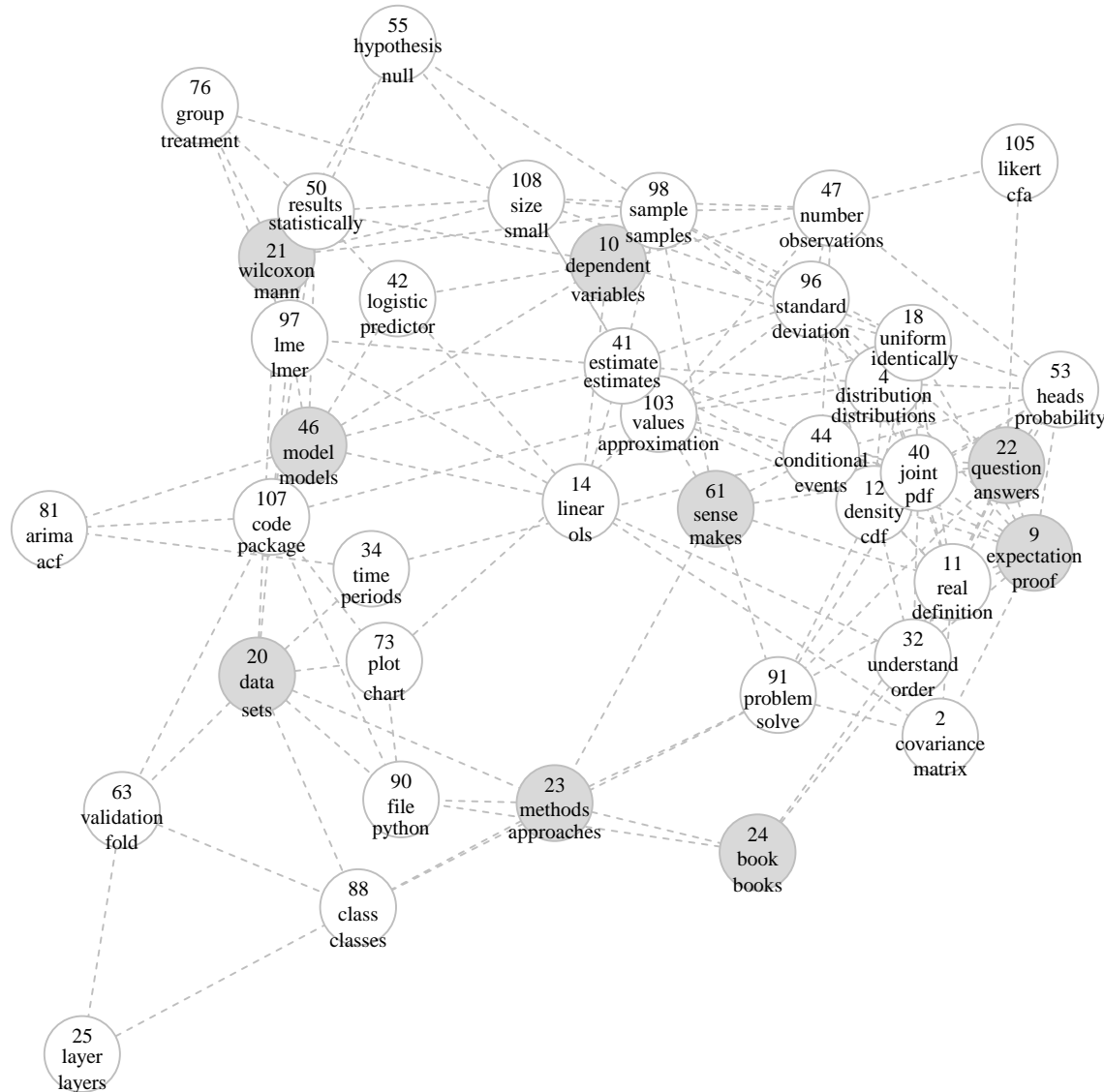


Crowdsourced Statistics Advice: Topic Modeling stats.stackexchange.com




Paul Hively
May 5, 2017

Agenda

- A few words on stats.stackexchange.com
- Topic modeling primer
- Latent Dirichlet allocation (LDA)
- Correlated topic models (CTM)
- Structural topic models (STM)
- Data analysis

stats.stackexchange.com


 Cross Validated

QUESTIONS TAGS USERS BADGES UNANSWERED ASK QUESTION


Cross Validated is a question and answer site for people interested in statistics, machine learning, data analysis, data mining, and data visualization. Join them; it only takes a minute:

Sign up


Here's how it works:



Anybody can ask a question



Anybody can answer



The best answers are voted up and rise to the top

Explore Our Questions

active 13 featured hot week month

r regression machine-learning time-series probability self-study hypothesis-testing distributions logistic classification

more tags

0 votes

2 answers

201 views

Using Covariates in Multi-State Modeling (MSM) in R

r markov-process transition-matrix

modified 2 mins ago Community ♦ 1

0 votes

0 answers


4 views


Can anyone answer this?


regression logistic


asked 4 mins ago Dingle 1


Hot Network Questions


 How did this bullet pass through Wolverine's adamantium-covered skull?

 Can a character wear armour that he's not proficient in?

 "sprinG summEr autumN winTer" - what does this mean?

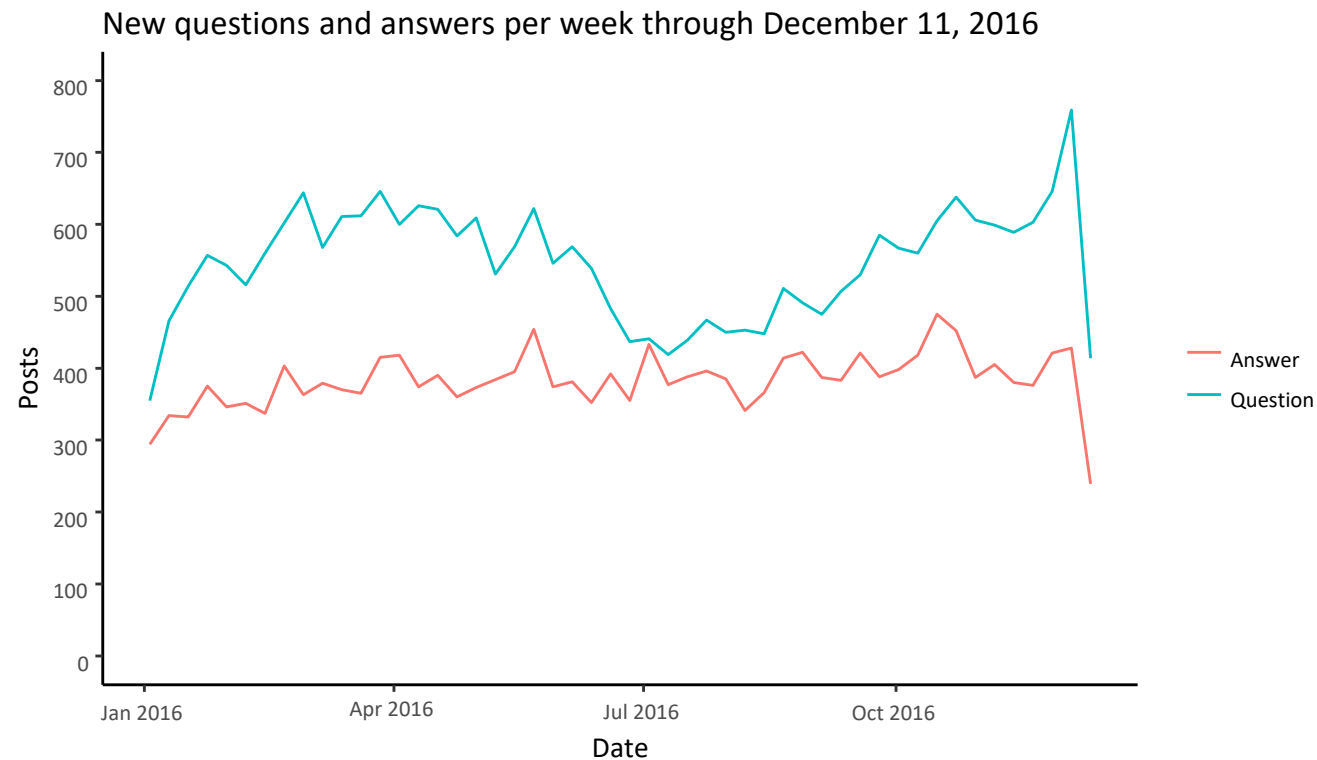
 A Message for the Solver!

 Was it possible to get more than 8 sprites per scanline on an Amiga with the copper list?

 Trolls: any evidence that they were mutilated Ents?

stats.stackexchange.com

- Statistics and machine learning question-and-answer website
- All content created by users
- Hundreds of new posts each week



stats.stackexchange.com

2 Answers

active oldest votes

▲
8

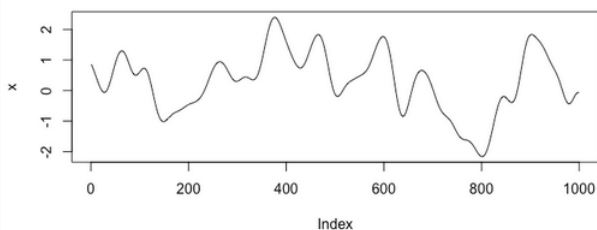
Choosing a kernel is equivalent to choosing a class of functions from which you will choose your model. If choosing a kernel feels like a big thing that encodes a lot of assumptions, that's because it is! People new to the field often don't think much about the choice of kernel and just go with the Gaussian kernel even if it's not appropriate.

▼
✓

How do we decide whether or not a kernel seems appropriate? We need to think about what the functions in the corresponding function space look like. The Gaussian kernel corresponds to very smooth functions, and when that kernel is chosen the assumption is being made that smooth functions will provide a decent model. That's not always the case, and there are tons of other kernels that encode different assumptions about what you want your function class to look like. There are kernels for modeling periodic functions, non-stationary kernels, and a whole host of other things. For example, the smoothness assumption encoded by the Gaussian kernel is not appropriate for text classification, as shown by Charles Martin in his blog [here](#).

Let's look at examples of functions from spaces corresponding to two different kernels. The first will be the Gaussian kernel $k_1(x, x') = \exp(-\gamma|x - x'|^2)$ and the other will be the Brownian motion kernel $k_2(x, x') = \min\{x, x'\}$. A single random draw from each space looks like the following:

Gaussian Kernel



The Two Cultures: statistics vs. machine learning?

▲
313

Last year, I read a blog post from [Brendan O'Connor](#) entitled "Statistics vs. Machine Learning, fight!" that discussed some of the differences between the two fields. [Andrew Gelman](#) responded favorably to this:

Simon Blomberg:

★
313

From R's fortunes package: To paraphrase provocatively, 'machine learning is statistics minus any checking of models and assumptions'. -- Brian D. Ripley (about the difference between machine learning and statistics) useR! 2004, Vienna (May 2004) :- Season's Greetings!

Andrew Gelman:

In that case, maybe we should get rid of checking of models and assumptions more often. Then maybe we'd be able to solve some of the problems that the machine learning people can solve but we can't!

There was also the "Statistical Modeling: The Two Cultures" paper by Leo Breiman in 2001 which argued that statisticians rely too heavily on data modeling, and that machine learning techniques are making progress by instead relying on the *predictive accuracy* of models.

Has the statistics field changed over the last decade in response to these critiques? Do the two cultures still exist or has statistics grown to embrace machine learning techniques such as neural networks and support vector machines?

machine-learning pac-learning

share improve this question

edited Apr 8 at 17:58

community wiki
7 revs, 6 users 63%
Shane

Conditional entropy and Spearman's correlation based lag in time series

▲

I have two time series A, B. Both are seasonal and B primarily is A driven(other temporal causes may exist). [graphs](#) <http://59.tinypic.com/2njfpd.jpg>

0

B-Red, A- Green

▼

I want to calculate lag of red series with respect to green as clearly, it exists.

★

Now, I am taking Spearman's correlation at different lags and choosing the max to decide the lag which gives satisfying answer.

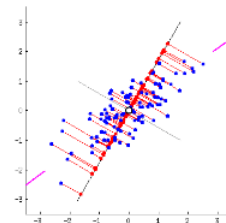
To confirm the confidence in lag, I am trying entropy based methods.

I tried conditional entropy and here are the results.

lag	C-Entropy	corr
0	1.0828745	-0.735406343
1	0.8978593	-0.830377446
2	1.1218689	-0.689623230
3	1.2412857	-0.336204576
4	1.2985196	0.854672496
5	1.2727747	0.485228731
6	1.1827205	0.771465042
7	0.9616463	0.839862100
8	1.1296509	0.677166842
9	1.2885970	0.396834333
10	1.3428290	0.005832166

Each dot in this "wine cloud" shows one particular wine. You see that the two properties (x and y on this figure) are correlated. A new property can be constructed by drawing a line through the center of this wine cloud and projecting all points onto this line. This new property will be given by a linear combination $w_1x + w_2y$, where each line corresponds to some particular values of w_1 and w_2 .

Now look here very carefully -- here is how these projections look like for different lines (red dots are projections of the blue dots):



stats.stackexchange.com

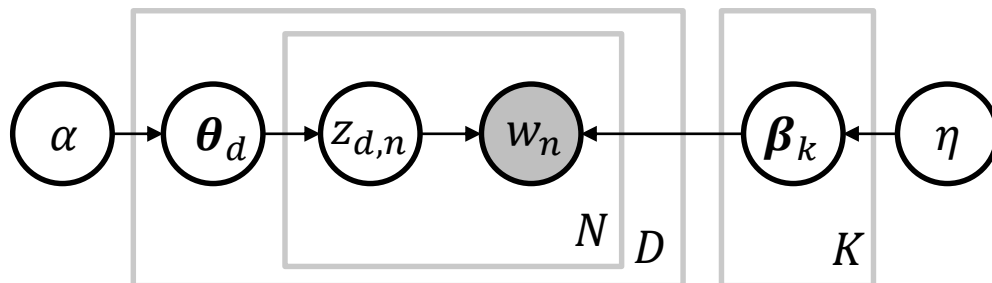
- As of December 2016: over 182,000 posts containing over 20 million words
- How can this rich collection of data be explored?

Explore Our Questions

active 13 featured hot week month

r regression machine-learning time-series probability self-study hypothesis-testing distributions logistic classification

more tags



Topic modeling primer

- Unsupervised method (no more manual tagging!)
- Hierarchical Bayesian generative probabilistic models
- Models are fit to a *corpus* $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_D\}$, a collection of D documents



Cross Validated

For people interested in statistics, machine learning, data analysis, data mining, and data visualization

- Documents \mathbf{w}_d are arbitrarily long vectors of words (can be repeated)

I have been wondering why people use Gaussian processes (GP) to model an unknown (sometimes deterministic) function. For instance consider an unknown function $y = f(x)$. We have three **independent** observations from this function :

- Documents are about some combination of K different topics (K known)
- Words w_n are discrete units of text taken from a vocabulary \mathcal{V} , a vector of length V that indexes all unique words in the corpus

Topic modeling primer

- A topic β_k is a distribution of words over the vocabulary

I have a deep neural network model and I need to train it on my dataset which consists of about 100,000 examples, my validation data contains about 1000 examples. Because it takes time to train each example (around 0.5s for each example) and in order to avoid overfitting, I would like to apply early stopping to prevent unnecessary computation. But I am not sure how to properly train my neural network with early stopping, several things I do not quite understand now:

- A topic proportion mixture θ_d is a distribution of topics over a document

Gaussian process and Correlation

machine-learning

correlation

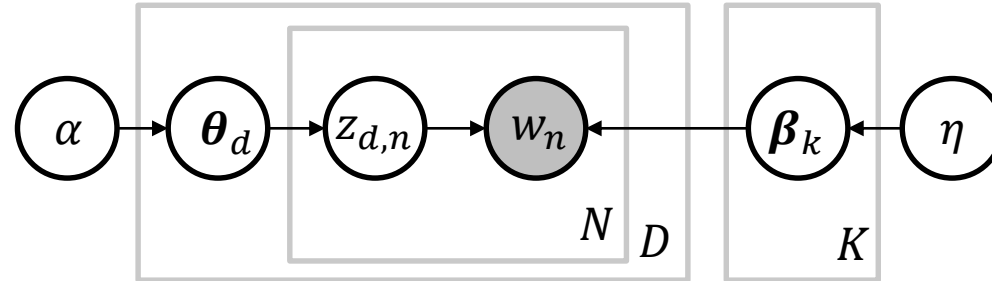
covariance

gaussian-process

- Each word w_n is assumed to be generated according to a *latent variable* $z_n \in \{1, \dots, K\}$, which is a random topic draw from θ_d determining which β_k generates the observed word

Latent Dirichlet allocation

- Blei, Ng, & Jordan (2003)



- Basic idea: the θ and β are the parameters of multinomial distributions generated from Dirichlet distributions $\text{Dir}(\alpha)$, $\text{Dir}(\eta)$
- Each word has a topic assignment $z_{d,n}$ drawn from $\text{Multi}(\theta_d)$; the observed word w_n is drawn from $\text{Multi}(\beta_{z_{d,n}})$
- On the board: model generation, node dimensions
- To fit, use Bayes' theorem to write the posterior and optimize:

$$\underset{\theta, \beta}{\operatorname{argmax}} p(\theta, \mathbf{z}_{1:D, 1:N}, \beta | \mathbf{w}_{1:D, 1:N}, \alpha, \eta)$$

Latent Dirichlet allocation

Some assumptions

- *Bag-of-words assumption*: word order doesn't matter (complete exchangeability)
 - i.e. $P(\mathbf{w}_d = \{w_1, \dots, w_{N_d}\}) = P(\mathbf{w}_d = \{w_{(1)}, \dots, w_{(N_d)}\})$
 - This isn't realistic – but it doesn't matter! (Why not? Advantages?)
- Topic structure: topics are nearly uncorrelated (can be relaxed)
- Document exchangeability: order of documents in the corpus doesn't matter (can be relaxed)

Latent Dirichlet allocation

Likelihood function

$$\mathcal{L} = \prod_{d=1}^D p(\boldsymbol{\theta}_d | \alpha) p(\boldsymbol{\beta} | \eta) \prod_{n=1}^N p(w_n | z_{d,n}, \boldsymbol{\beta}) p(z_{d,n} | \boldsymbol{\theta}_d)$$

$$\text{Dir}(\boldsymbol{\theta} | \alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}$$

$$\text{Multi}(\mathbf{z} | \boldsymbol{\theta}) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_i \theta_i^{z_i}$$

Latent Dirichlet allocation

How can we compute the posterior?

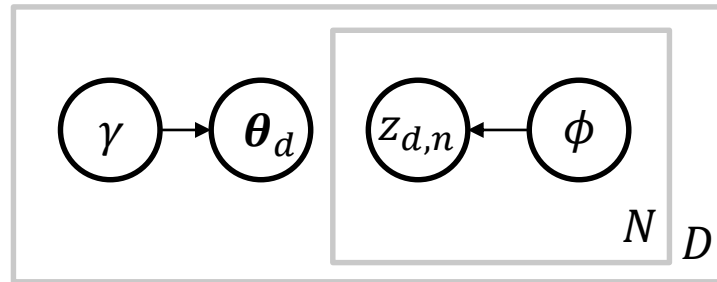
$$p(\boldsymbol{\theta}, \mathbf{z}_{1:D,1:N}, \boldsymbol{\beta} | \mathbf{w}_{1:D,1:N}, \alpha, \eta) = \frac{p(\boldsymbol{\theta}, \mathbf{z}_{1:D}, \boldsymbol{\beta} | \mathbf{w}_{1:D}, \alpha, \eta)}{\int_{\boldsymbol{\beta}_{1:K}} \int_{\boldsymbol{\theta}_{1:D}} \sum_{\mathbf{z}} p(\boldsymbol{\theta}, \mathbf{z}_{1:D}, \boldsymbol{\beta} | \mathbf{w}_{1:D}, \alpha, \eta)}$$

- The denominator is problematic: the \mathbf{z} are hidden
- Approximate via collapsed Gibbs sampling or variational inference

Latent Dirichlet allocation

Variational inference + expectation maximization

- Basic idea: replace the intractable distribution p with a simpler distribution q “close to” the original one (KL divergence)
- On the board: removing the dependence between θ and β



- The simplified distribution q can now be factored:

$$q(\boldsymbol{\theta}, \mathbf{z} | \gamma, \phi) = q_1(\boldsymbol{\theta} | \gamma) \prod_{n=1}^N q_2(z_n | \phi_n)$$

Latent Dirichlet allocation

Variational inference + expectation maximization

- E-step: iterate through each document and set the variational parameters minimizing KL divergence between p and q

$$(\gamma^*, \phi^*) = \operatorname{argmin}_{\gamma, \phi} KL[q(\boldsymbol{\theta}, \mathbf{z} | \gamma, \phi) || p(\boldsymbol{\theta}, \mathbf{z} | \alpha, \eta)]$$

- Intuition: q as a lower bound on the log likelihood
- M-step: set α and η to their maximum likelihood estimates given γ^*, ϕ^*

$$(\alpha^*, \eta^*) = \operatorname{argmax}_{\alpha, \eta} \left[\sum_{d=1}^D \log q(\mathbf{w}_d | \alpha, \eta) \right]$$

- Dirichlet is the conjugate prior to the multinomial

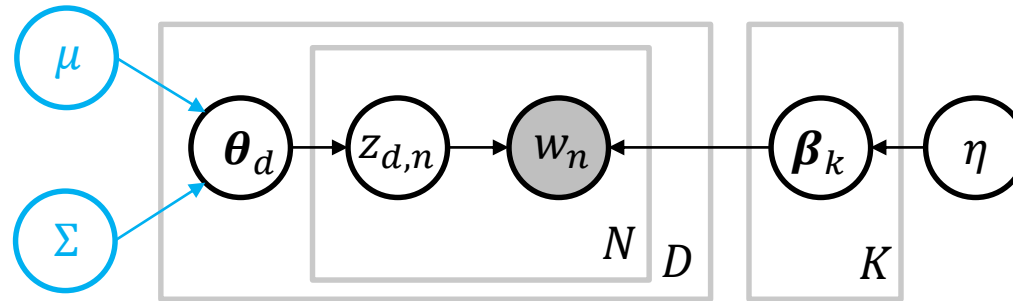
Latent Dirichlet allocation

Model initialization

- Set the hyperparameters via some rule of thumb (e.g. $\propto K^{-1}$)
- Topic initialization: choose your favorite approach
- Naïve approach: random initialization (slow to converge)
- Better approach: initialize topics via collapsed Gibbs sampling
- Tricky approach: deterministic initialization of topics via anchor words (Arora et al., 2013)

Correlated topic models

- Blei & Lafferty, 2007



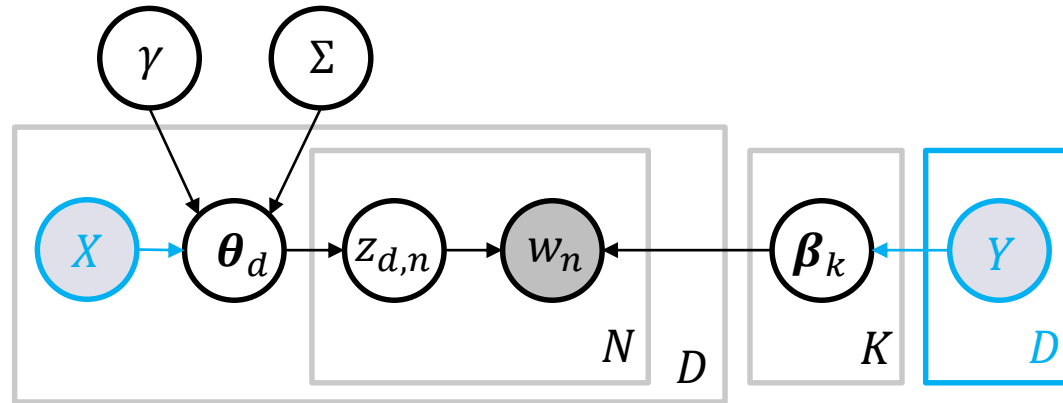
- Replace the Dirichlet prior on θ with a logistic normal distribution:

$$\text{LN}(\pi_i) = \frac{\exp(\pi_i)}{\sum_j \exp(\pi_j)} \quad \forall \pi_i: \boldsymbol{\pi} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), i \in \{1, \dots, K\}$$

- Richer model allowing for topic correlations
- Harder to estimate: no more Dirichlet/multinomial conjugacy
- E-step is still done via (modified) variational inference but the M-step now requires numerical optimization

Structural topic models

- Roberts, Stewart, Tingley, & Airolidi (2013)



- Allows for topic correlations and also allows observed *metadata* into the model via a *logistic normal linear regression* involving covariates X and factors Y
- Harder to estimate: no conjugacy, plus γ , Σ , β may vary per document (no longer global)

Structural topic models

How does it work?

- Basically, given documents exhibiting different values of \mathbf{X}, \mathbf{Y} , word occurrences are allowed to vary according to some GLM
- Call m_v the baseline occurrence of word \mathcal{V}_v in the corpus, and $\kappa^{(t)}, \kappa^{(c)}, \kappa^{(i)}$ the log deviations for topics, covariates, and interactions:

$$\beta_{k,d,v} = \frac{\exp \left(m_v + \kappa_{k,v}^{(t)} + \kappa_{y_d,v}^{(c)} + \kappa_{y_d,k,v}^{(i)} \right)}{\sum_v \exp \left(m_v + \kappa_{k,v}^{(t)} + \kappa_{y_d,v}^{(c)} + \kappa_{y_d,k,v}^{(i)} \right)}$$

- The $\boldsymbol{\theta}_d$ are allowed to have different mean vectors $\gamma_{1:K}$ per topic:

$$\begin{aligned} \boldsymbol{\Gamma} &= [\gamma_1, \dots, \gamma_K]_{P, K-1} \\ \boldsymbol{\theta}_d &\sim \text{LN}(\boldsymbol{\Gamma}' \mathbf{X}'_d, \boldsymbol{\Sigma}) \end{aligned}$$

Data analysis

Dataset

- Corpus: 182,308 stats.stackexchange.com posts made between Feb 2, 2009 and Dec 16, 2016 (92,335 questions, 89,973 answers)

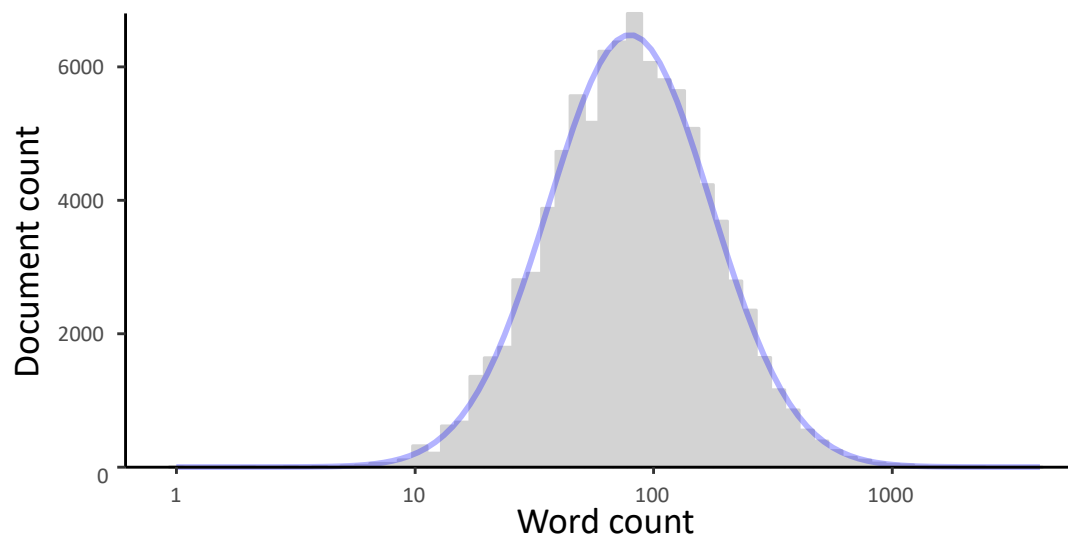
```
<?xml version="1.0" encoding="utf-8"?>
<posts>
  <row Id="1" PostTypeId="1" AcceptedAnswerId="15" CreationDate="2010-07-19T19:12:12.510" Score="36" ViewCount="2186"
  Body="&lt;p&gt;How should I elicit prior distributions from experts when fitting a Bayesian model?&lt;/p&gt;&#xA;"
  OwnerUserId="8" LastActivityDate="2010-09-15T21:08:26.077" Title="Eliciting priors from experts"
  Tags="&lt;bayesian&gt;&lt;prior&gt;&lt;elicitation&gt;" AnswerCount="5" CommentCount="1" FavoriteCount="23" />
  <row Id="2" PostTypeId="1" AcceptedAnswerId="59" CreationDate="2010-07-19T19:12:57.157" Score="29" ViewCount="20123"
  Body="&lt;p&gt;In many different statistical methods there is an &quot;assumption of normality&quot;. What is
  &quot;normality&quot; and how do I know if there is normality?&lt;/p&gt;&#xA;" OwnerUserId="24" LastEditorUserId="88"
  LastEditDate="2010-08-07T17:56:44.800" LastActivityDate="2016-06-27T06:44:40.147" Title="What is normality?"
  Tags="&lt;distributions&gt;&lt;normality&gt;" AnswerCount="7" CommentCount="1" FavoriteCount="10" />
</posts>
```

- Used an XML parser to extract post contents and metadata
- Documents: 92,335 question titles, questions, and all associated answers

Data analysis

Dataset

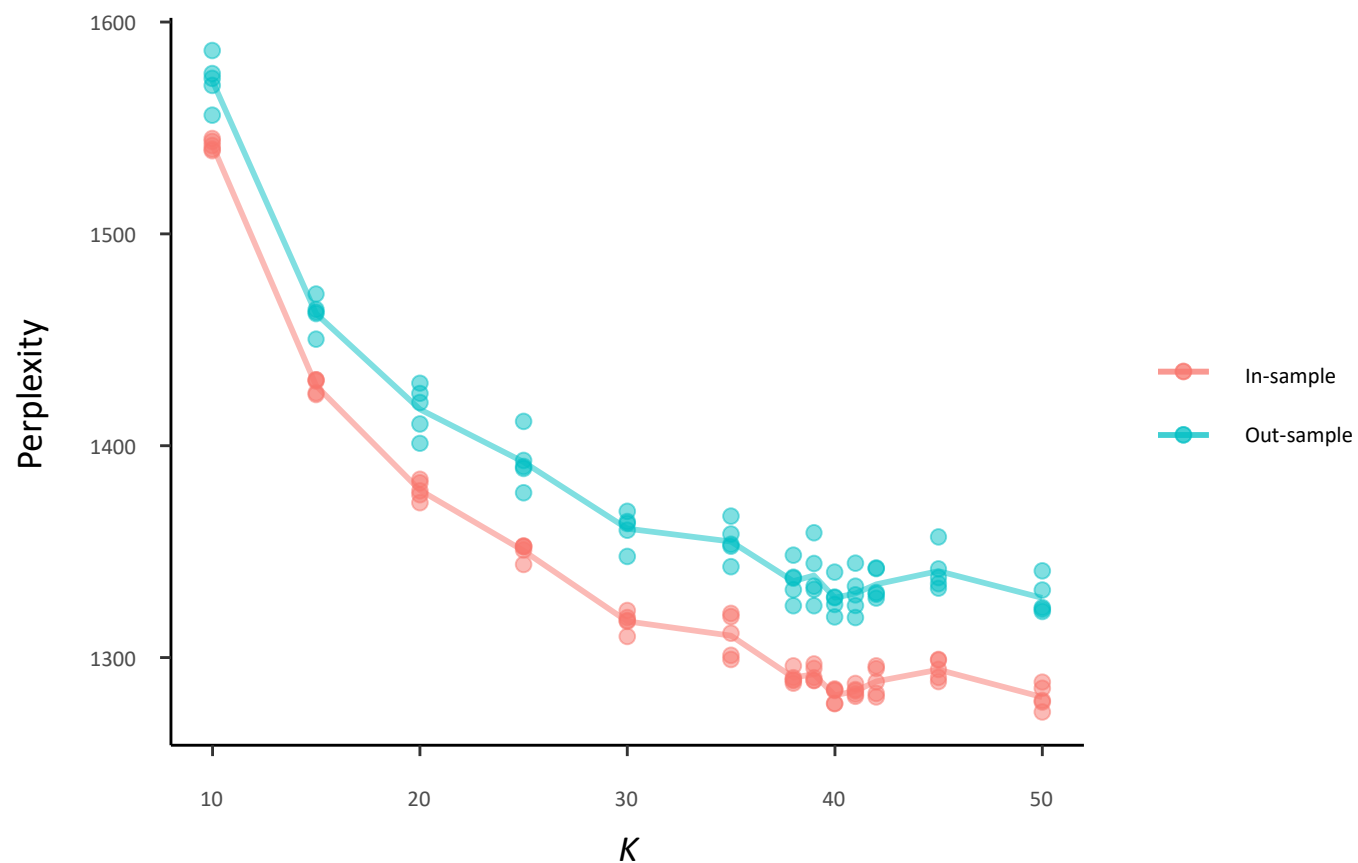
- Vocabulary: deleted code, LaTeX, and *stop words*
- Removed any words appearing less than 30 times
- Original $N = 20,594,592$ words, $V = 121,804$ vocabulary
- Cleaned $N = 10,011,781$ words, $V = 11,905$ vocabulary



Data analysis

Cross-validation to select K

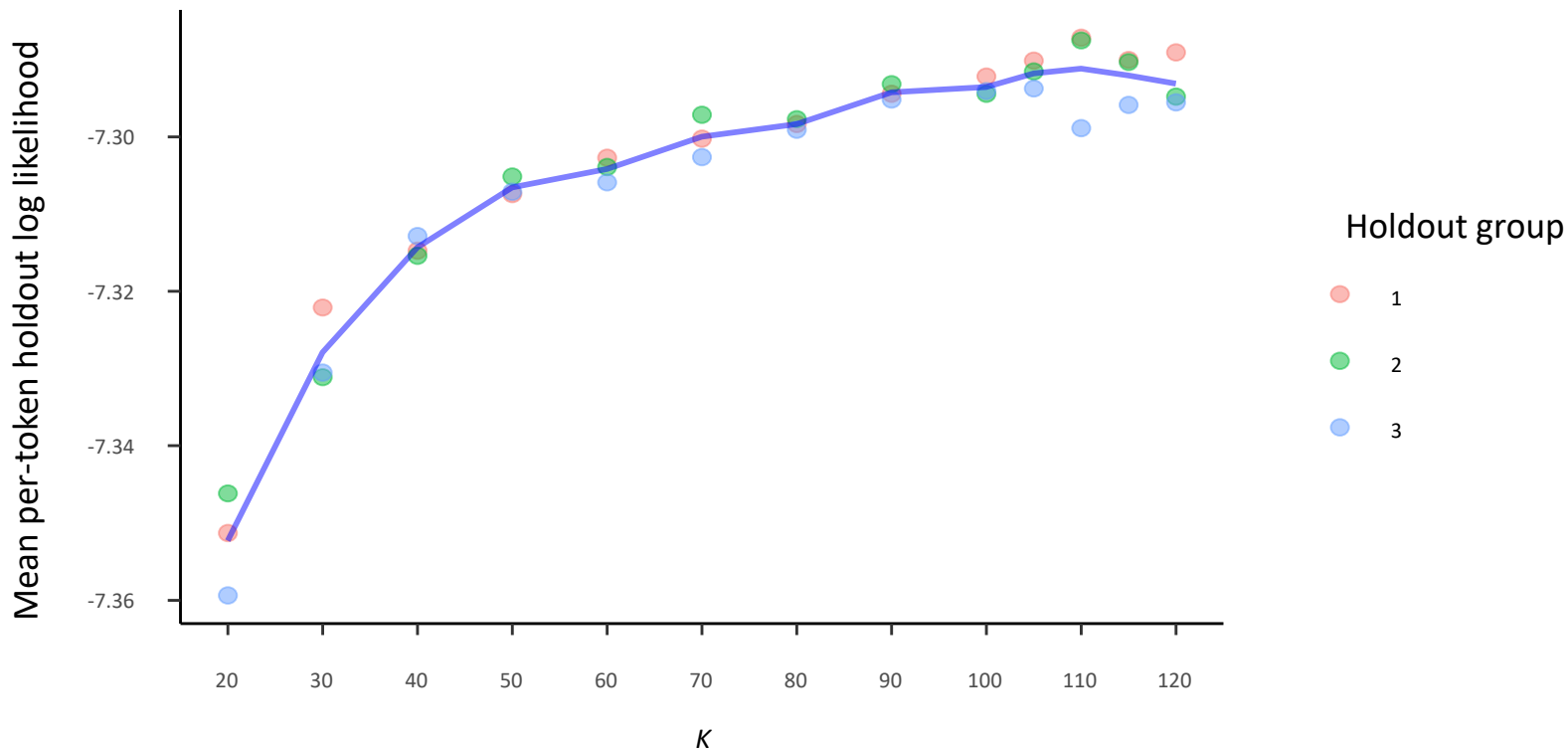
- LDA perplexity minimized at $K = 40$ (function of log likelihood)



Data analysis

Cross-validation to select K

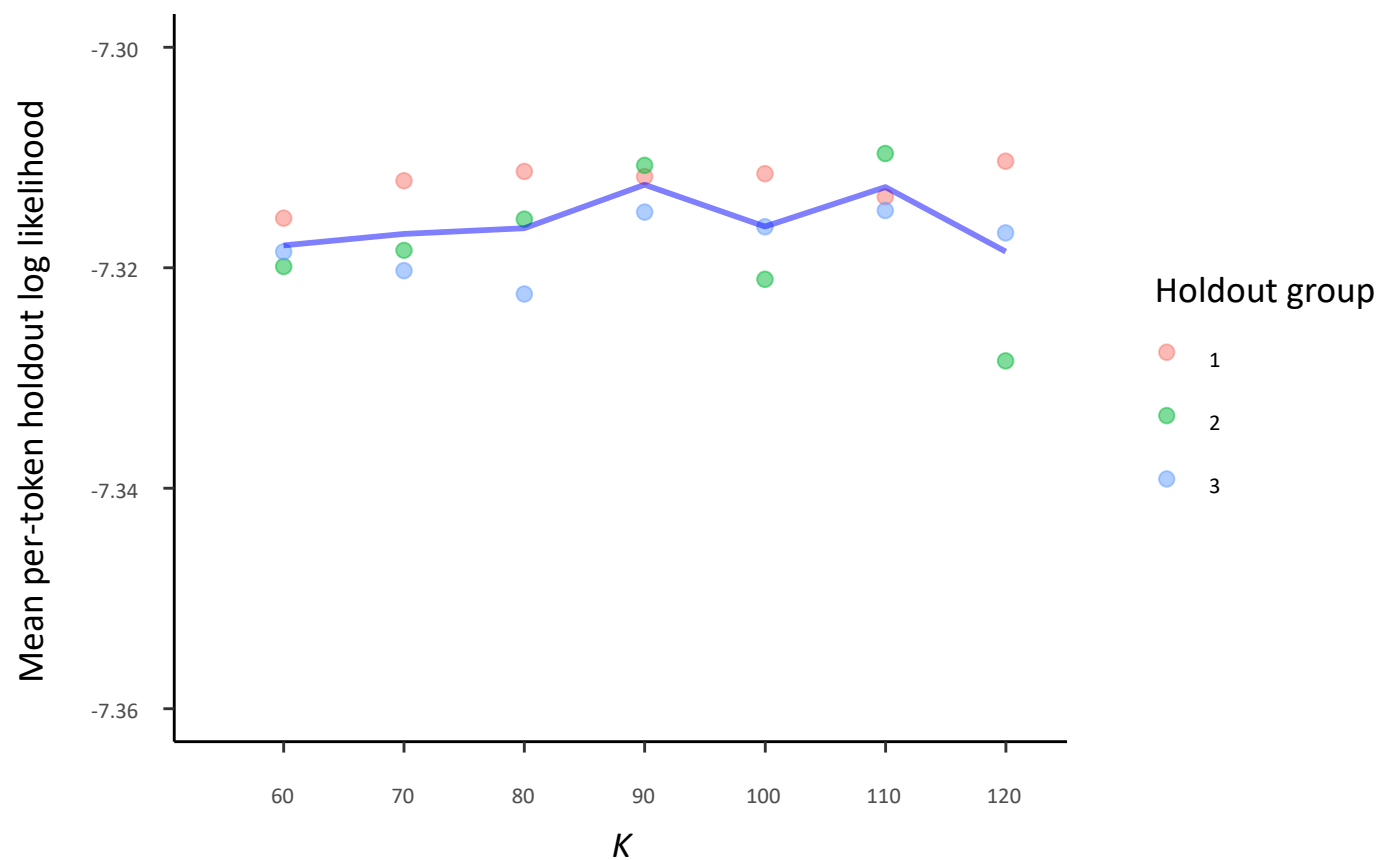
- CTM per-word holdout log likelihood maximized at $K = 110$



Data analysis

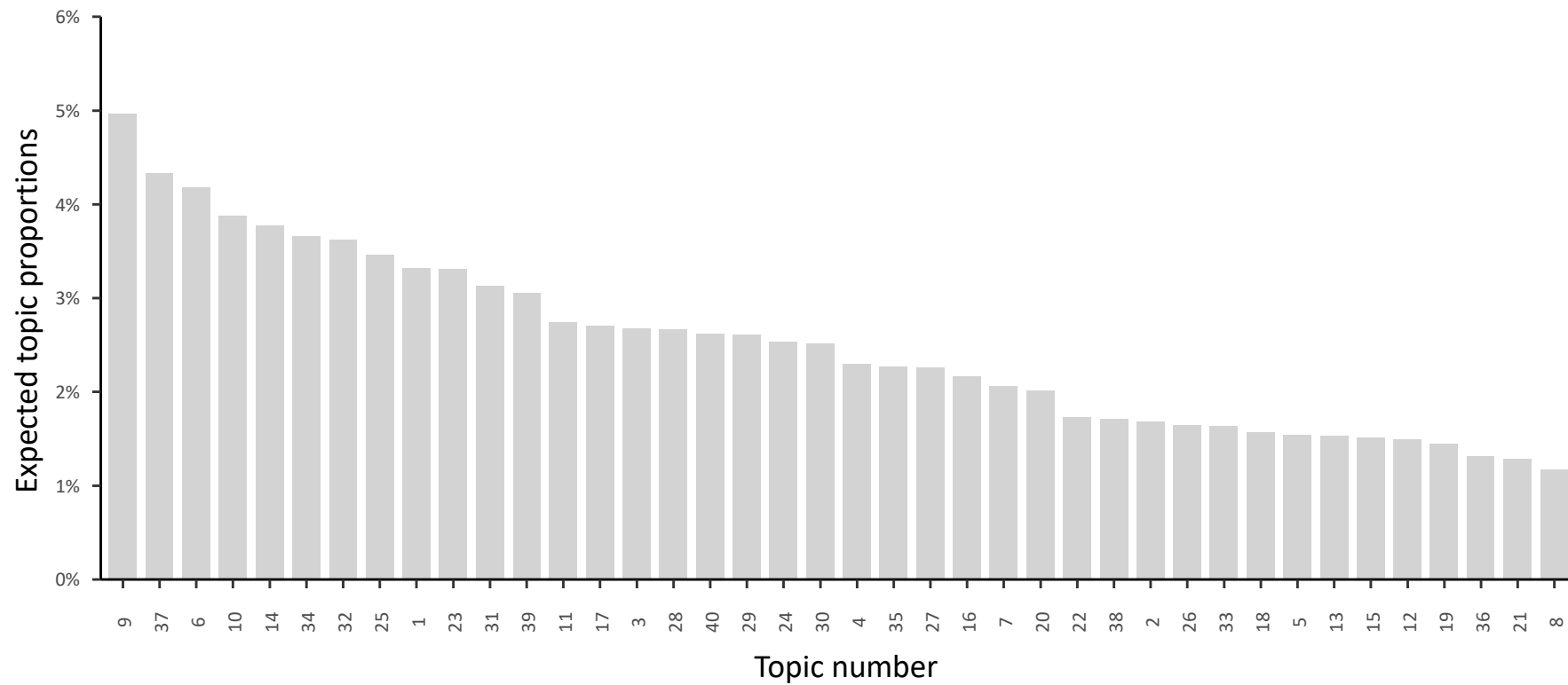
Cross-validation to select K

- STM per-word log likelihood maximized at $K = 90$



Data analysis

Topic proportions $E(\theta_d)$ (LDA)



Data analysis

Topic proportions $E(\theta_d)$ (LDA)

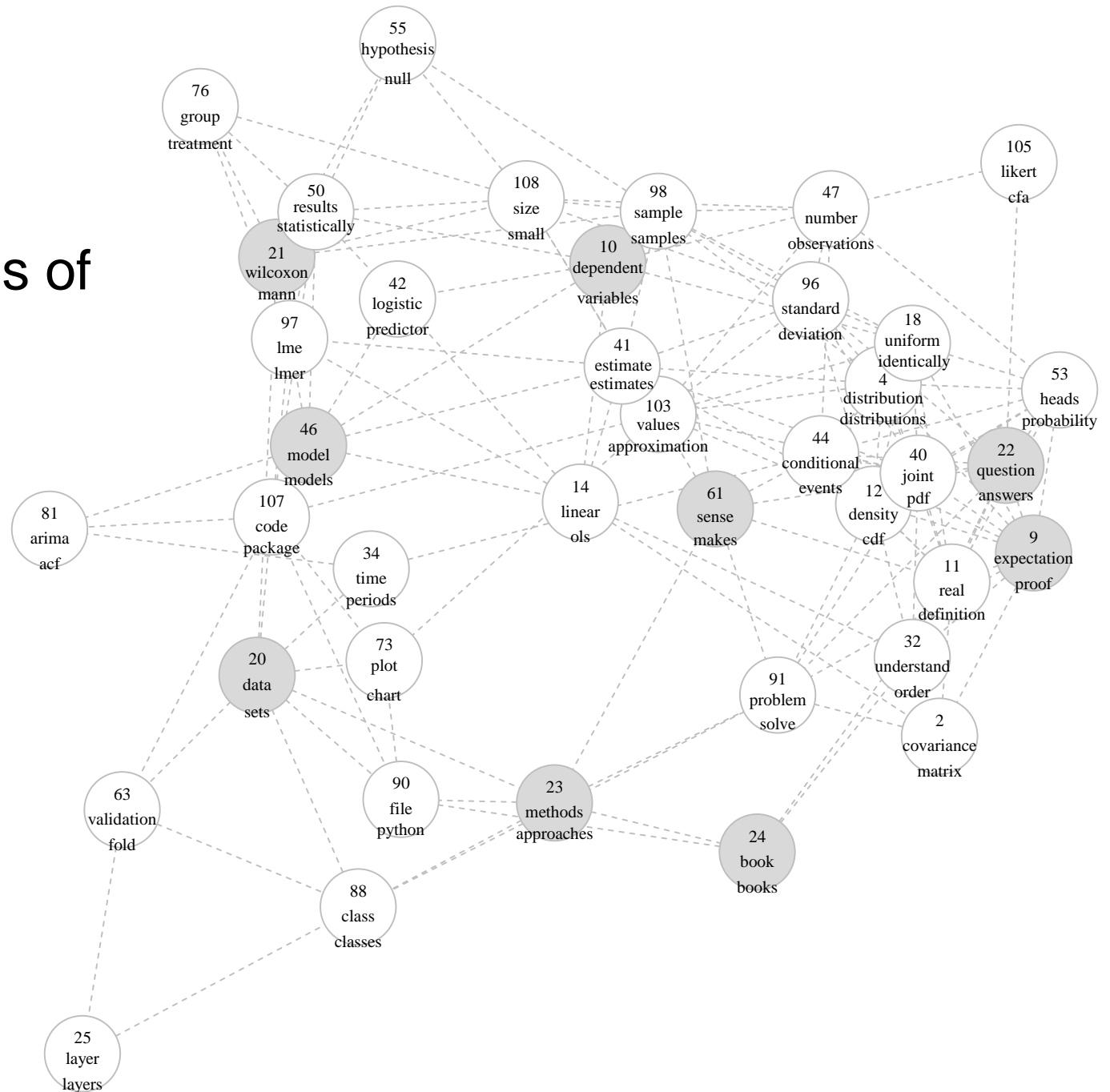
Suggestions for labels?

Distributions	Linear Models	Commentary on Posts	Data	Hypothesis Testing	Model Selection	Coding	Prediction	Experimental Design
9	37	6	10	14	34	32	25	1
random function	variables variable	question answer	data set	test hypothesis	model models	code function	class training	group groups
distribution	regression	people	values	tests	fit	package	set	anova
variables variable	logistic dependent	make good	dataset points	null testing	regression linear	values method	features validation	treatment control
probability	independent	sense	observations	significance	residuals	results	cross	design
density	model	problem	sets	significant	data	output	classification	subjects
independent	predictors	results	analysis	difference	parameters	bootstrap	feature	effect
conditional pdf	categorical predictor	important statistical	measurements problem	statistic statistical	fitting fitted	run generate	data classifier	difference post

Data analysis

Topic correlations (CTM)

- Clustering implies groups of related questions?



Data analysis

Covariate effects (STM)

- Chosen covariates: question score (natural spline), question creation date (natural spline), user-added Homework tag (factor)
- STM model observed variables:

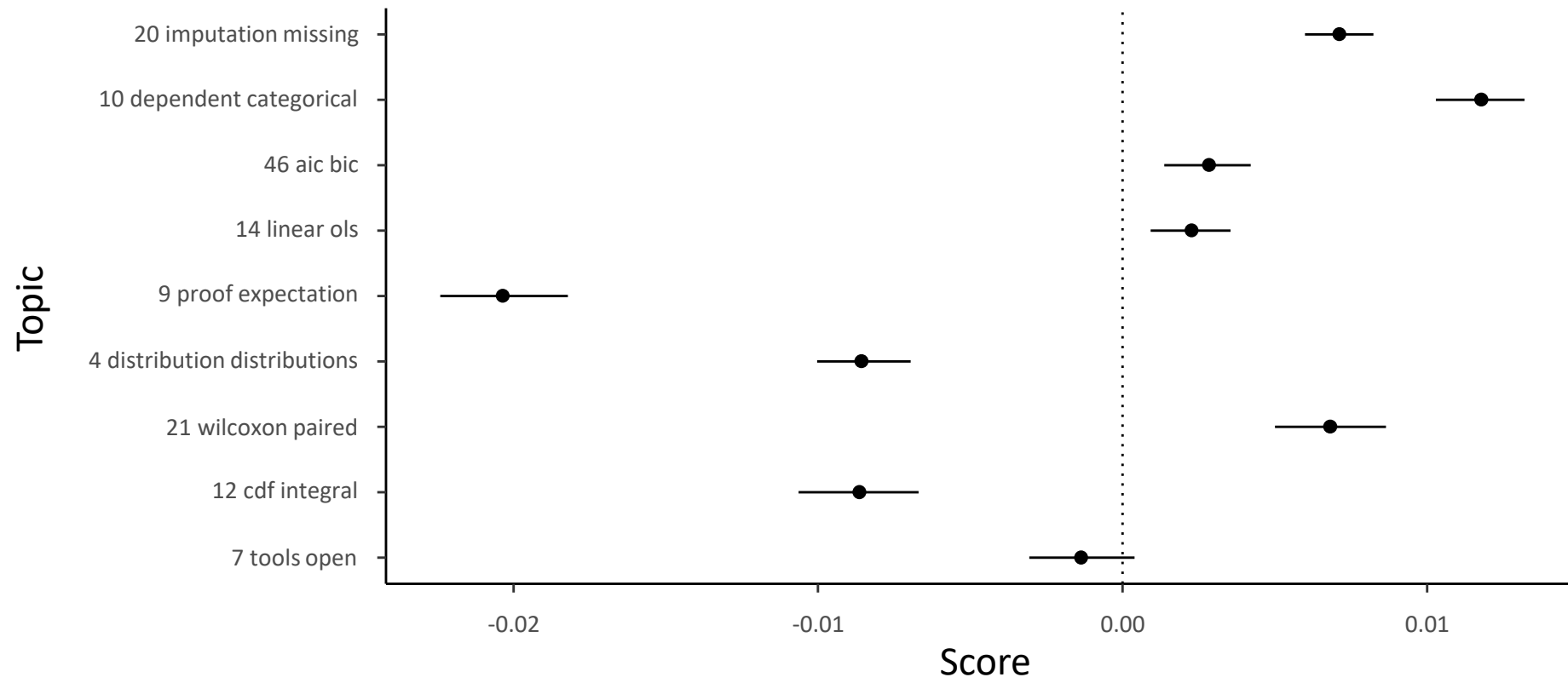
$$\mathbf{X} = \begin{bmatrix} f(\text{Score}_d) & g(\text{CreationDate}_d) & \mathbf{1}\{\text{Tag}_{\text{HW},d}\} \end{bmatrix}_{D,3}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{1}\{\text{Tag}_{\text{HW},d}\} \end{bmatrix}_{D,1}$$

$$\mathbf{w}_{1:D} = \{w_1, \dots, w_{N_d}\}$$

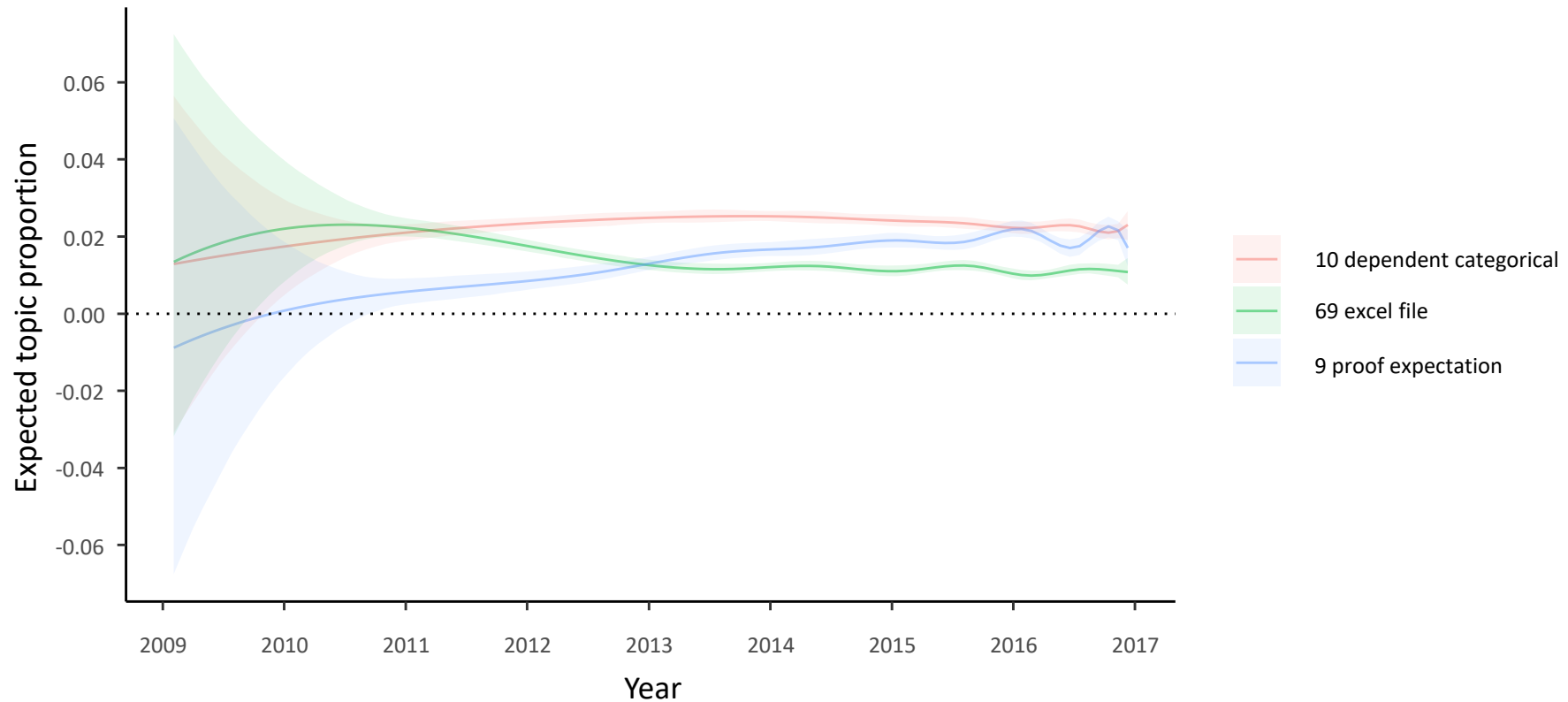
Data analysis

Covariate effects (STM)



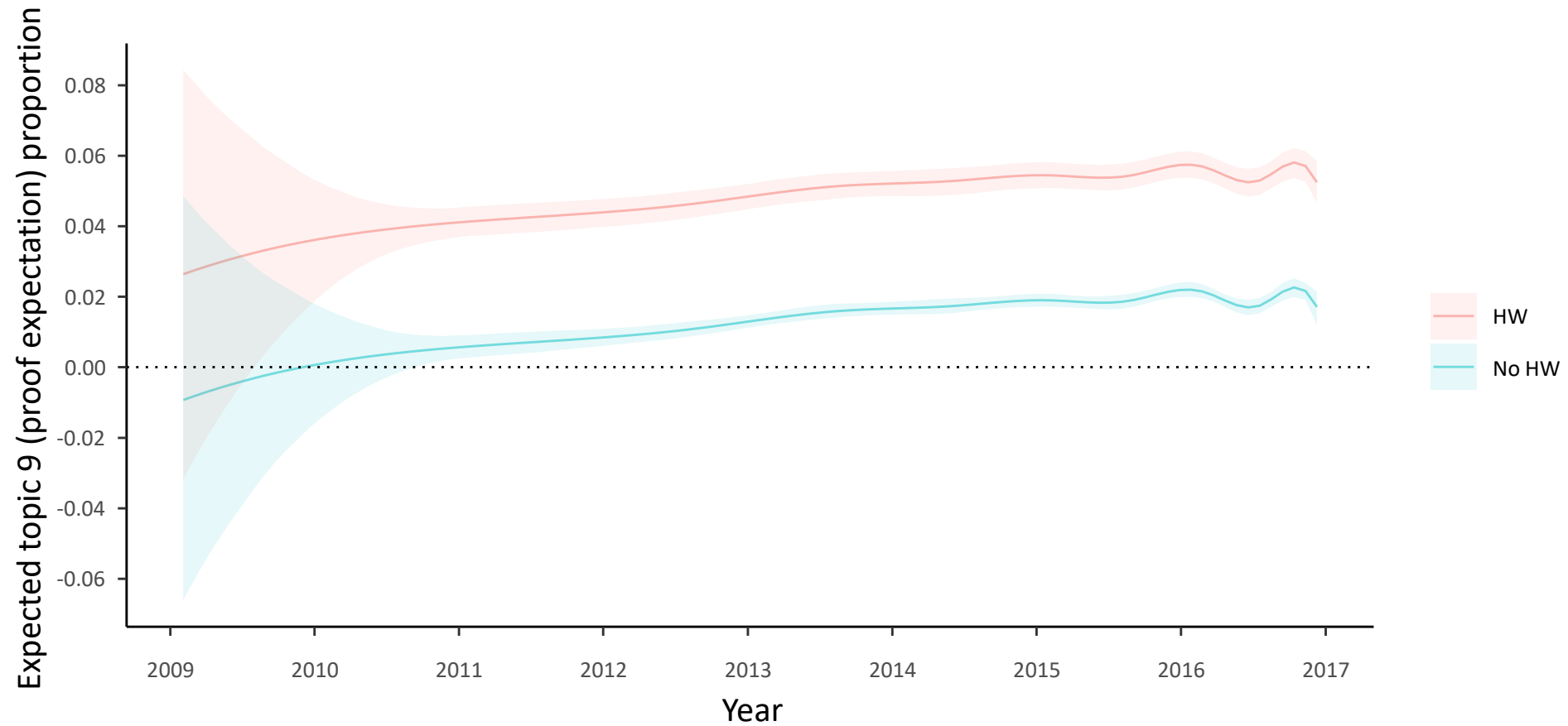
Data analysis

Covariate effects (STM)



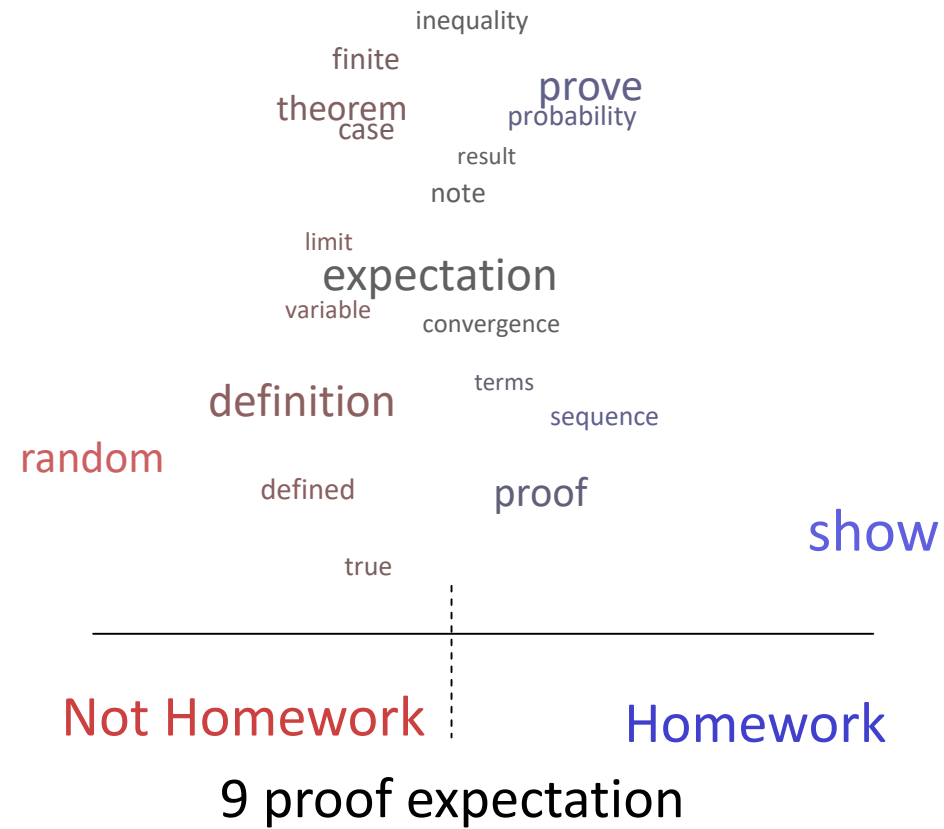
Data analysis

Covariate effects (STM)



Data analysis

Covariate effects (STM)



Data analysis

Model comparison

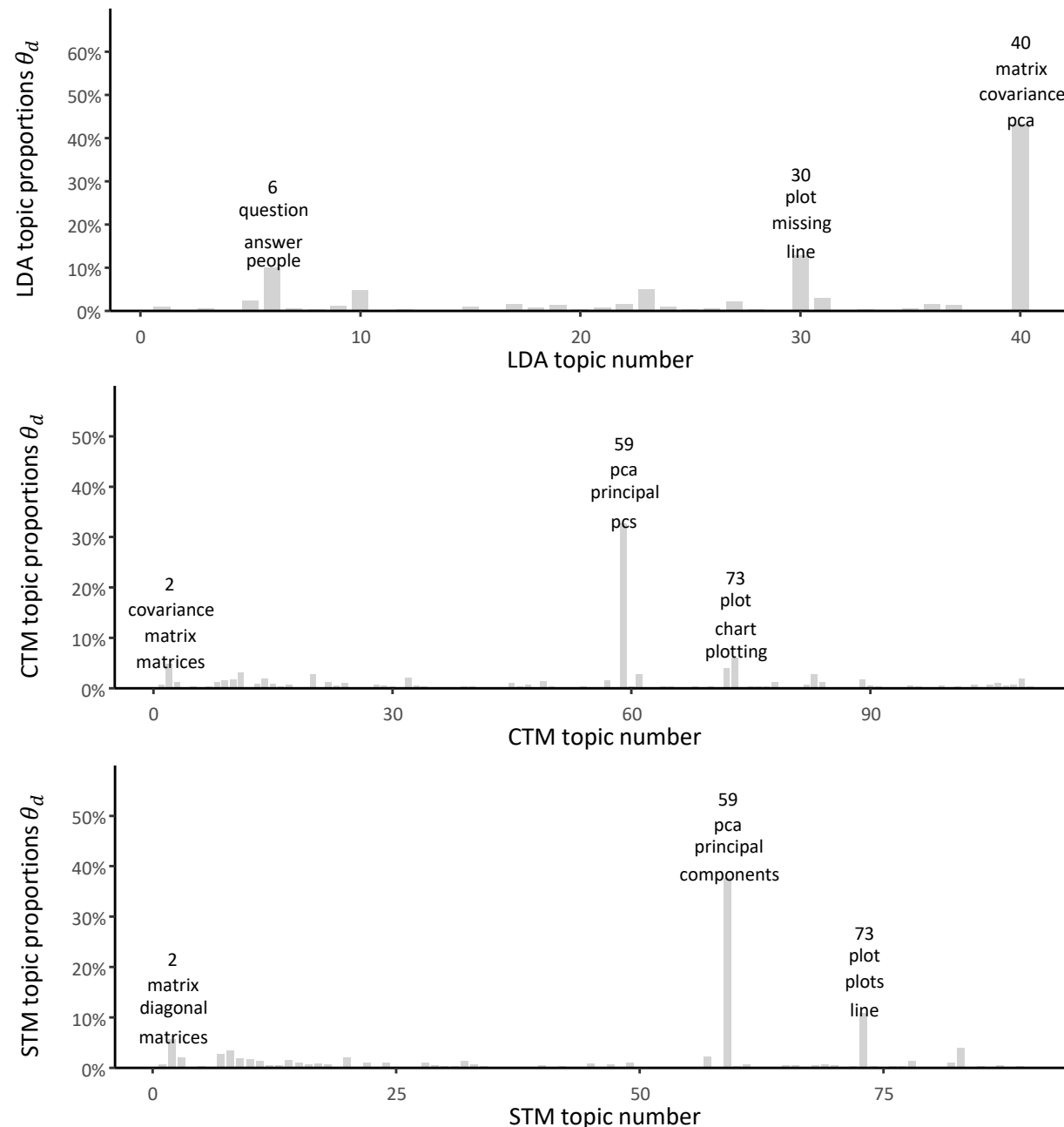
- CTM (110) and STM (90) support more topics than LDA (40), mirroring previous findings
- Models found similar topics, but they are not identical:

LDA		CTM		STM	
10 Data	34 Model Selection	20 Data	46 Model Selection	20 Data	46 Model Selection
data	model	data	model	imputation	aic
set	models	sets	models	missing	bic
values	fit	set	modelling	data	model
dataset	regression	raw	modeling	datasets	fit
points	linear	collected	building	dataset	models
observations	residuals	synthetic	fit	sets	fits
sets	data	consists	aicc	imputed	fitting
analysis	parameters	kind	saturated	impute	aicc
measurements	fitting	generated	full	set	fitted
problem	fitted	apply	fitting	handle	reml

Data analysis

Model comparison

- Most popular post, [2691](#)
- “Explain PCA to grandma?”
- Similar but not identical results:



Data analysis

Model comparison

- Chang, Boyd-Graber, Gerrish, Wang, & Blei (2009): more complicated models associated with better fit, *but not greater interpretability*
- Which topic model variant is best? Depends on the objective

Thank you!

Questions?

A few references

- Arora, Ge, Halpern, Mimno, Moitra, Sontag, Wu, & Zhu (2013). A practical algorithm for topic modeling with provable guarantees.
- Blei & Lafferty (2007). A correlated topic model of Science.
- Blei, Ng, & Jordan (2003). Latent Dirichlet allocation.
- Chang, Boyd-Graber, Gerrish, Wang, & Blei (2009). Reading tea leaves: how humans interpret topic models.
- Roberts, Stewart, & Airolidi (2016). A model of text for experimentation in the social sciences.
- Roberts, Stewart, Tingley, Lucas, Leder-Luis, Gadarian, Albertson Rand (2014). Structural topic models for open-ended survey responses.
- Wallach, Murray, Salakhutdinov, & Mimno (2009). Evaluation methods for topic models.