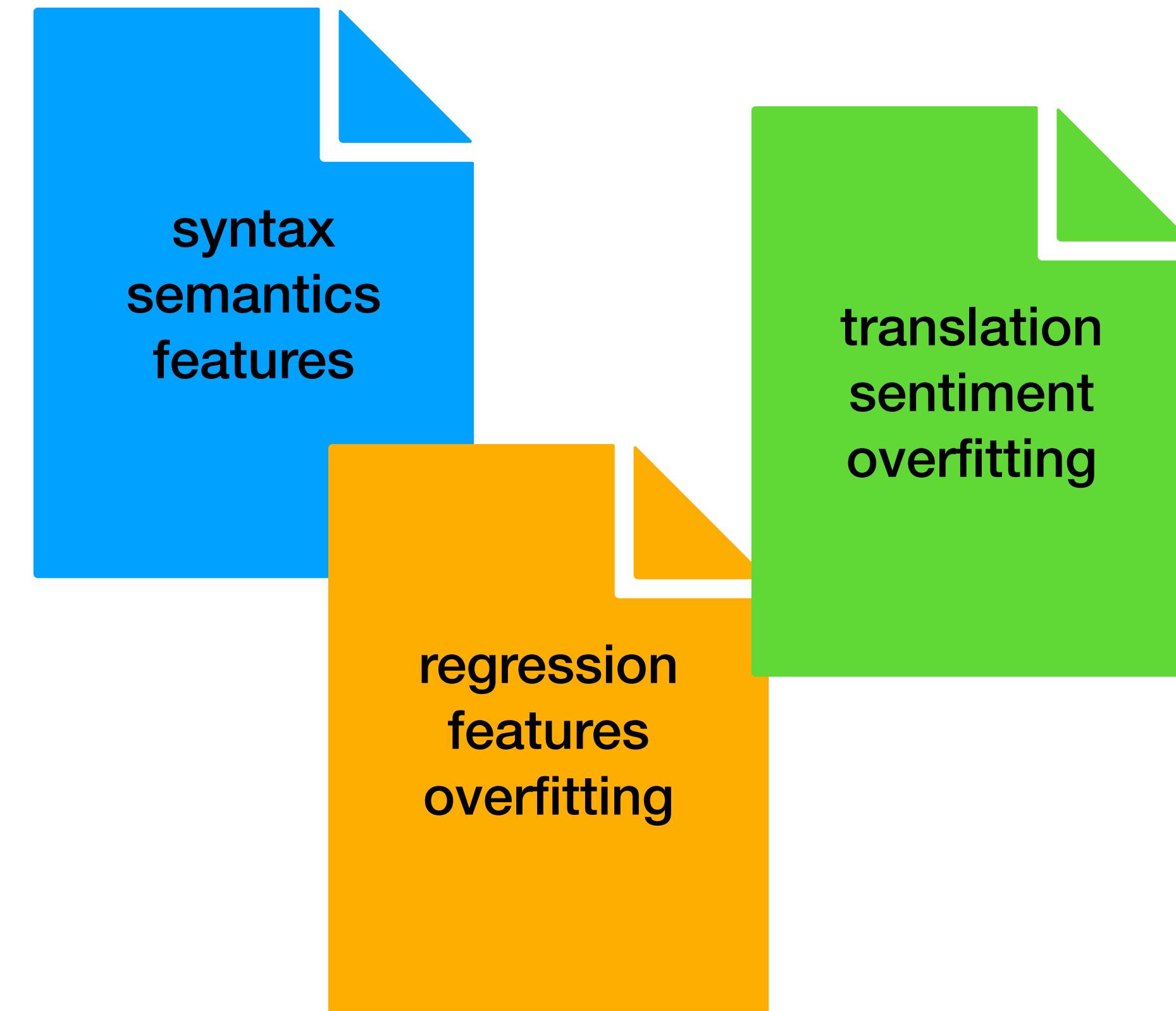
Repeat

Latent Dirichlet Allocation (LDA)

Generative Story

Still like BOW. Ignores issues of syntax/
word order, semantics, discourse, etc...

Semantics regression
models features
evaluation difficult
overfitting parameters
loss prediction
errors multilingual
morphology



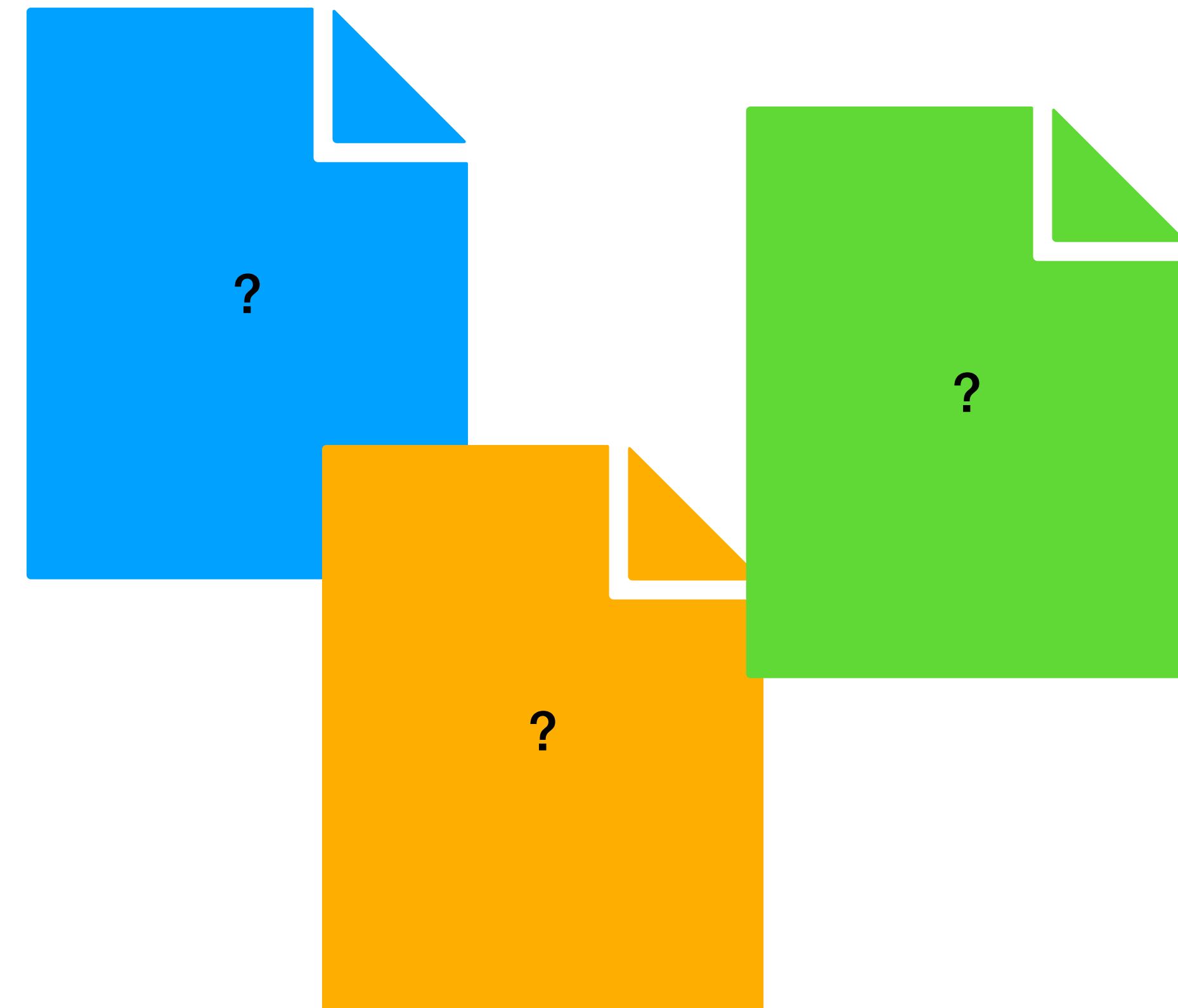
Repeat

Latent Dirichlet Allocation (LDA)

Generative Story

When training a model we get only this

Semantics regression
models features
evaluation difficult
overfitting parameters
loss prediction
errors multilingual
morphology

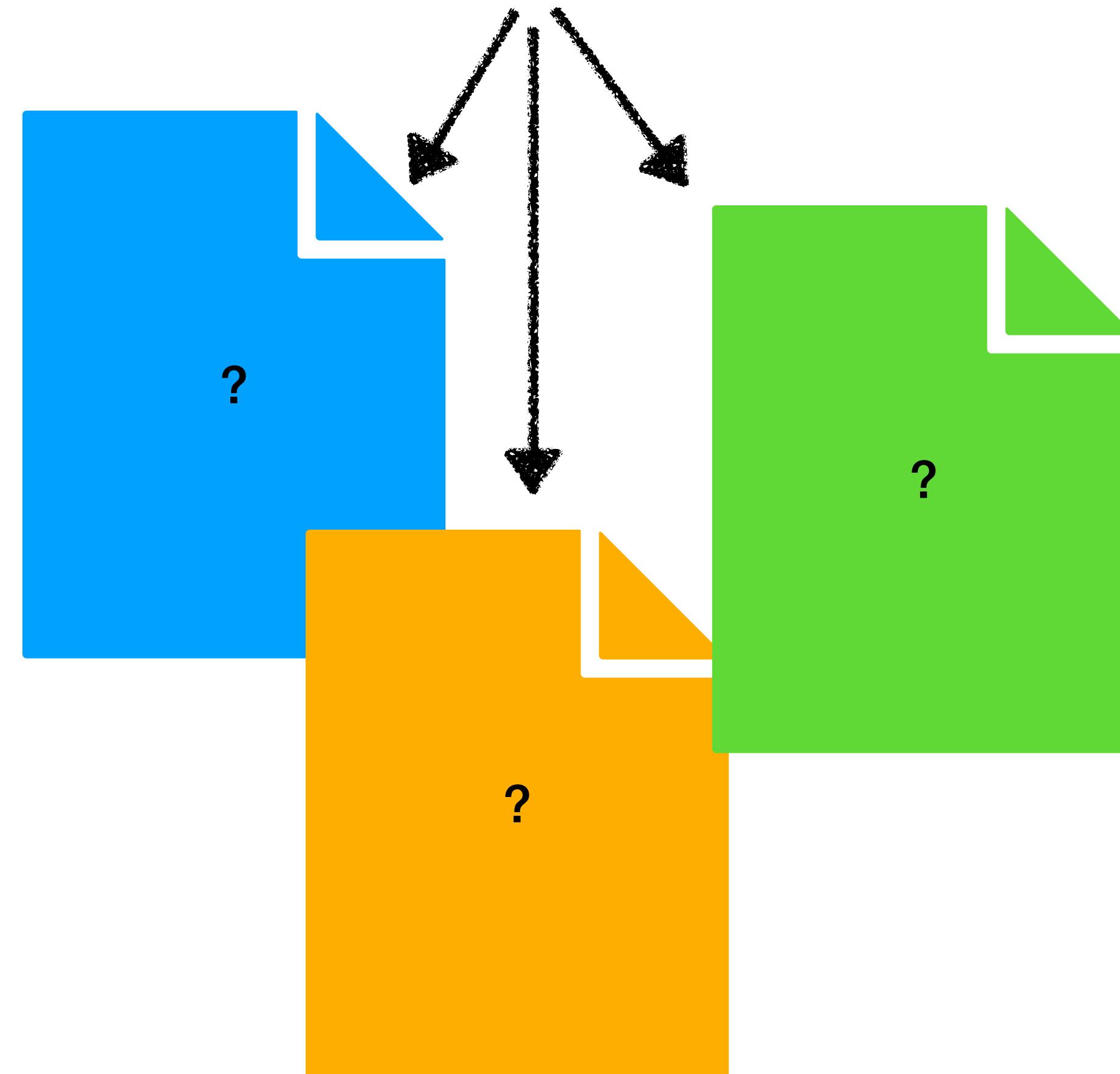


Latent Dirichlet Allocation (LDA)

Generative Story

Semantics regression
models features
evaluation difficult
overfitting parameters
loss prediction
errors multilingual
morphology

Want to infer this



Latent Dirichlet Allocation (LDA)

Generative Story

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Generative Story

Probability of the data
(a given word appearing)

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Generative Story

Probability of that word for a given topic

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Generative Story

Overall probability of that topic

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Generative Story

Marginalized over all topics

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Generative Story

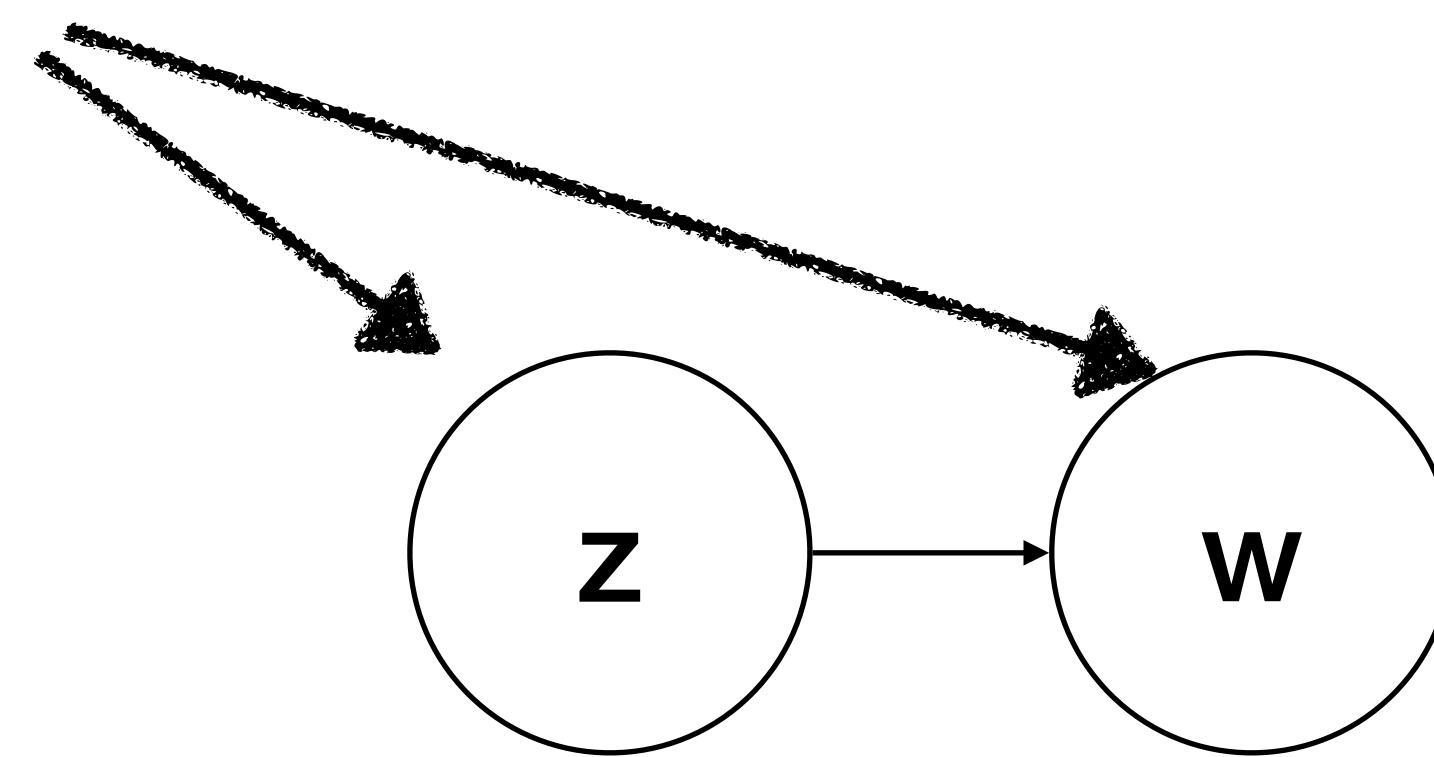
$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Given the observed w_s ,
find the $P(z)$ s and $P(w|z)$ s that
make the observed w_s most likely

Latent Dirichlet Allocation (LDA)

Graphical Model Notation

Random variables

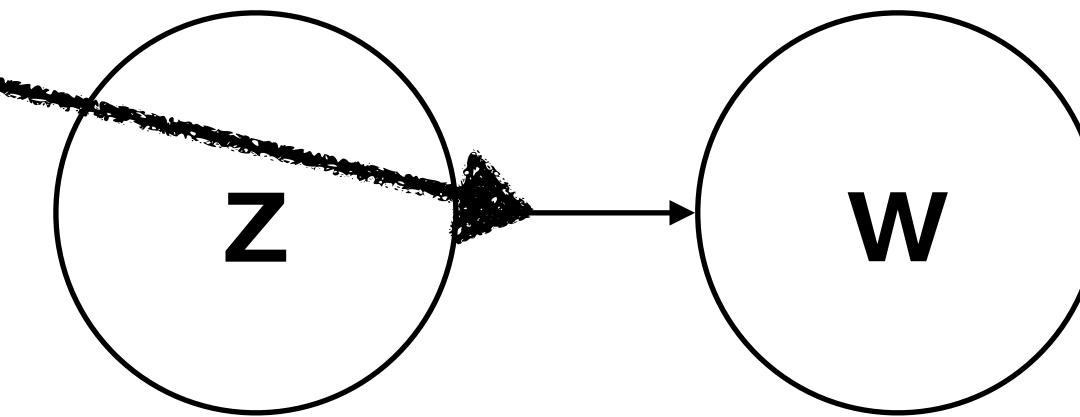


$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Graphical Model Notation

Random variables
in a dependence
relationship

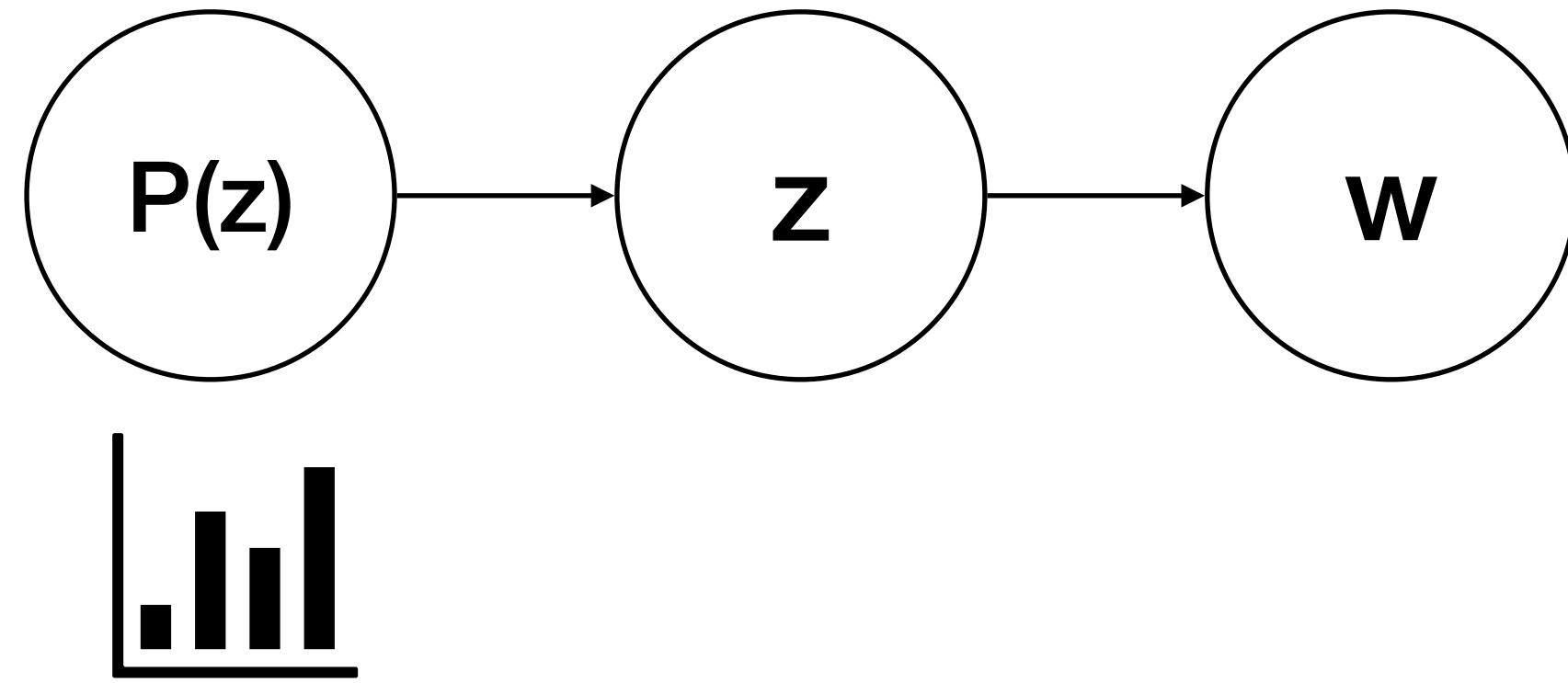


$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Graphical Model Notation

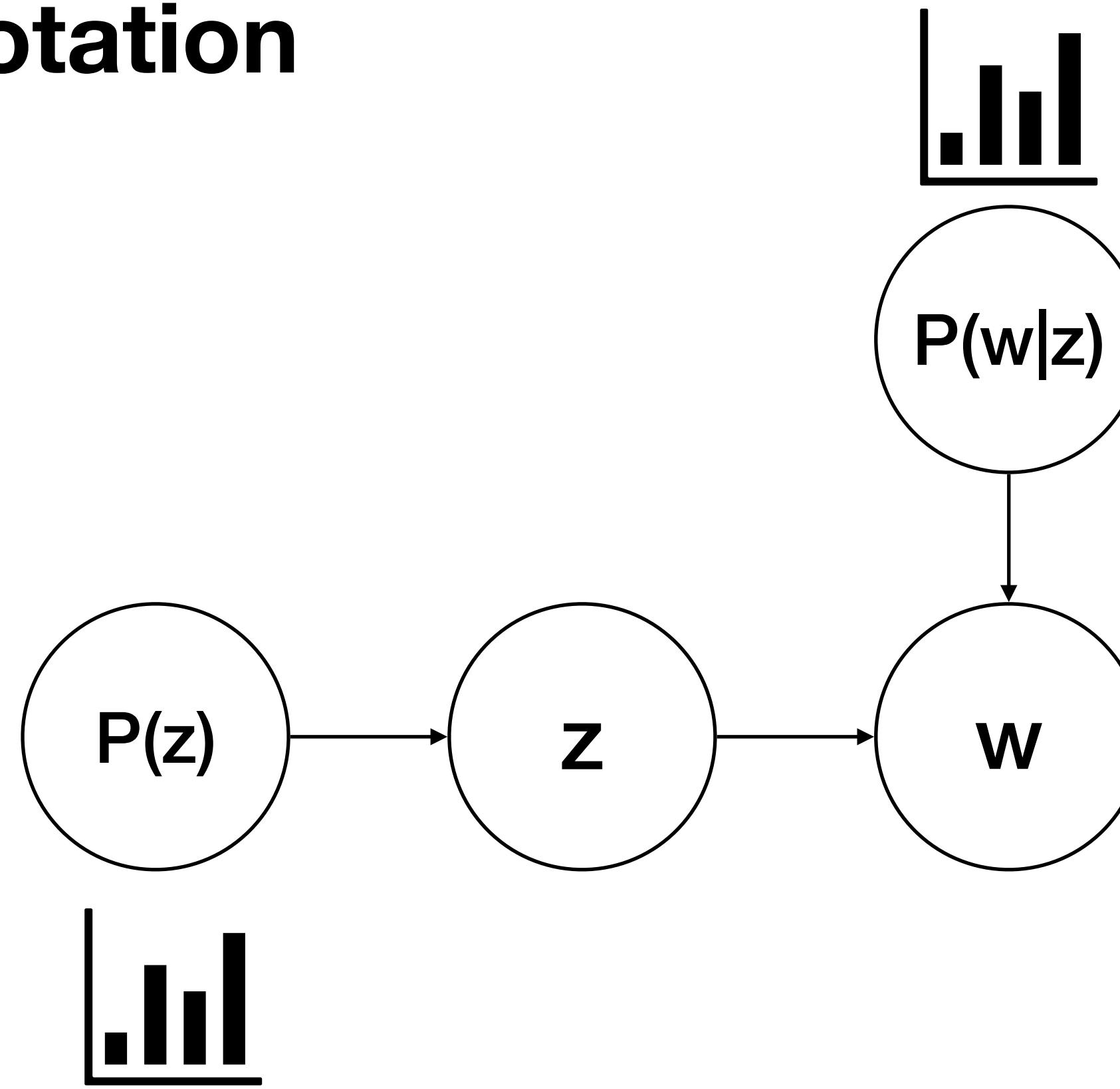
z depends on a multinomial distribution $P(z)$



$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

Graphical Model Notation

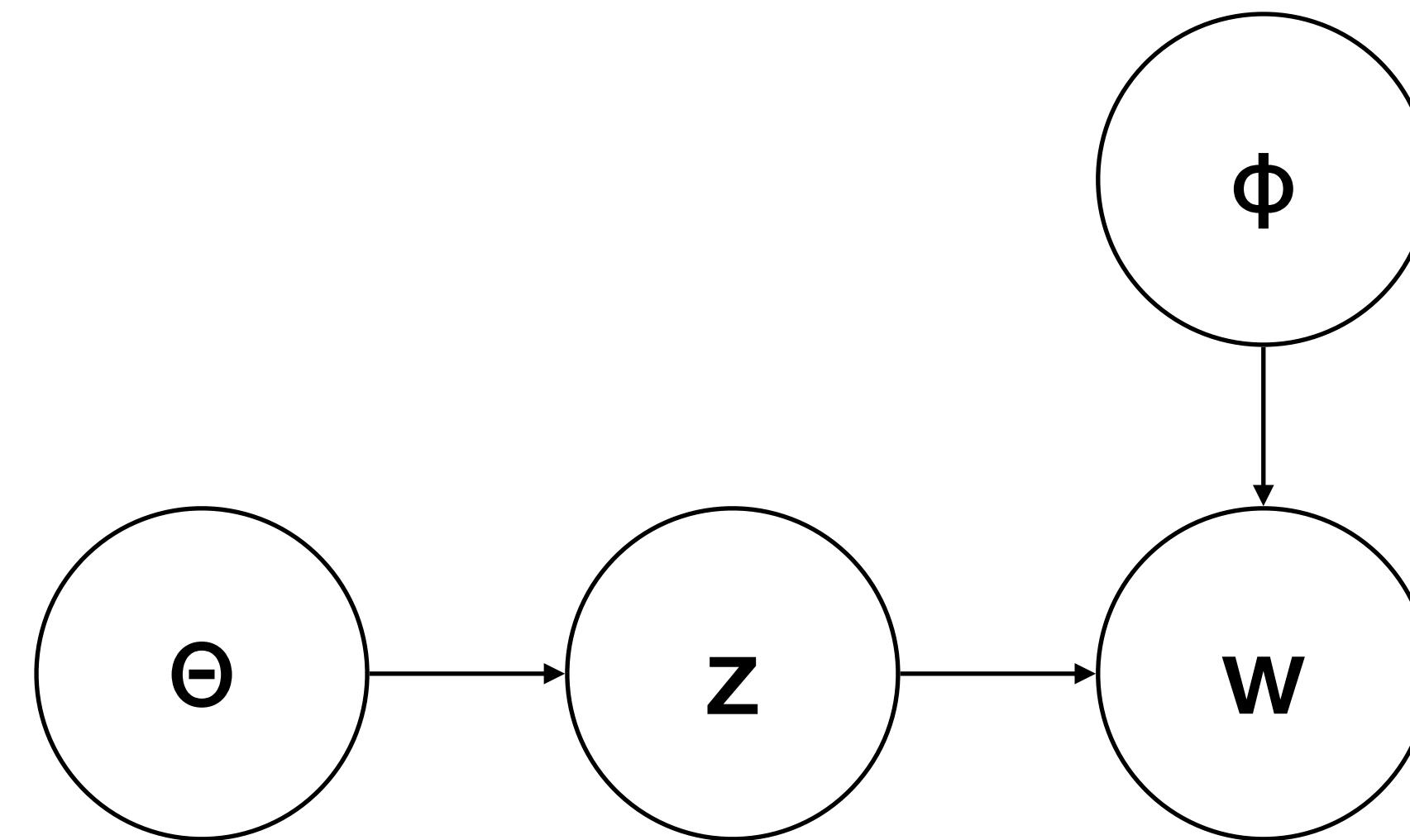


w depends on a multinomial distribution $P(w|z)$

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

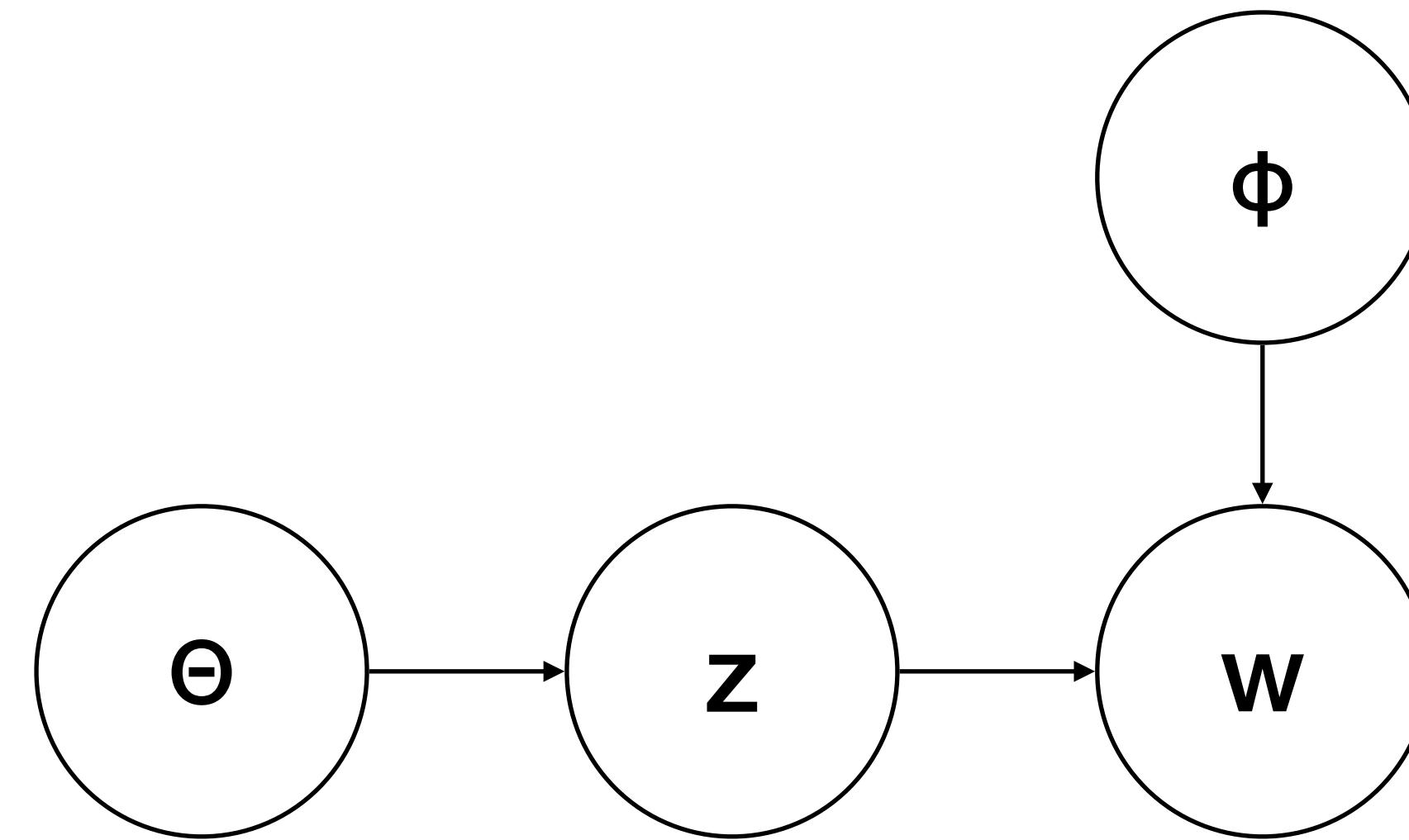
Graphical Model Notation



$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) P(z_i = j)$$

Latent Dirichlet Allocation (LDA)

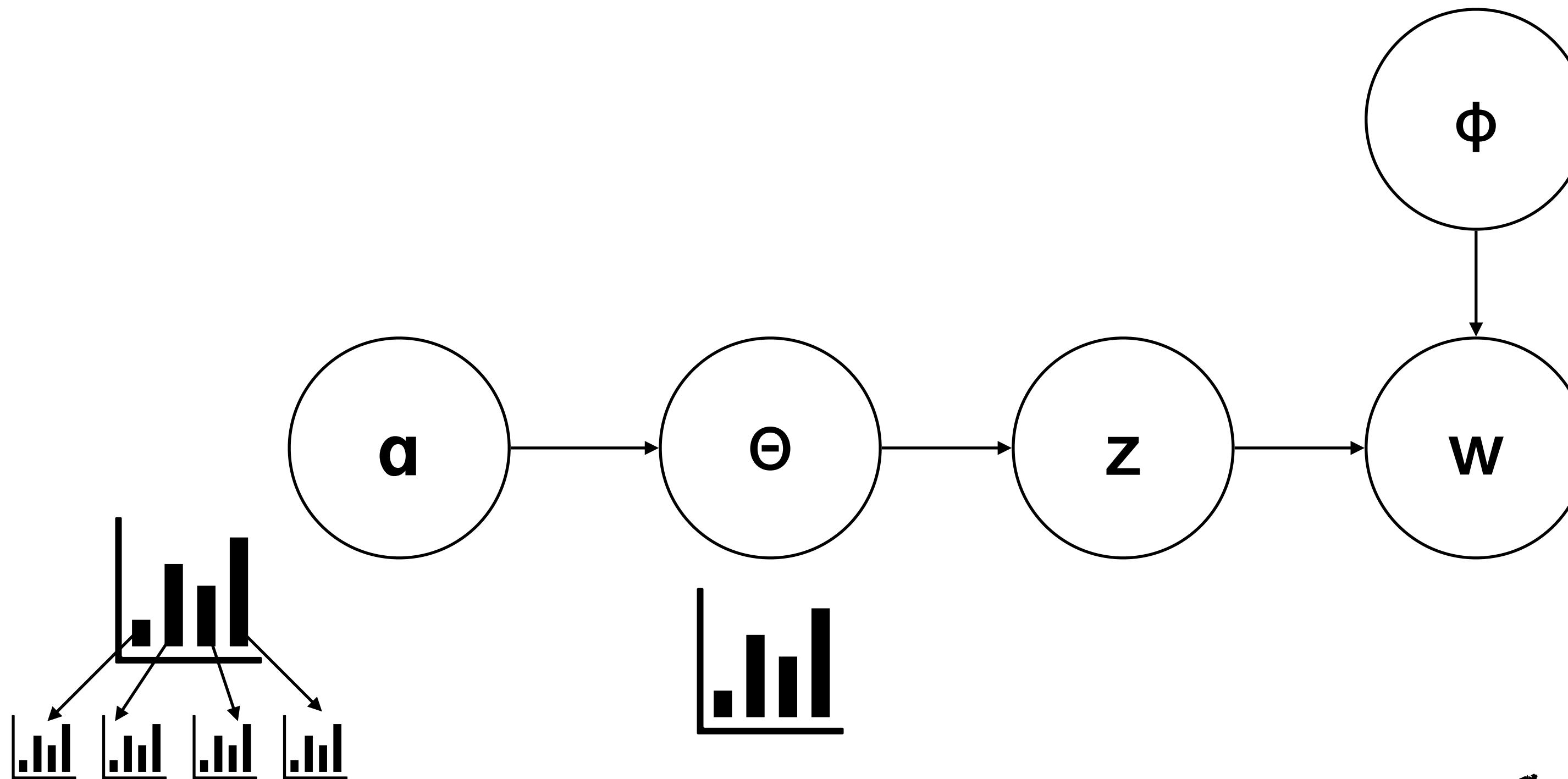
Graphical Model Notation



*Θ needs to come from
somewhere!*

Latent Dirichlet Allocation (LDA)

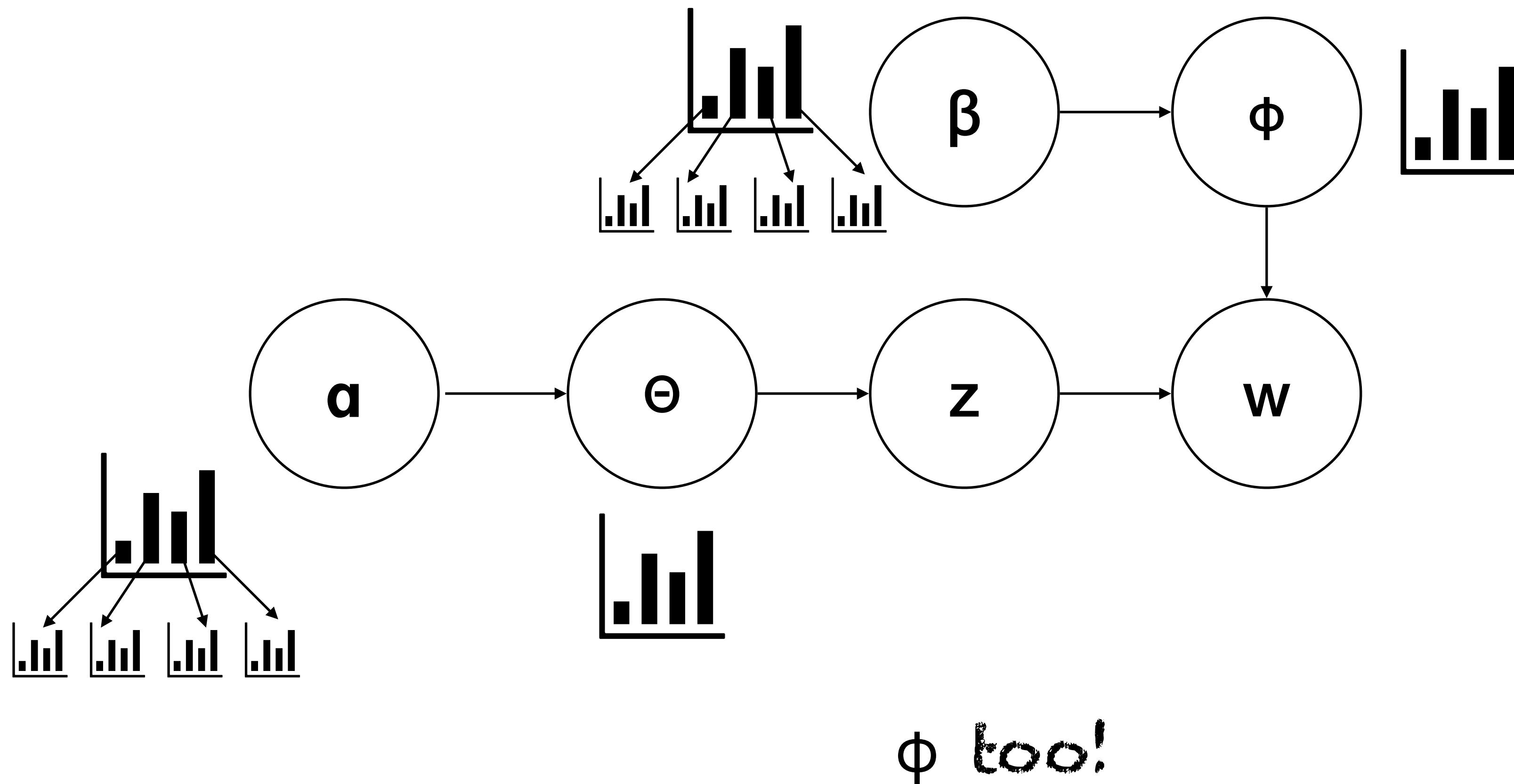
Graphical Model Notation



we'll assume it comes from a
Dirichlet distribution a

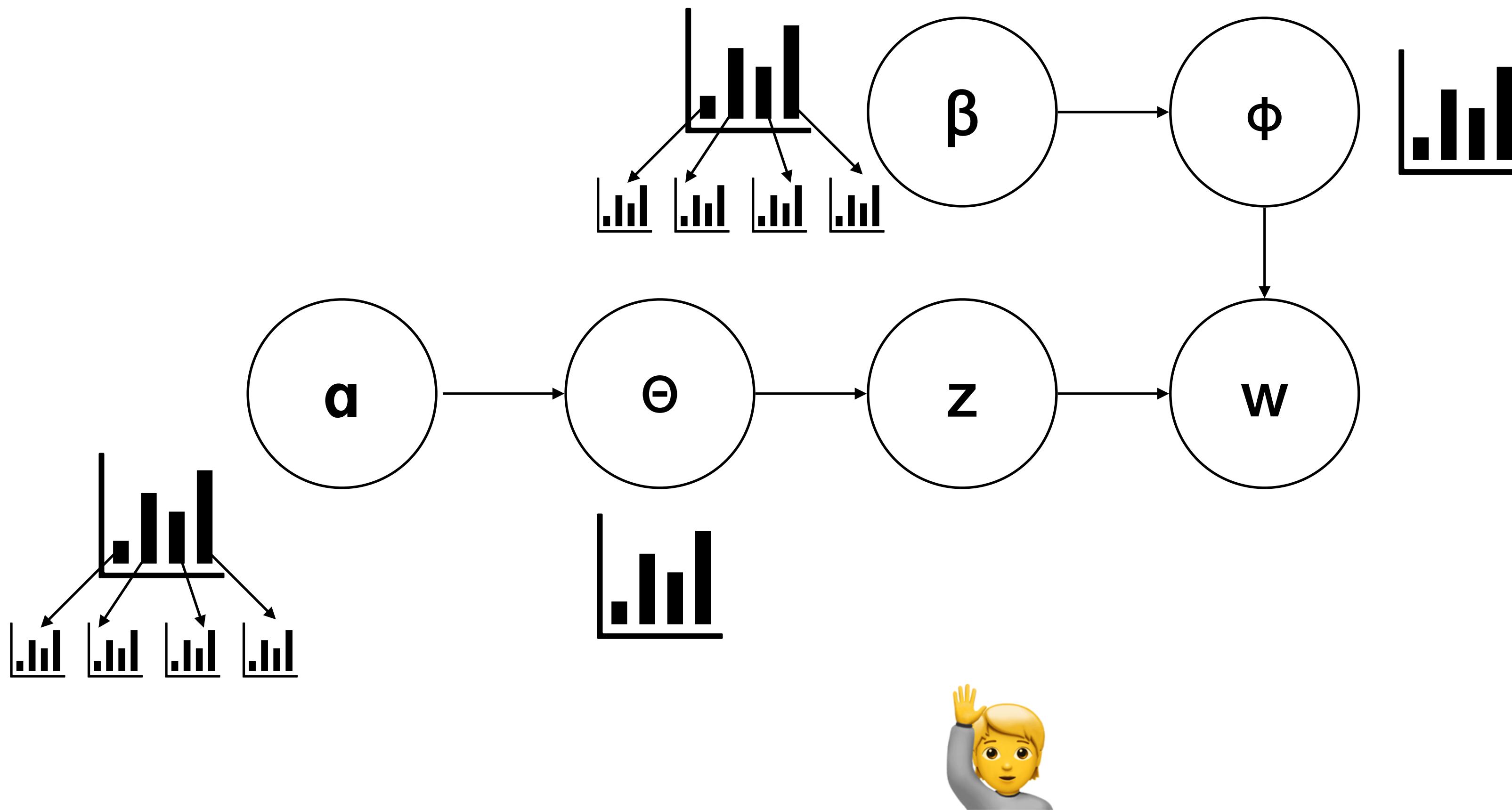
Latent Dirichlet Allocation (LDA)

Graphical Model Notation



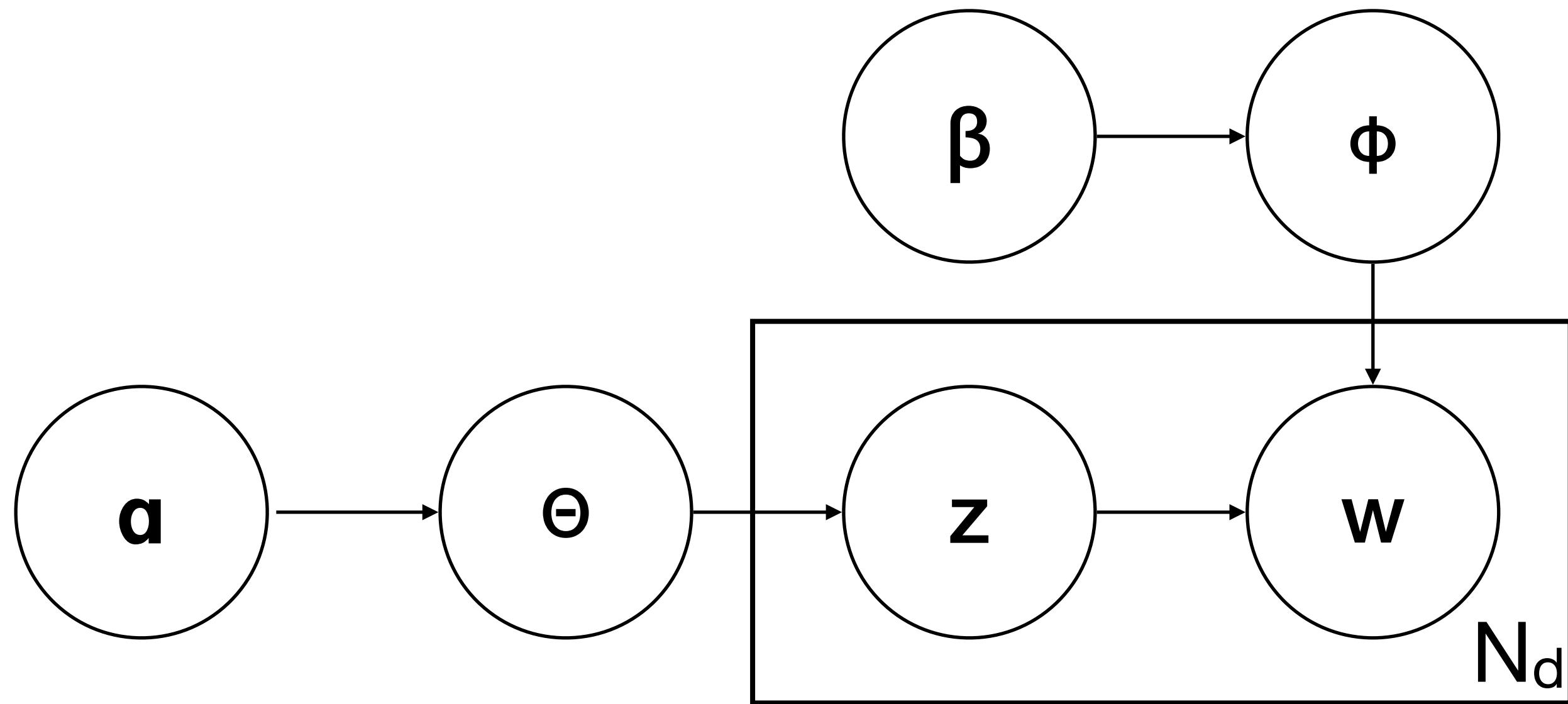
Latent Dirichlet Allocation (LDA)

Graphical Model Notation



Latent Dirichlet Allocation (LDA)

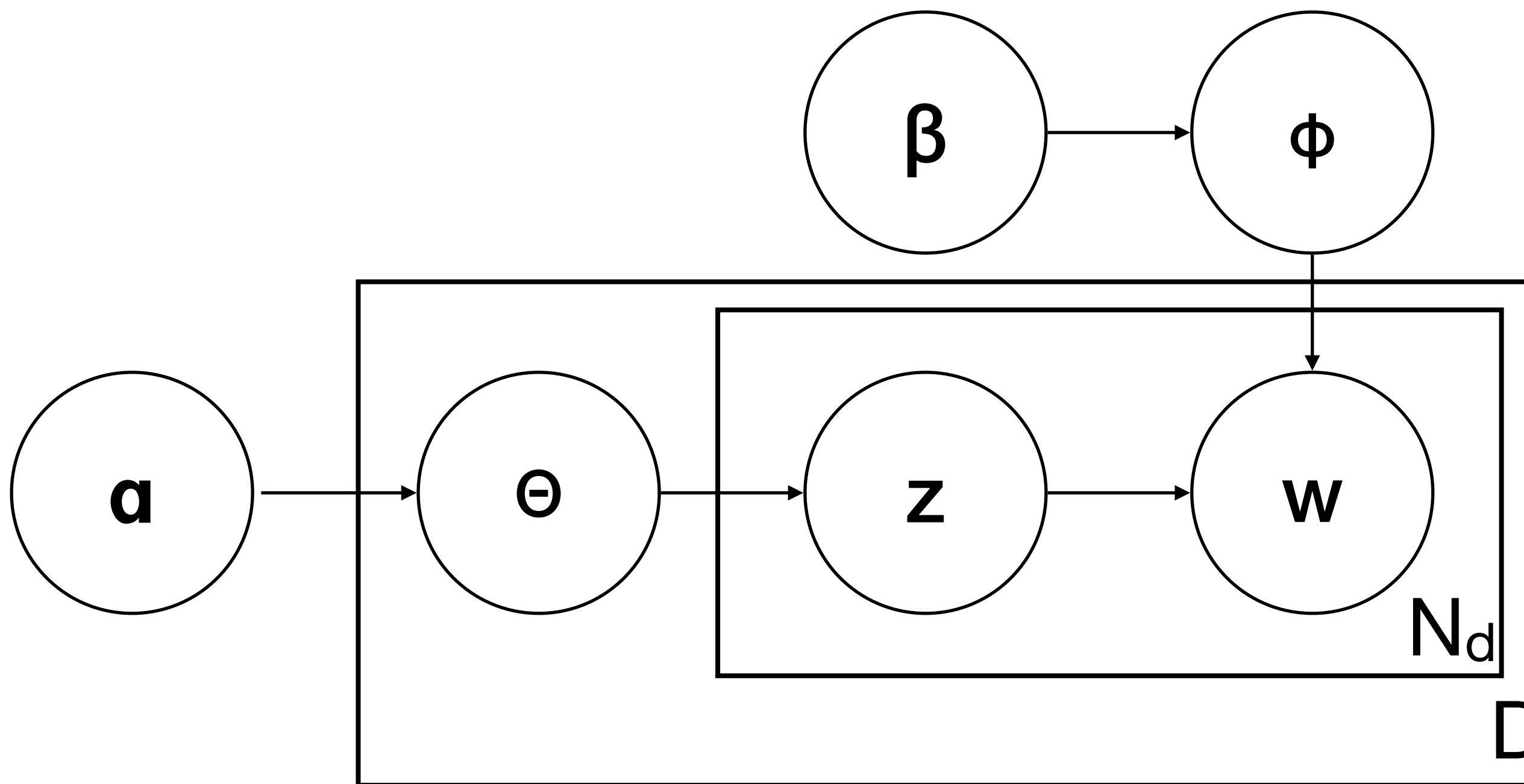
Graphical Model Notation



this step happens
 N_d times, once for
each of N_d words
in a document d

Latent Dirichlet Allocation (LDA)

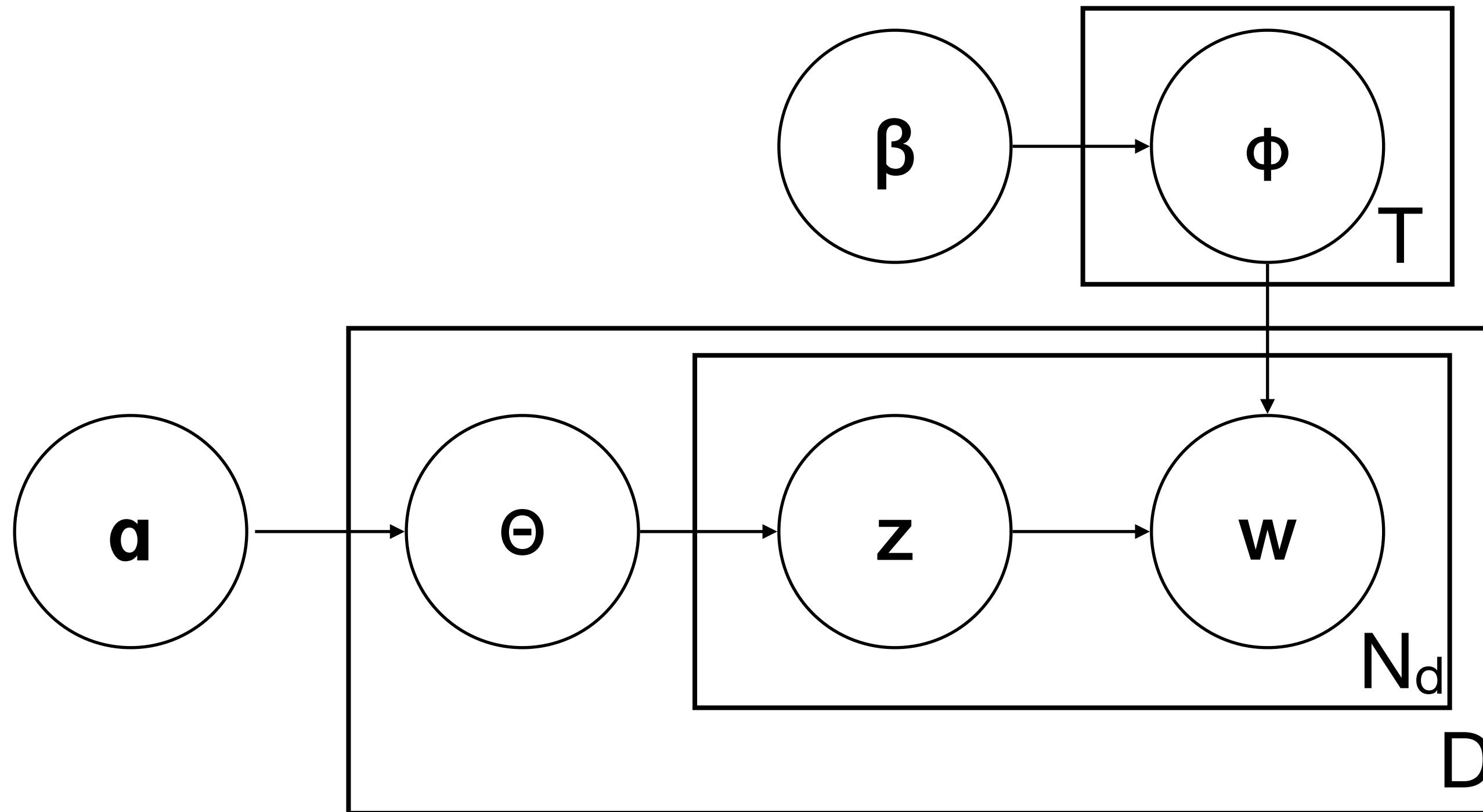
Graphical Model Notation



this step happens D
times, once for each
of D documents

Latent Dirichlet Allocation (LDA)

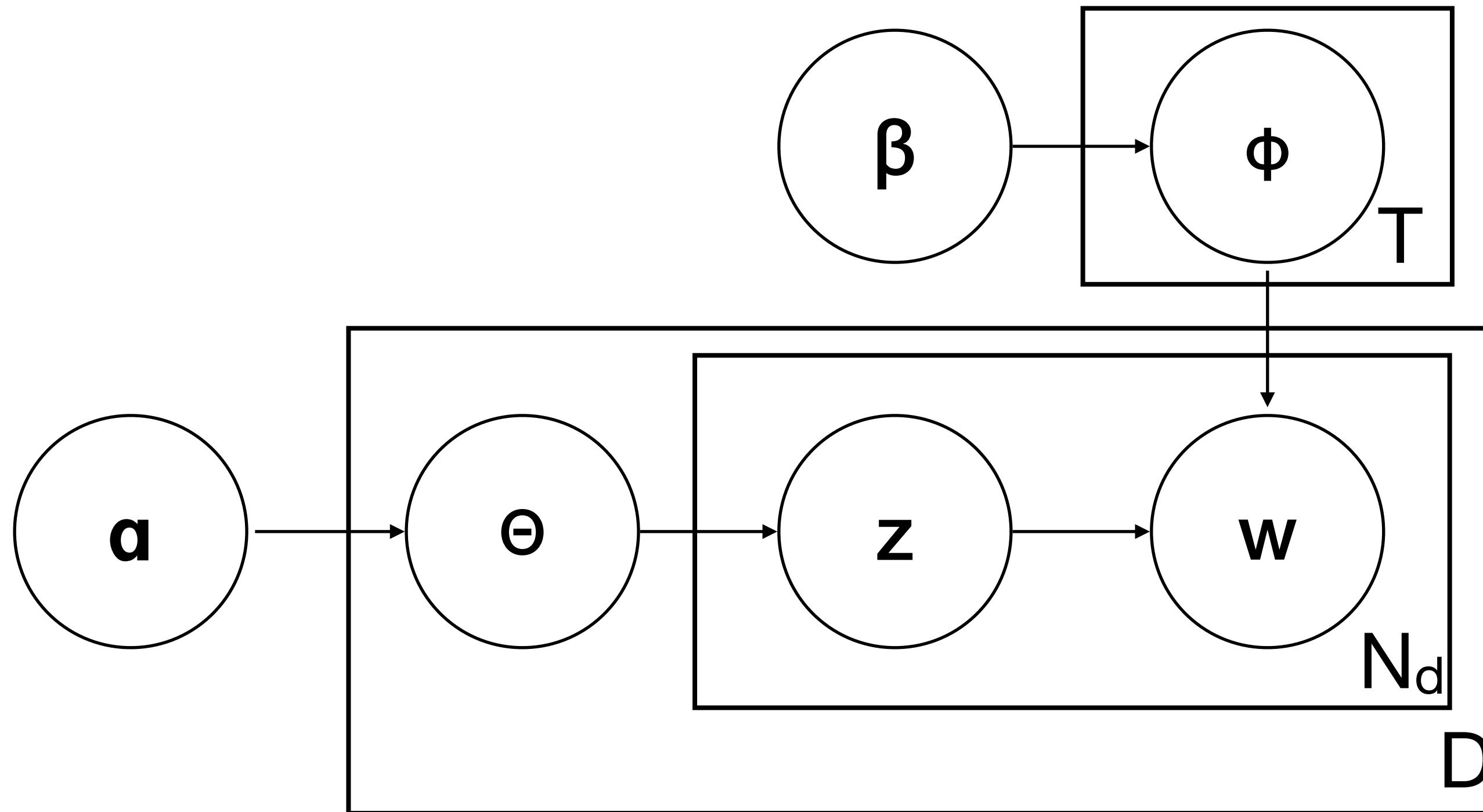
Graphical Model Notation



this step
happens T
times, once
for each of T
topics

Latent Dirichlet Allocation (LDA)

Graphical Model Notation



this step
happens T
times, once
for each of T
topics

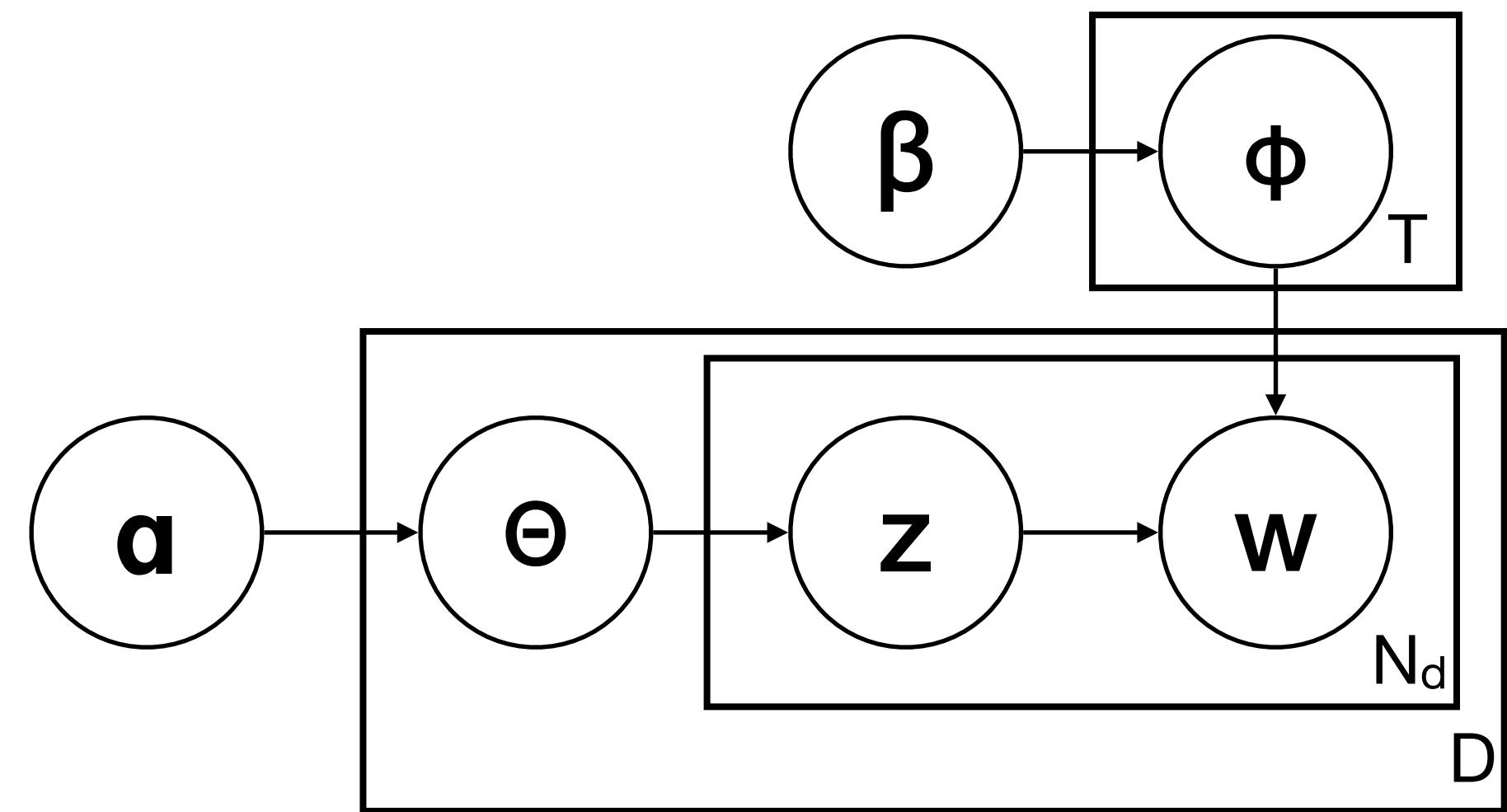


Latent Dirichlet Allocation (LDA)

Graphical Model Notation <-> Generative Story

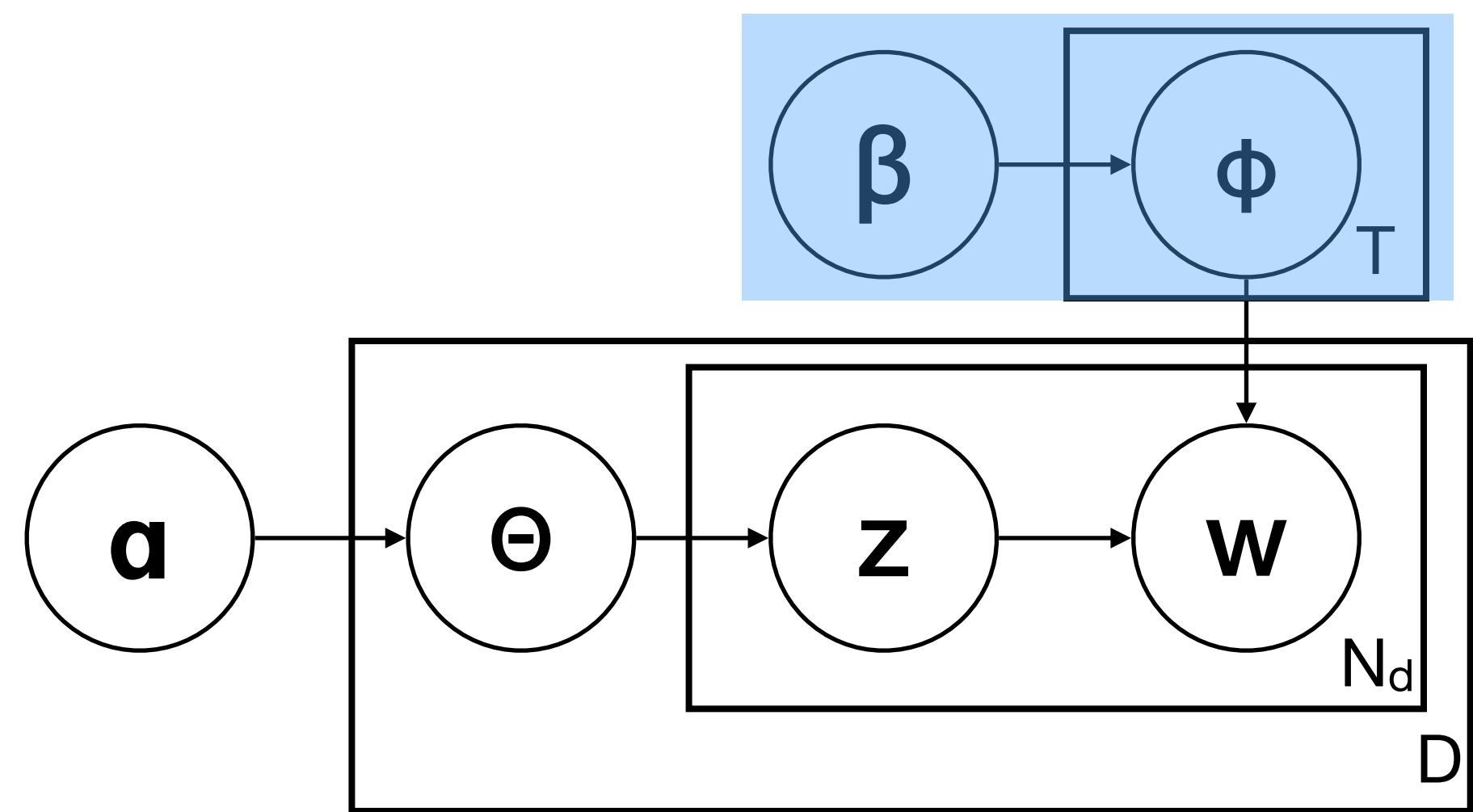
Latent Dirichlet Allocation (LDA)

Graphical Model Notation <-> Generative Story

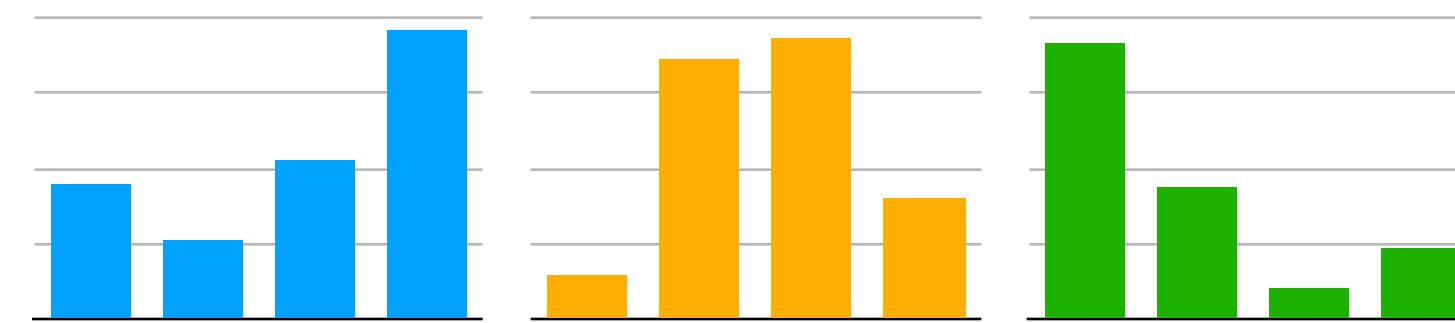


Latent Dirichlet Allocation (LDA)

Graphical Model Notation <-> Generative Story



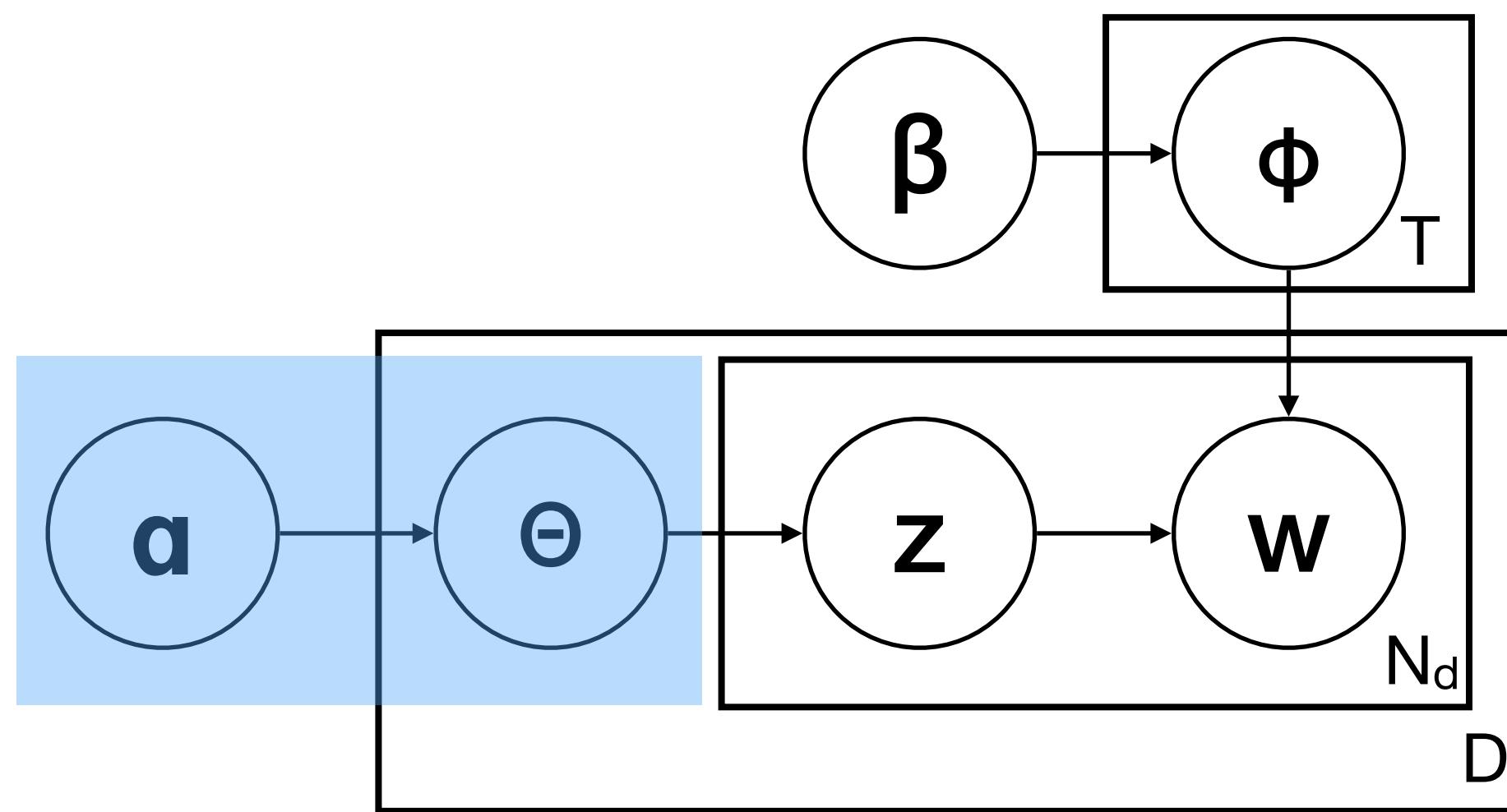
T1 T2 T3



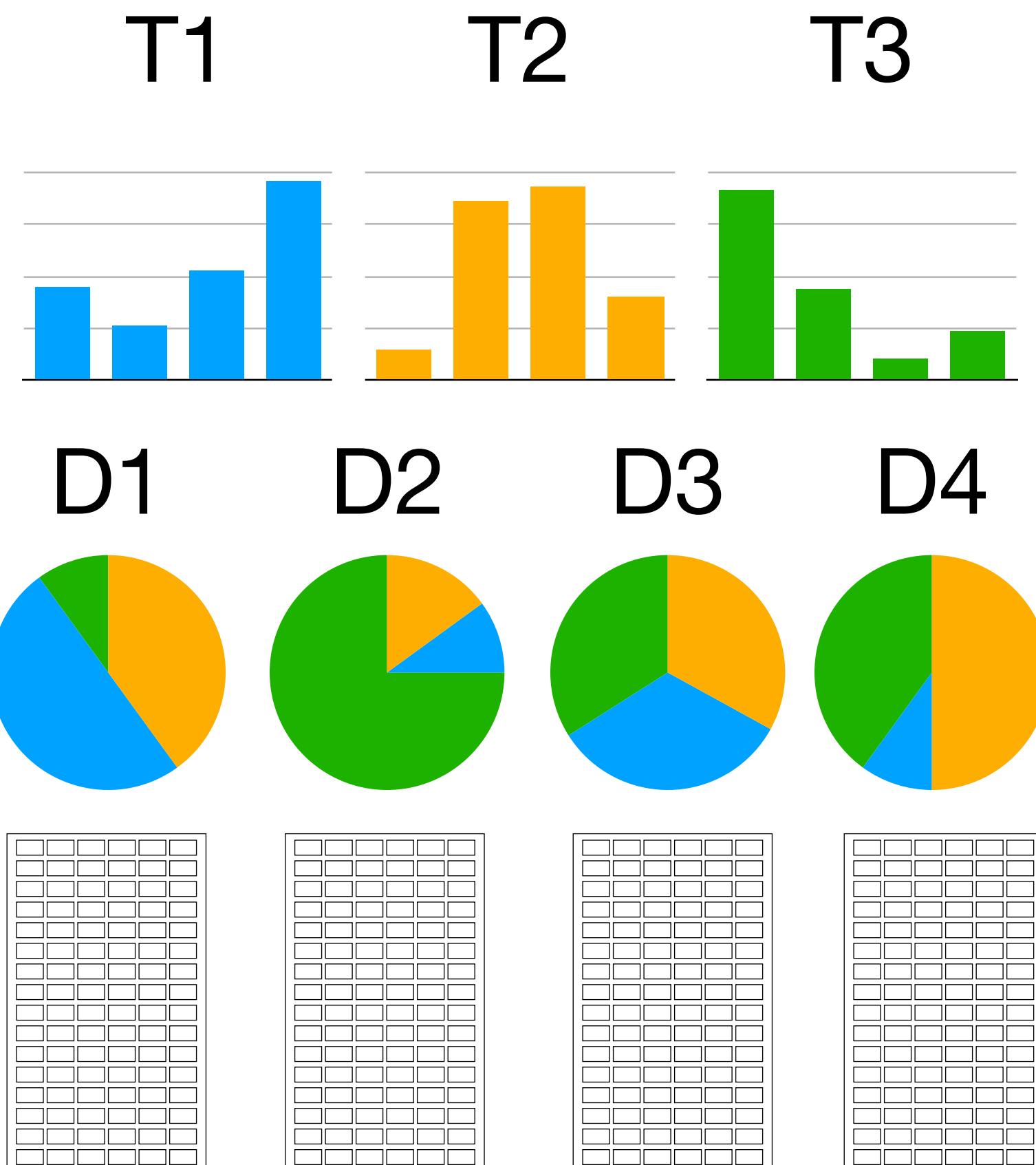
For each topic, sample a
multinomial distribution
over words

Latent Dirichlet Allocation (LDA)

Graphical Model Notation <-> Generative Story

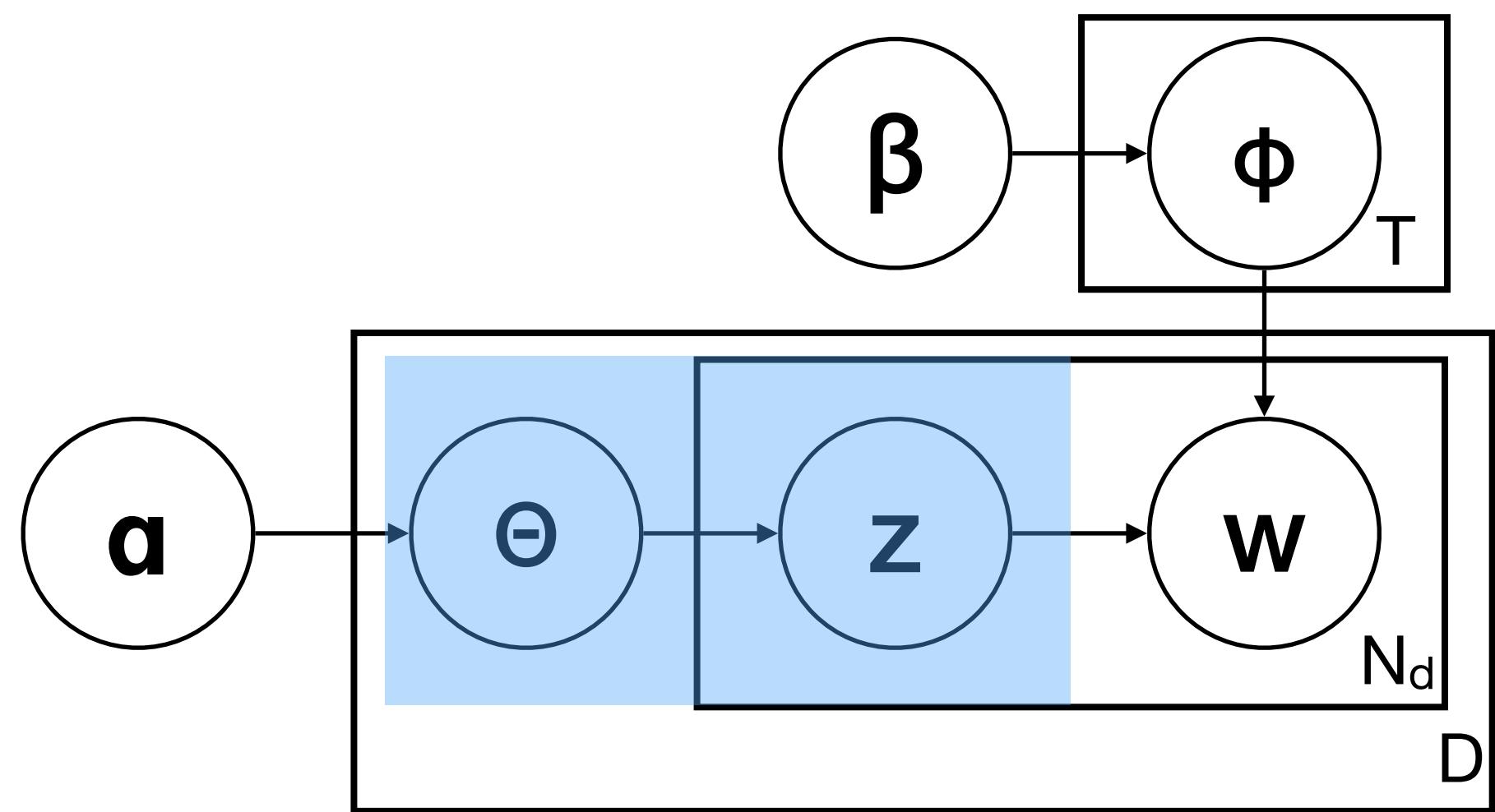


For each document,
sample a multinomial
distribution over topics

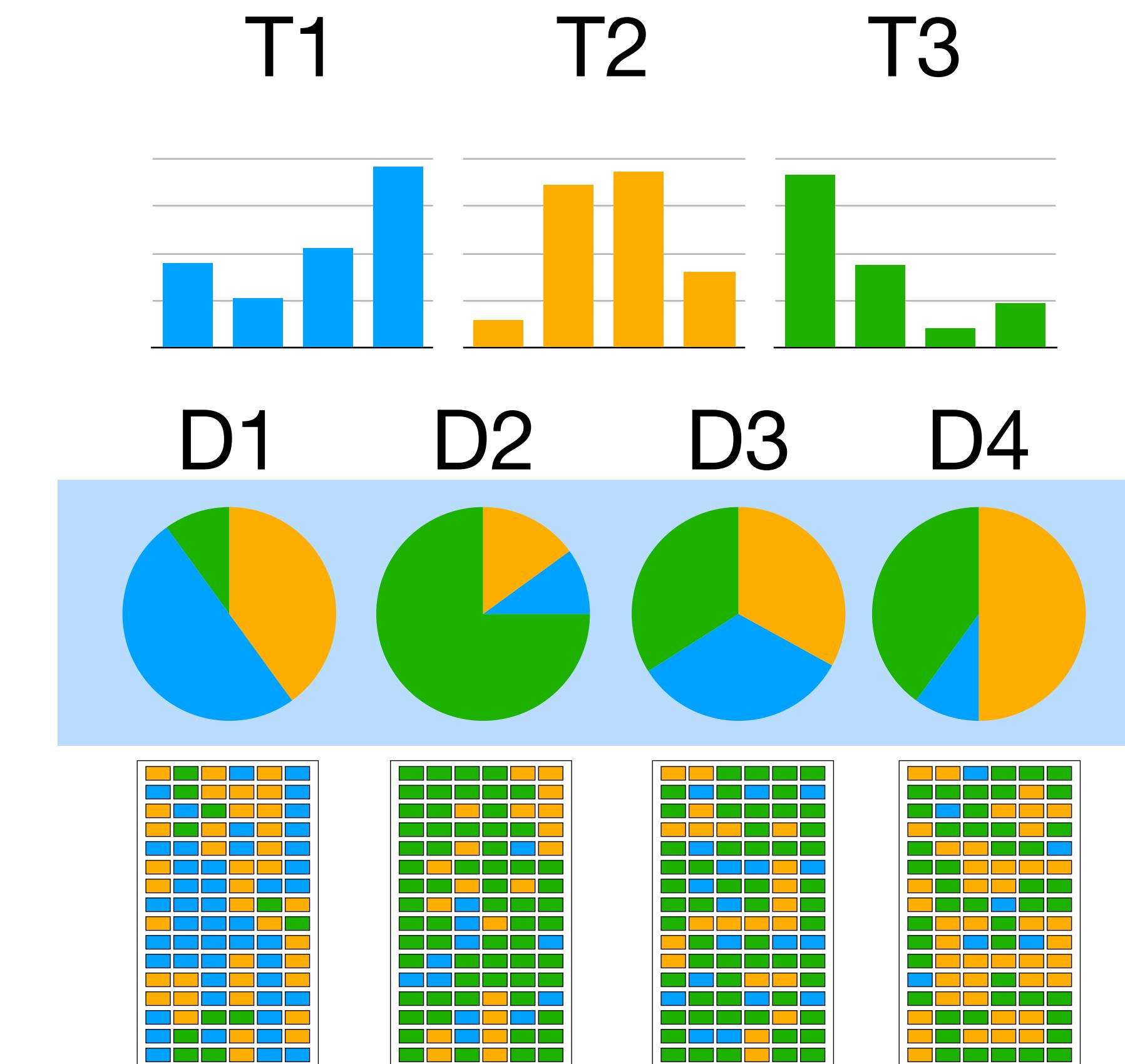


Latent Dirichlet Allocation (LDA)

Graphical Model Notation <-> Generative Story

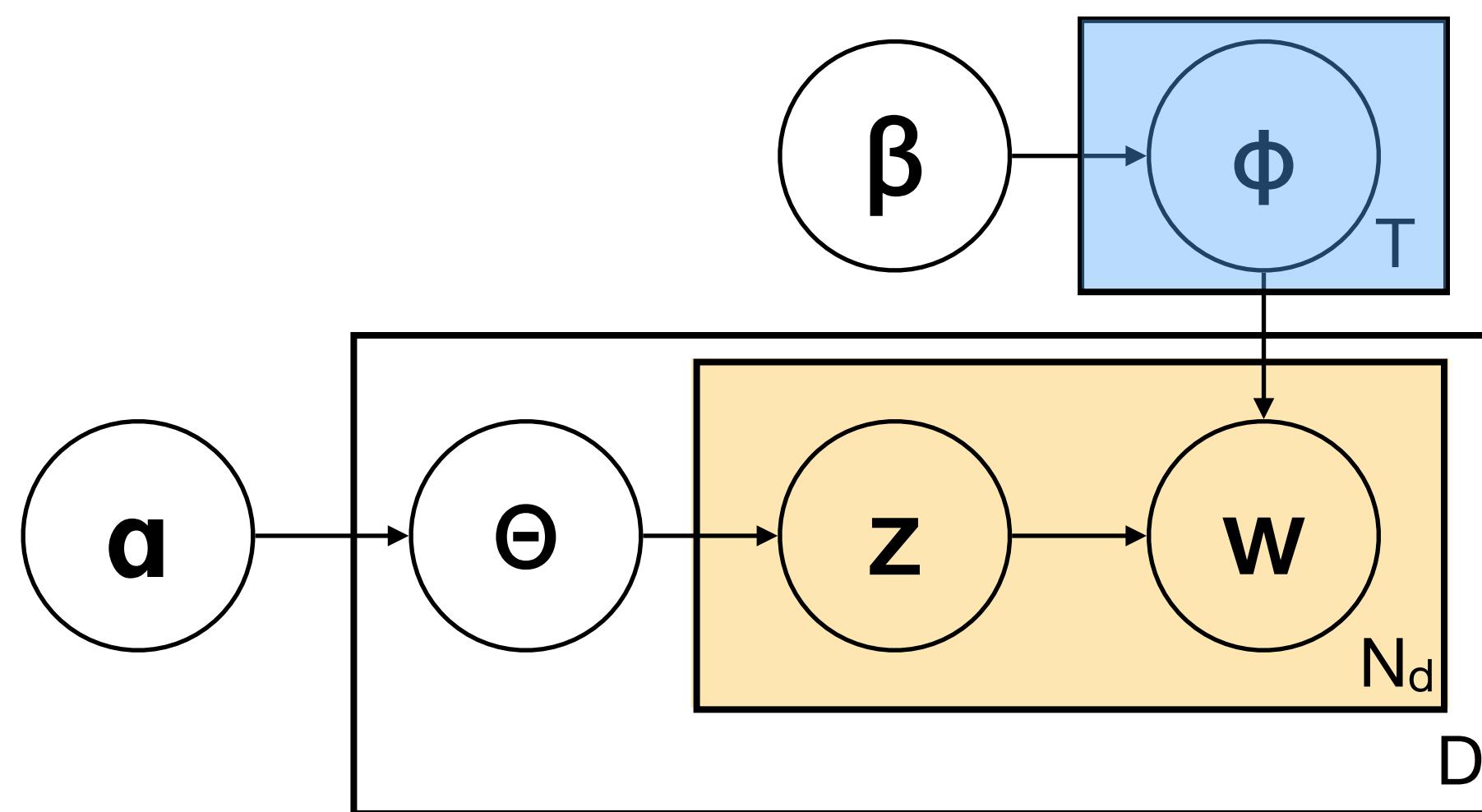


For each document, for
each word, sample a
topic

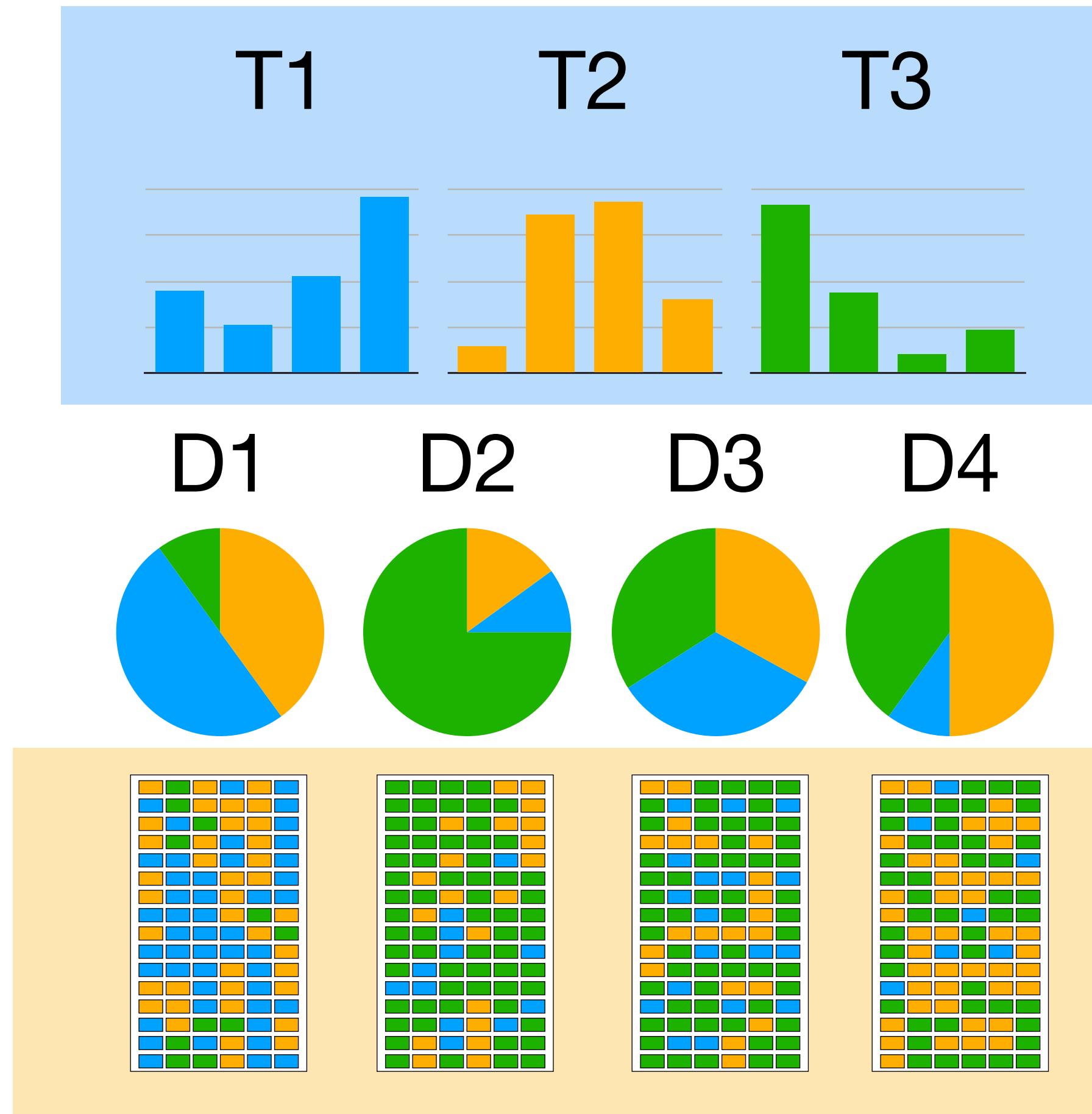


Latent Dirichlet Allocation (LDA)

Graphical Model Notation <-> Generative Story



For document, for each word, given the sampled topic, sample a word



Latent Dirichlet Allocation (LDA)

Posterior Estimation

- We set up a probability model, but how do we set the parameters of the model?
 - I.e., how do we decide the actual probabilities of words for a given topic, probabilities of topics, etc?
- Recall:
 - We are trying to $P(z|w) = P(z,w)/P(w)$
 - $P(w)$ is intractable to estimate in general
 - (Same issue we dealt with in Naive Bayes! In NB, we made aggressive independence assumptions, but we don't want to do that here)

Latent Dirichlet Allocation (LDA)

Posterior Estimation

- Resort to approximate methods for **posterior estimation**
 - Details beyond the scope of this course (see, maybe, APMA 1740)
- Broadly, two types of approaches:
 - Sampling Methods (MCMC, Gibbs Sampling)
 - Variational Inference

Latent Dirichlet Allocation (LDA)

Posterior Estimation: Basic Ideas

- Gibbs Sampling
 - Start by randomly assigning words to topics
 - Iteratively make small updates until your underlying model begins to mirror the observed posterior
 - <https://www.youtube.com/watch?v=u7l5hhmdc0M>
- Variational Inference
 - Posit some simpler probability distribution parameterized by some parameters Θ
 - Use optimization algorithms to optimize Θ until the result is close to the true, observed posterior

