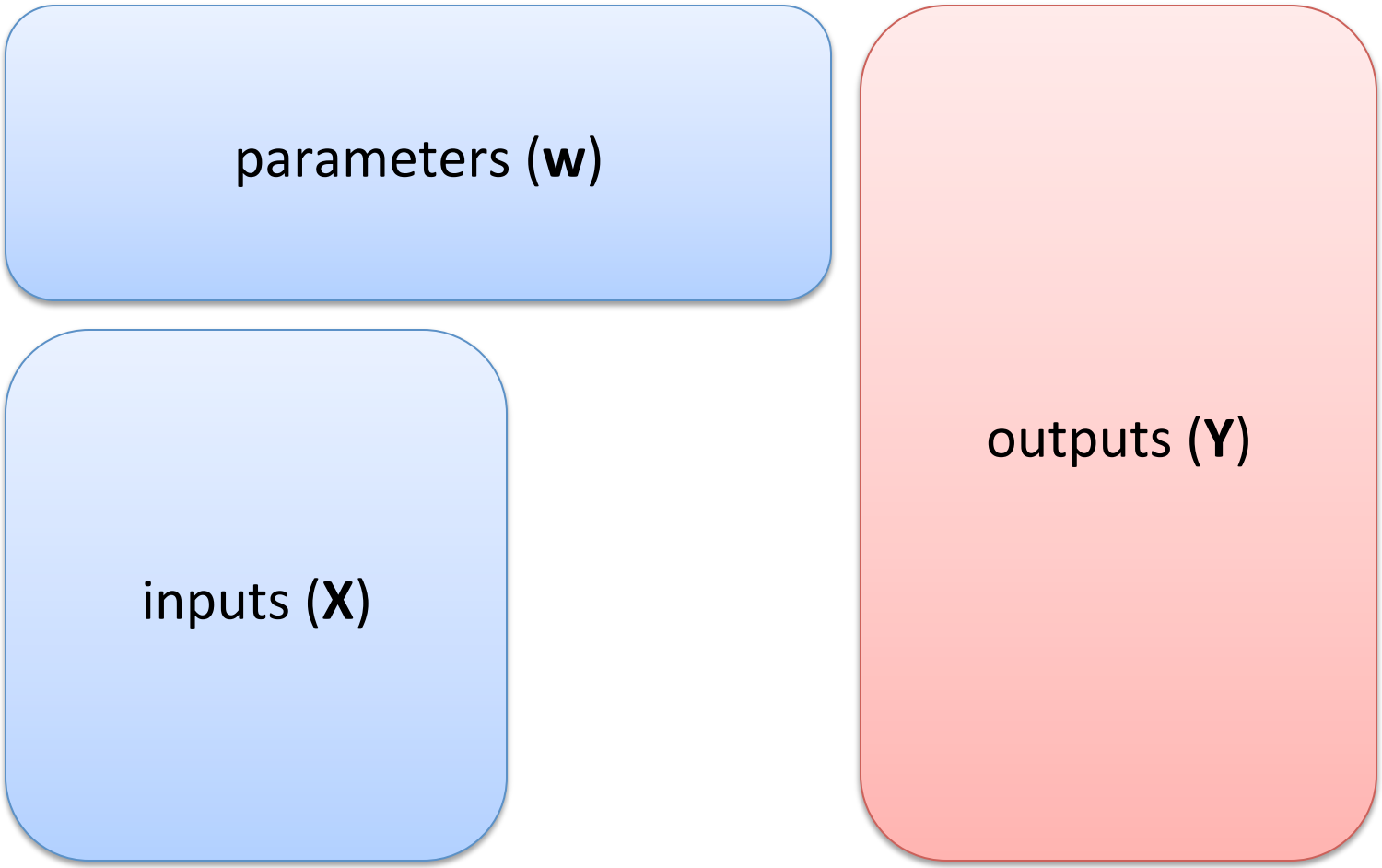
Probability and Structure in Natural Language Processing

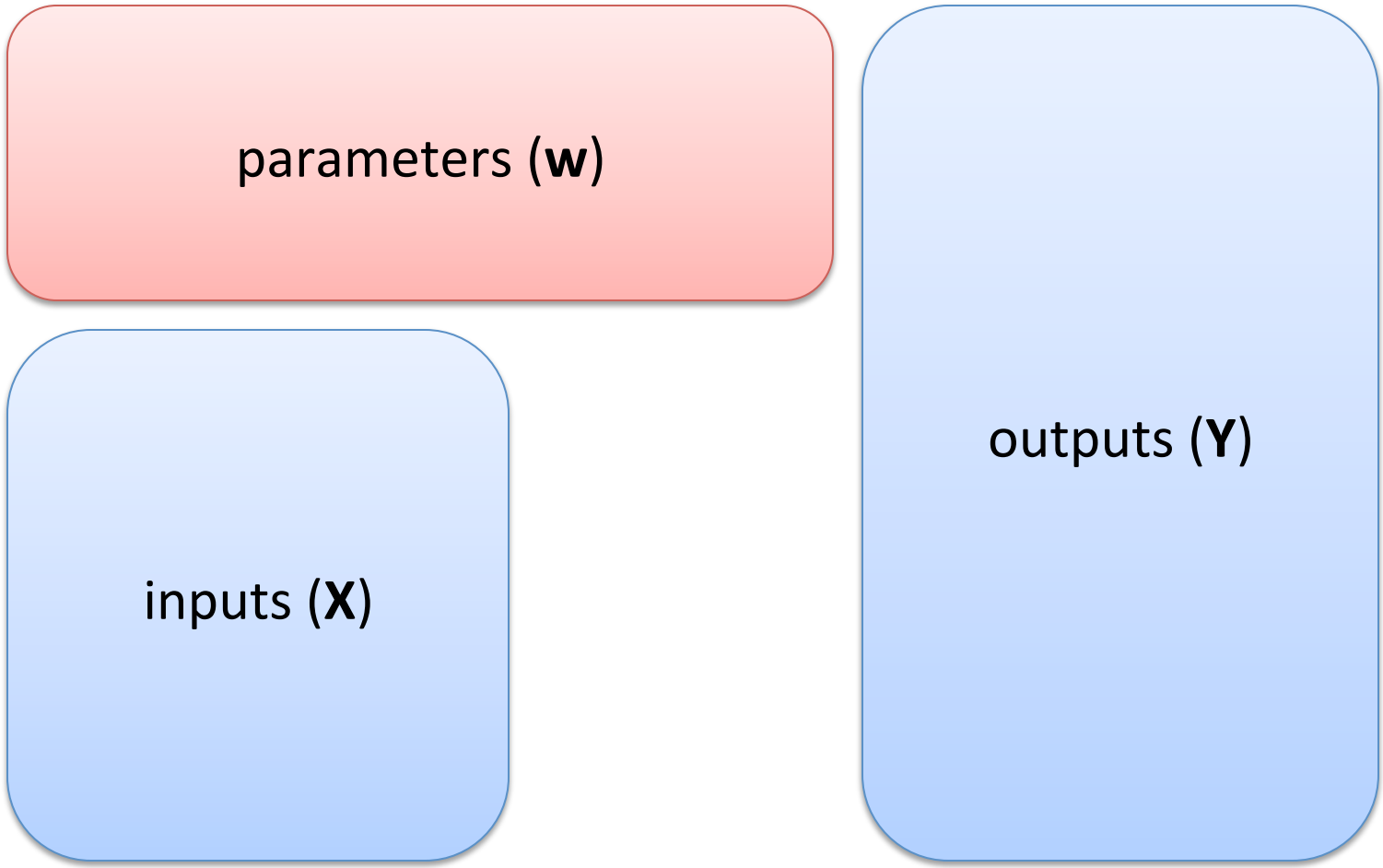
Noah Smith

Heidelberg University, November 2014

Random Variables in Decoding



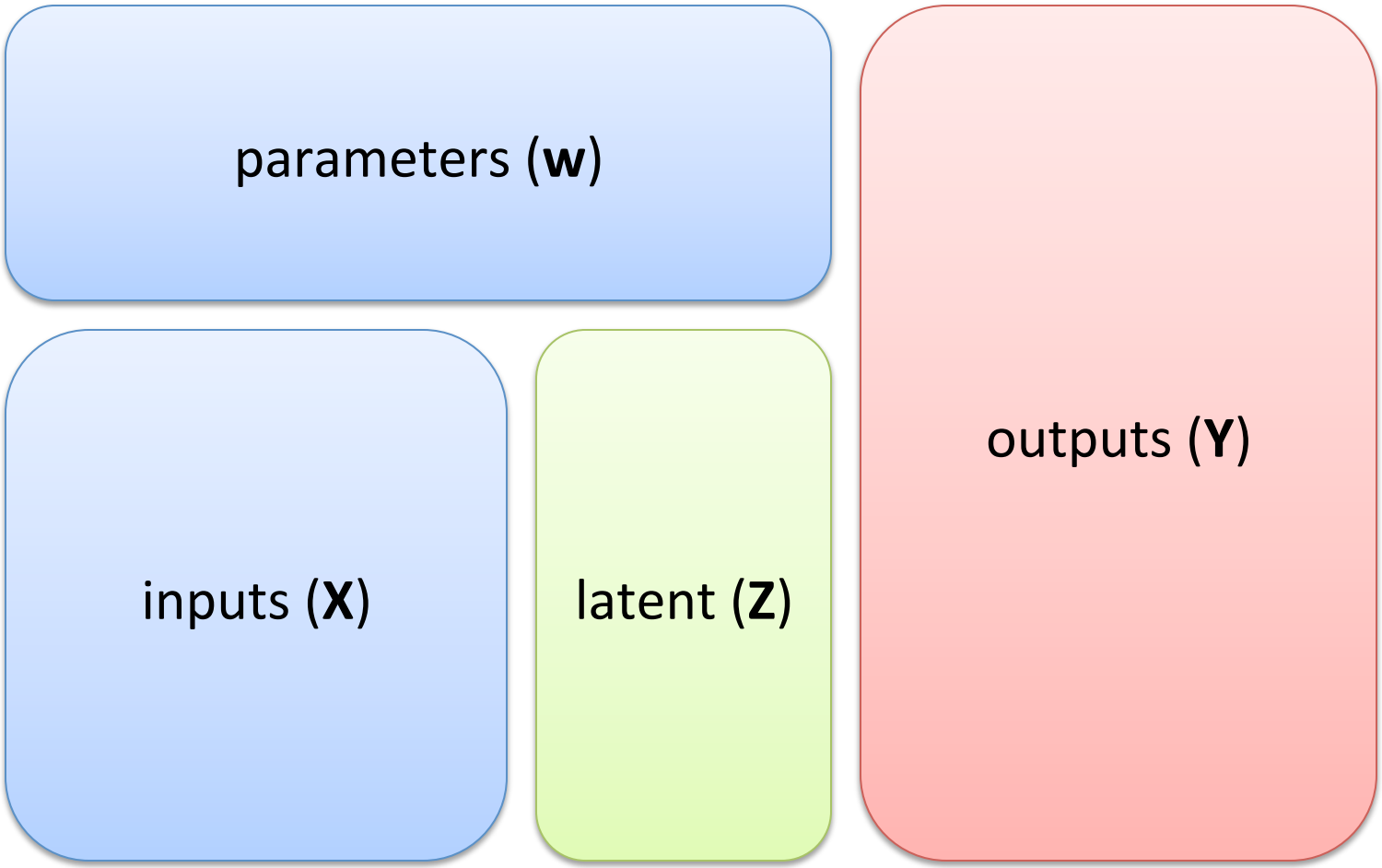
Random Variables in Supervised Learning



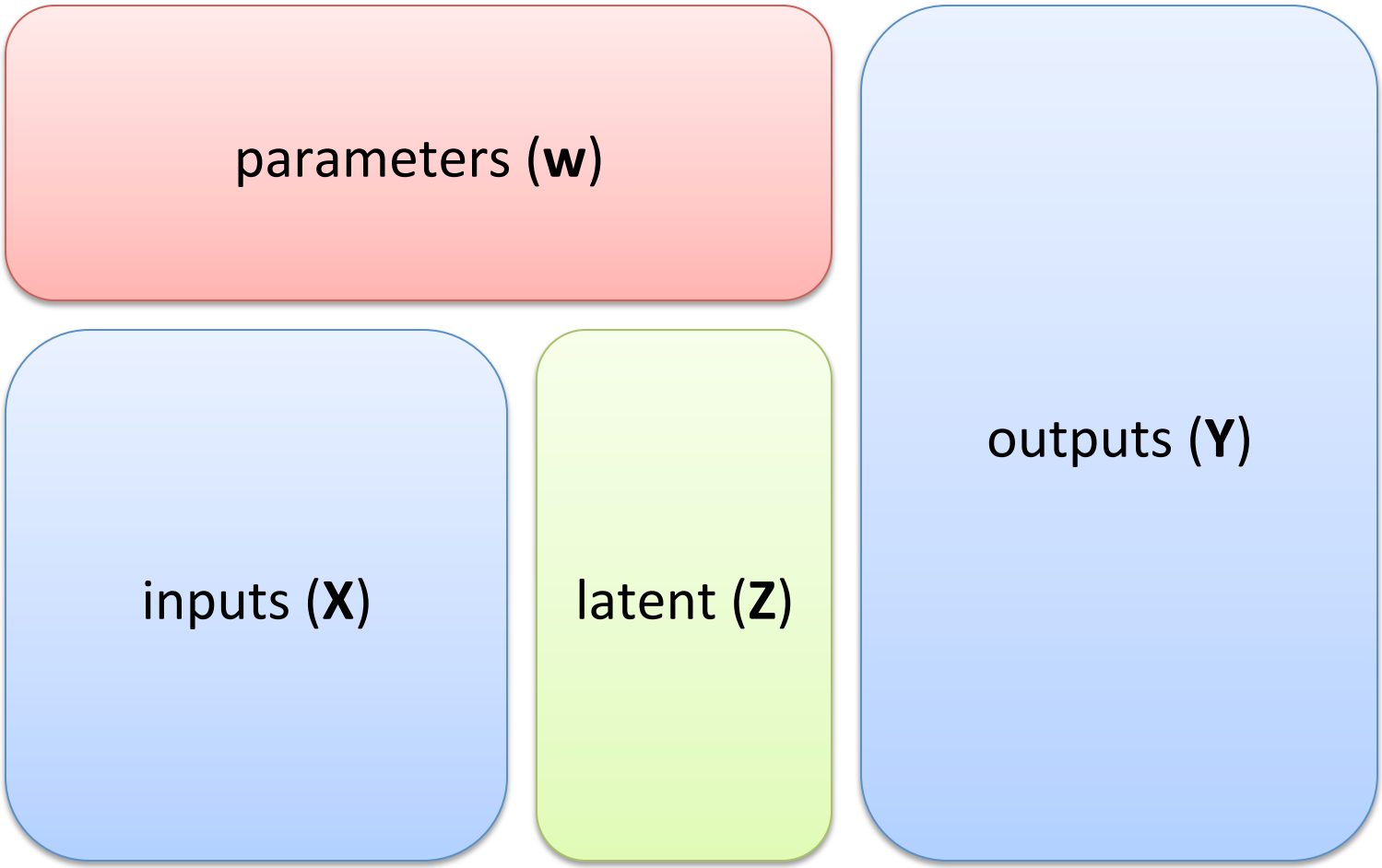
Hidden Variables are Different

- We use the term "hidden variable" (or "latent variable") to refer to something we *never* see.
  - Not even in training.
  - Sometimes we believe they are real.
  - Sometimes we believe they only approximate reality.

Random Variables in Decoding



Random Variables in Supervised Learning



Latent Variables and Inference

- Both learning and decoding can be still be understood as inference problems.
- Usually "mixed":
  - some variables are getting maximized
  - some variables are getting summed

Word Alignments

- Since IBM model 1, word alignments have been the prototypical hidden variable.
- Ultimately, in translation, we do not care what they are.
- Current approach: learn the word alignments unsupervised, then fix them to their most likely values.
  - Then construct models for translation.
- Alignment on its own: unsupervised problem.
- MT on its own: supervised problem.
- MT + alignment: supervised problem with latent variables.

Alignments in Text-to-Text Problems

- Wang et al. (2007): "Jeopardy" model for answer ranking in QA.
  - Align questions to answers.
  - Similar model for paraphrase detection (Das and Smith, 2009)

Latent Annotations in Parsing

- Treebank categories (N, NN, NP, etc.) are too coarse-grained.
  - Lexicalization (Collins, Eisner)
  - Johnson's (1998) parent annotation
  - Klein and Manning (2003) parser
- Treat the true, fine-grained category as hidden, and infer it from data.
  - Matsuzaki, Petrov, Dreyer, many others.

Richer Formalisms

- Cohn et al. (2009): tree substitution grammar.
  - Derived tree is observed (output variable).
  - Derivation tree (segmentation into elementary trees) is hidden.
- Zettlemoyer and Collins (2005 and later): infer CCG syntax from first-order logical expressions and sentences.
- Liang et al. (2011): infer semantic representation from text and database.

Topic Models

- Infer topics (or topic blends) in documents.
- Latent Dirichlet allocation (Blei et al., 2003) is a great example.
  - Sometimes augmented with an output variable (Blei and McAuliffe, 2007) – "supervised" LDA.
  - Many extensions!

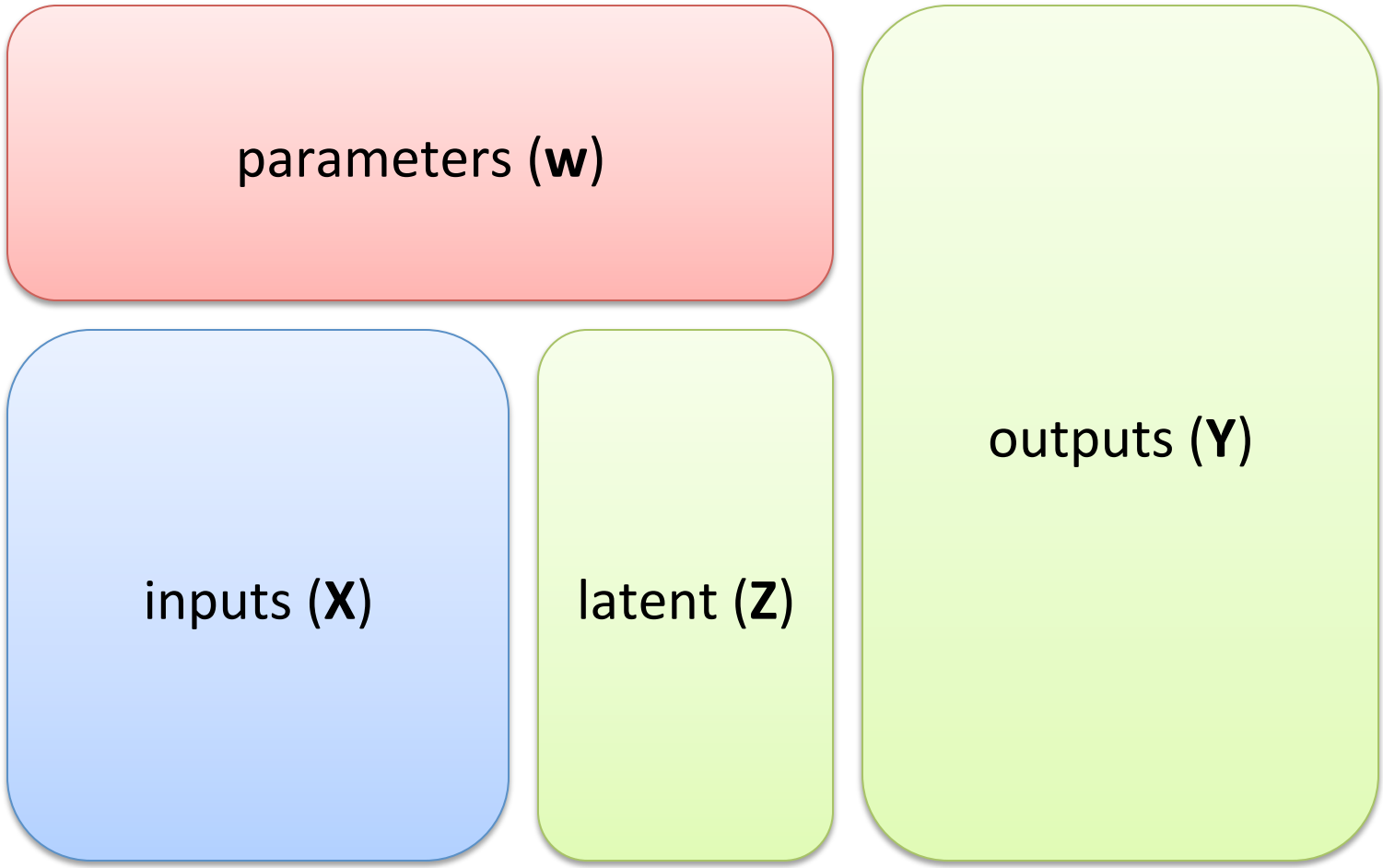
Unsupervised NLP

- Clustering (Brown, 1992, many more)
- POS tagging (Merialdo, 1994, many more)
- Parsing (Pereira and Schabes, Klein and Manning, …)
- Segmentation (word – Goldwater; discourse – Eisenstein)
- Morphology
- Lexical semantics
- Syntax-semantics correspondences
- Sentiment analysis
- Coreference resolution
- Word, phrase, and tree alignment

Supervised or Unsupervised?

- Depends on the task, not the model.
  - I say "unsupervised" when the output variables are hidden at training time.

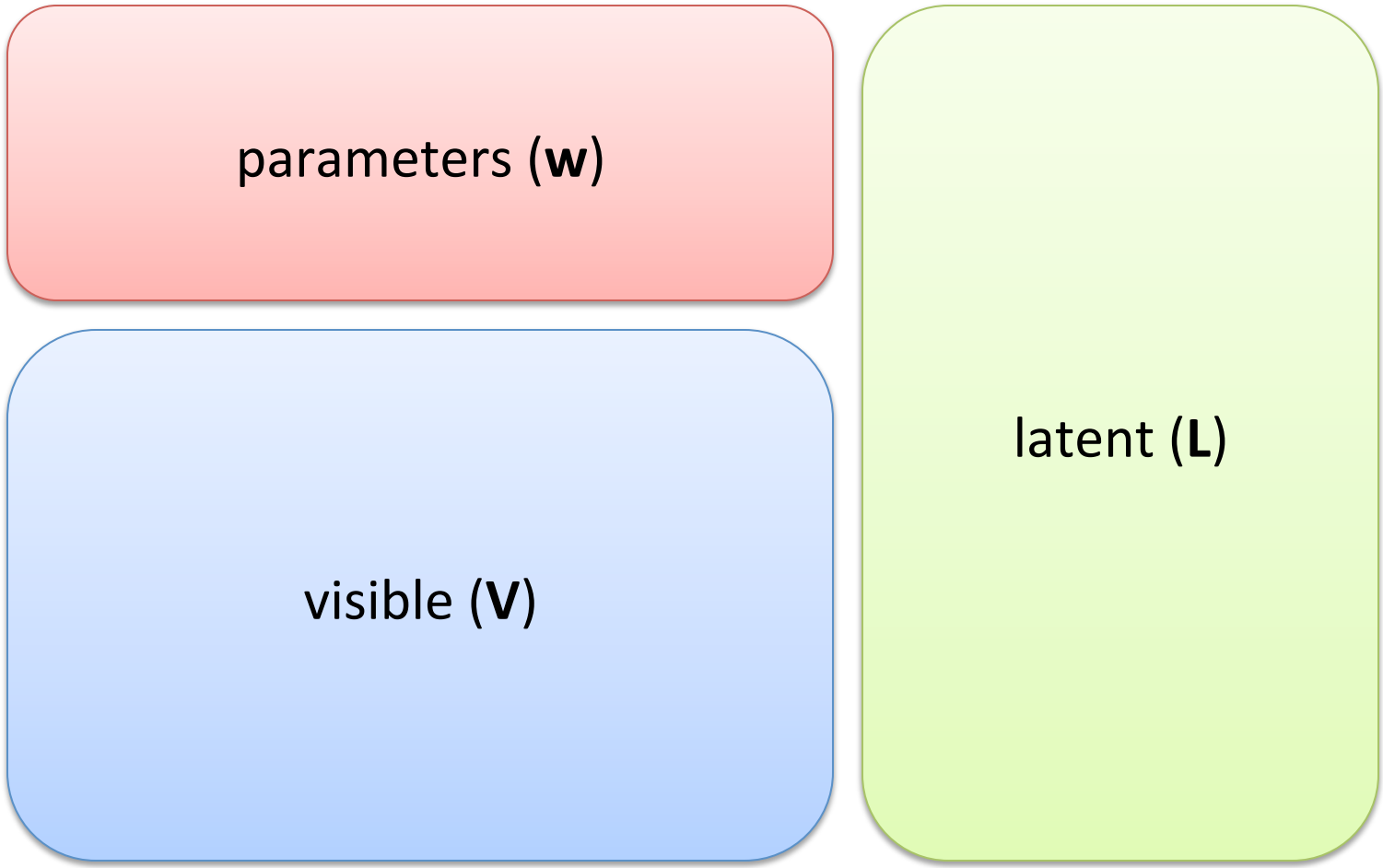
Random Variables in Unsupervised Learning



Probabilistic View

- The usual starting point for hidden variables is maximum likelihood.
  - "Input" and "output" do not matter; only observed/latent.

Random Variables in General Probabilistic Modeling



Empirical Risk View

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{N}\sum_i loss(\boldsymbol{v}_i; h_{\mathbf{w}}) + R(\mathbf{w})$$

$$\begin{aligned} loss(\boldsymbol{v}; h_{\mathbf{w}}) &= -\log p_{\mathbf{w}}(\boldsymbol{v}) \\ &= -\log \sum_{\boldsymbol{\ell}} p_{\mathbf{w}}(\boldsymbol{v}, \boldsymbol{\ell}) \end{aligned}$$

- Log-loss
  - Equates to maximum *marginal* likelihood (or MAP if R(**w**) is a negated log prior).
  - Unlike loss functions in lecture 3, this is not convex!
  - EM seeks to solve this problem (but it's not the only way).
  - Regularization decisions are orthogonal.

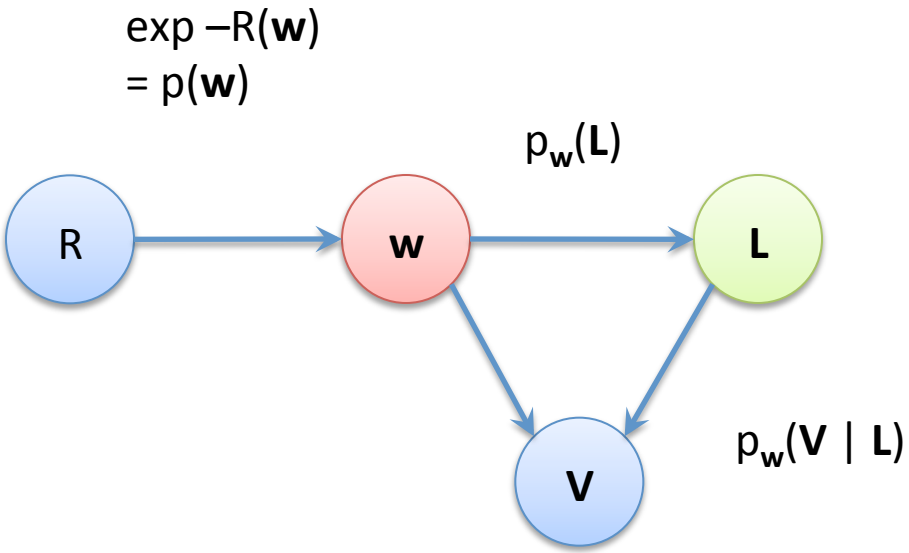
Optimizing the Marginal Log-Loss

- EM as inference
- EM as optimization
- Direct optimization

Generic EM Algorithm

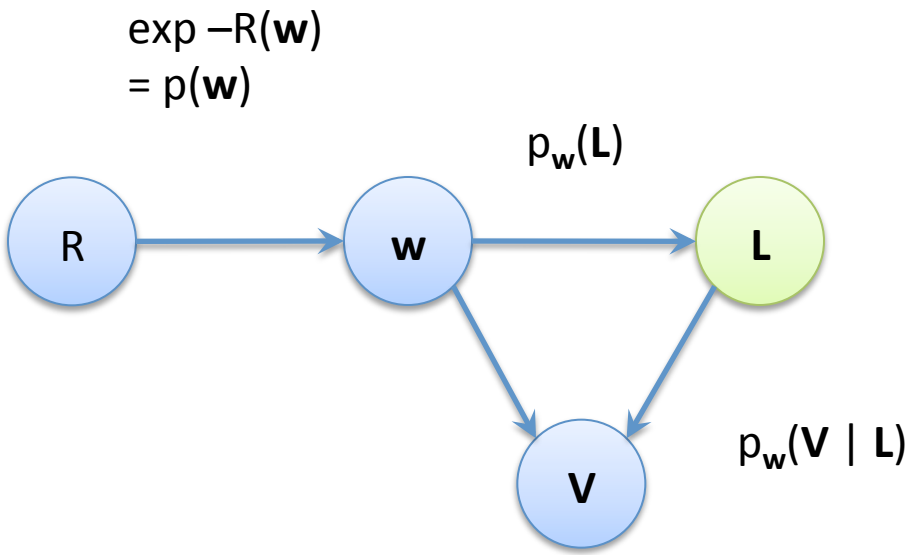
- Input: $\mathbf{w}^{(0)}$ and observations $\mathbf{v}_1$, $\mathbf{v}_2$, … $\mathbf{v}_N$
- Output: learned $\mathbf{w}$
- t = 0
- Repeat until $\mathbf{w}^{(t)} \approx \mathbf{w}^{(t-1)}$:
  - E step: $\forall i, \forall \boldsymbol{\ell}, \quad q_i^{(t)}(\boldsymbol{\ell}) \leftarrow p_{\mathbf{w}^{(t)}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i)$
  - M step: $\mathbf{w}^{(t+1)} \leftarrow \max_{\mathbf{w}} \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell})$
  - ++t
- Return $\mathbf{w}^{(t)}$

MAP Learning as a Graphical Model

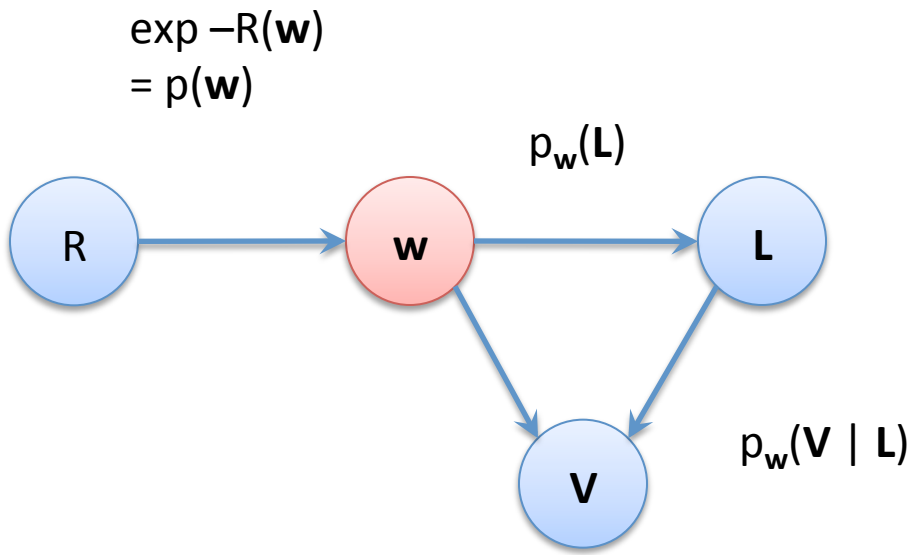


- Combined inference (max over **w**, sum over **L**) is very hard.
  - If **w** were fixed, getting the posterior over **L** wouldn't be so bad.
  - If **L** were fixed, maximizing over **w** wouldn't be so bad.

MAP Learning as a Graphical Model



E step

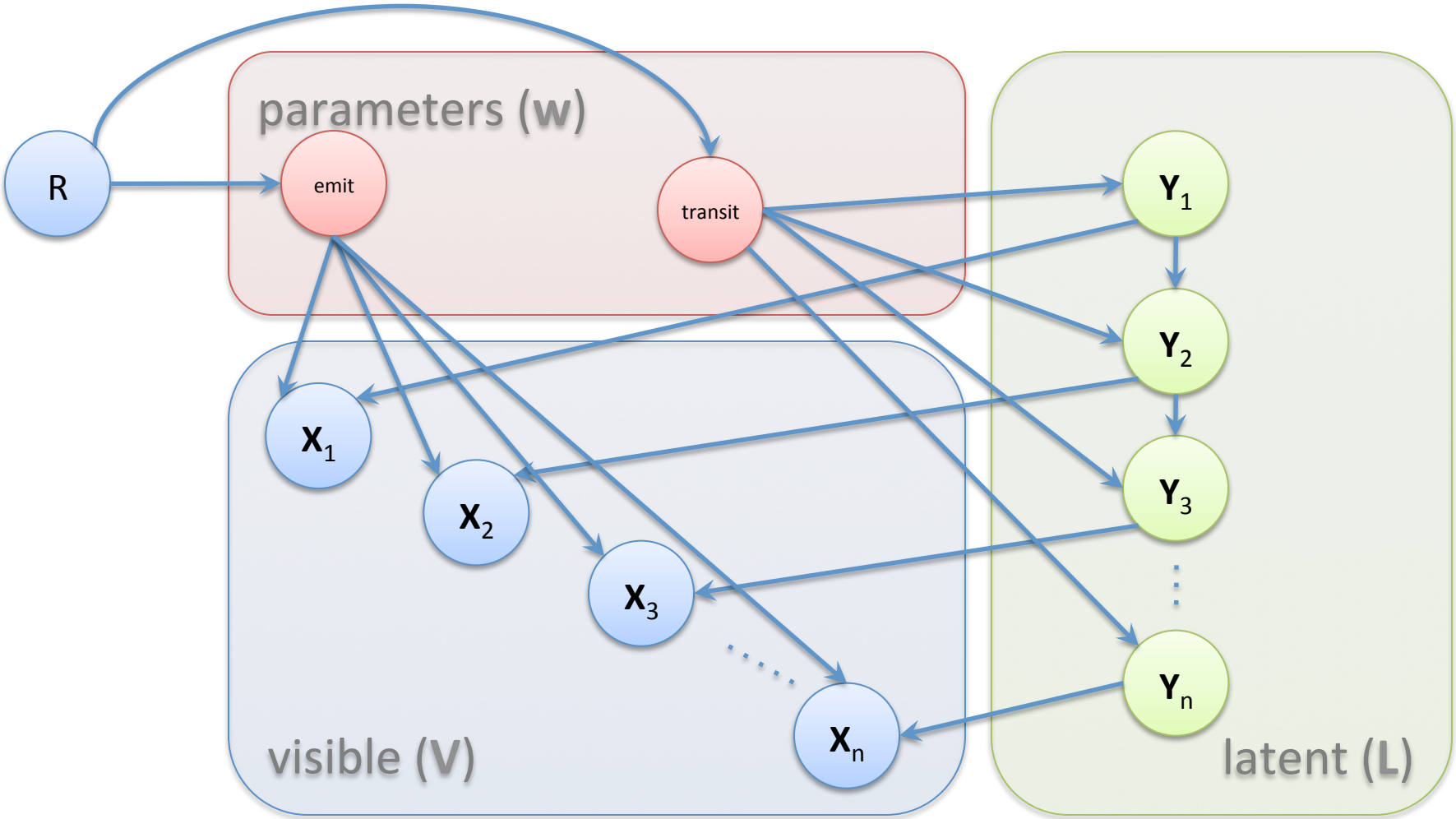


M step

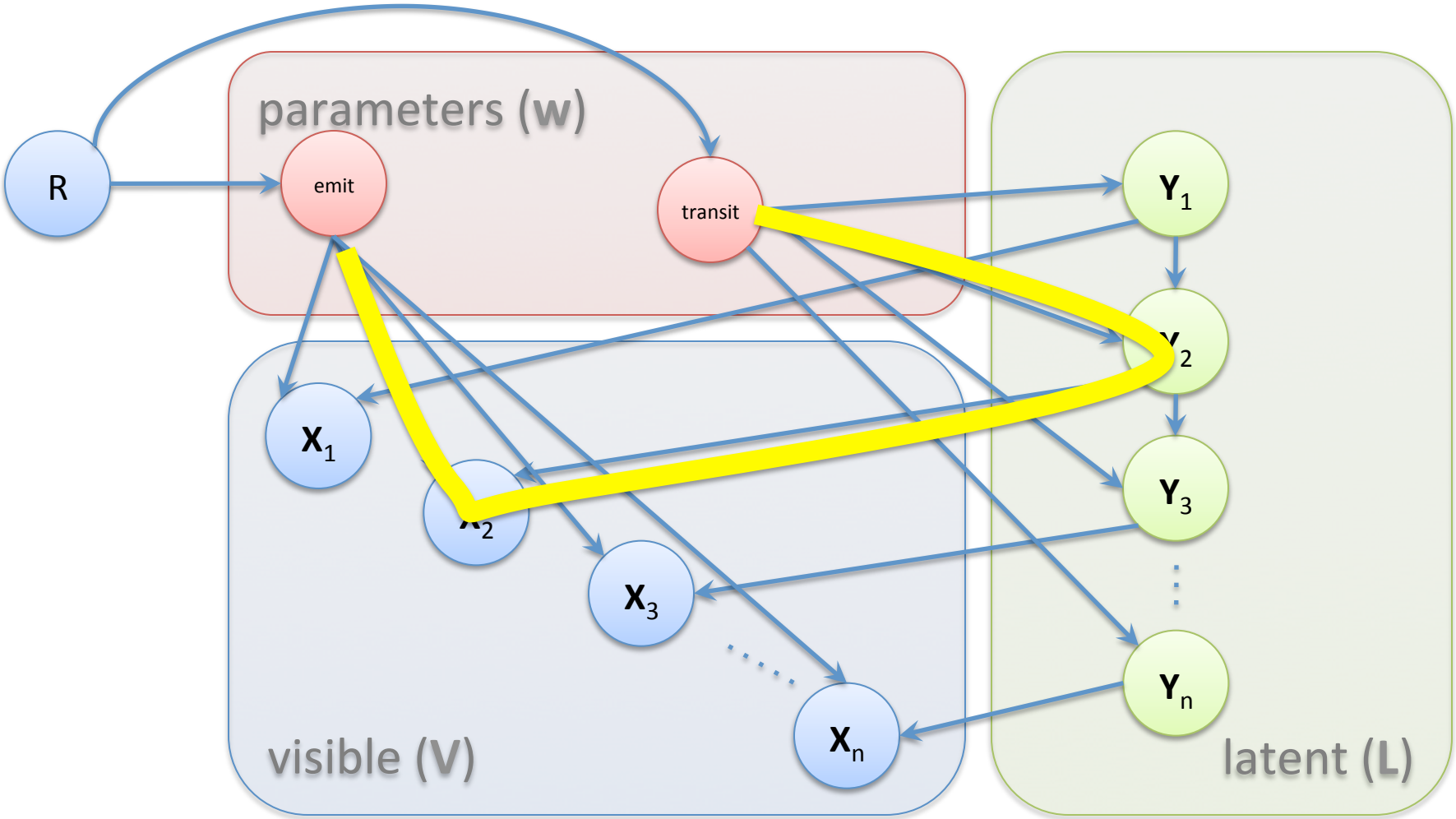
Baum-Welch (EM for HMMs) as an Example

- E step: forward-backward algorithm (on each example).
  - This is exact marginal inference by variable elimination.
  - The structure of the graphical model lets us do this by dynamic programming.
  - The marginals are probabilities of transition and emission events at each position.
- M step: MLE based on soft event counts.
  - Relative frequency estimation accomplishes MLE for multinomials.

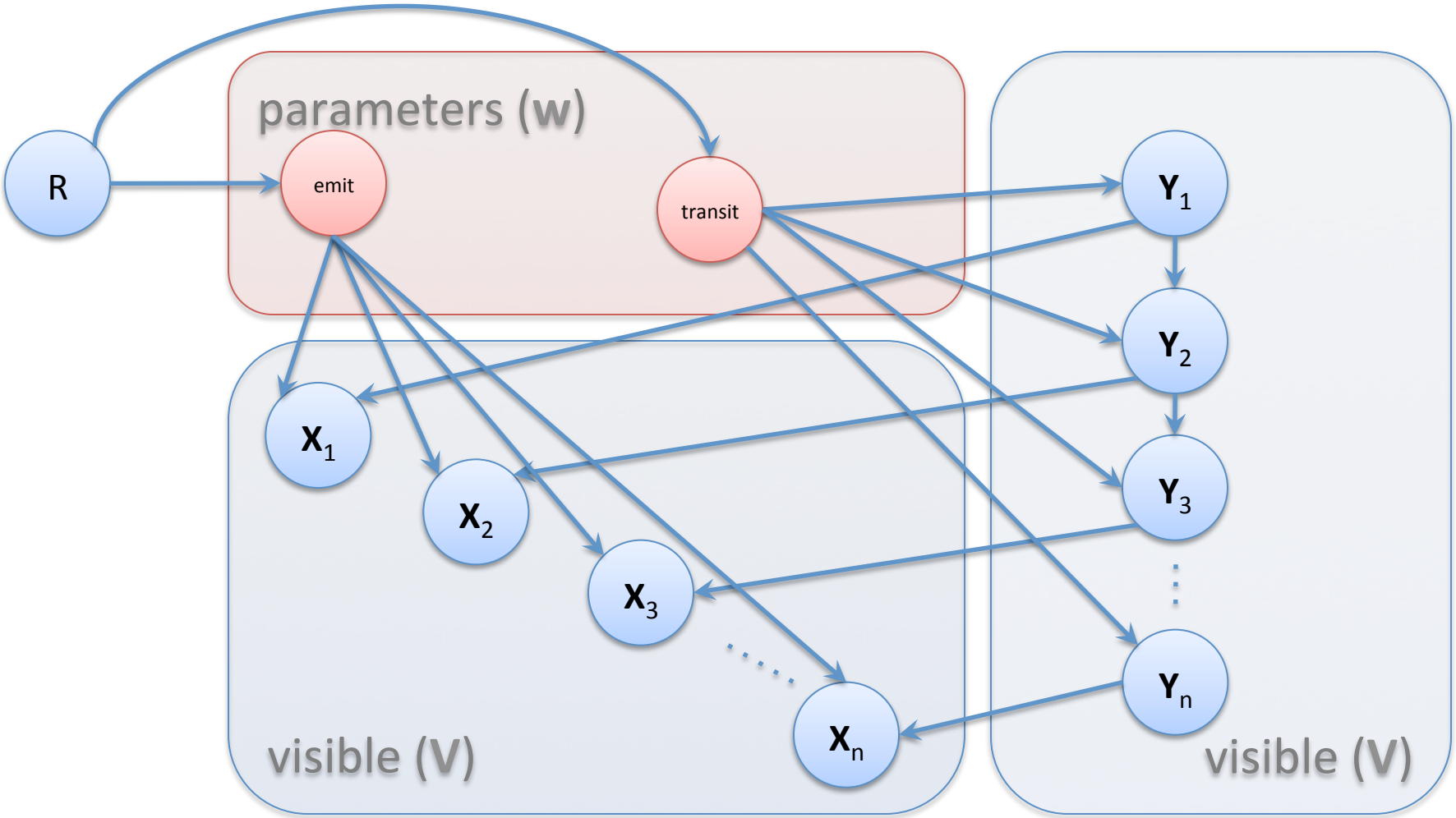
Baum-Welch as a Graphical Model



Active Trail!



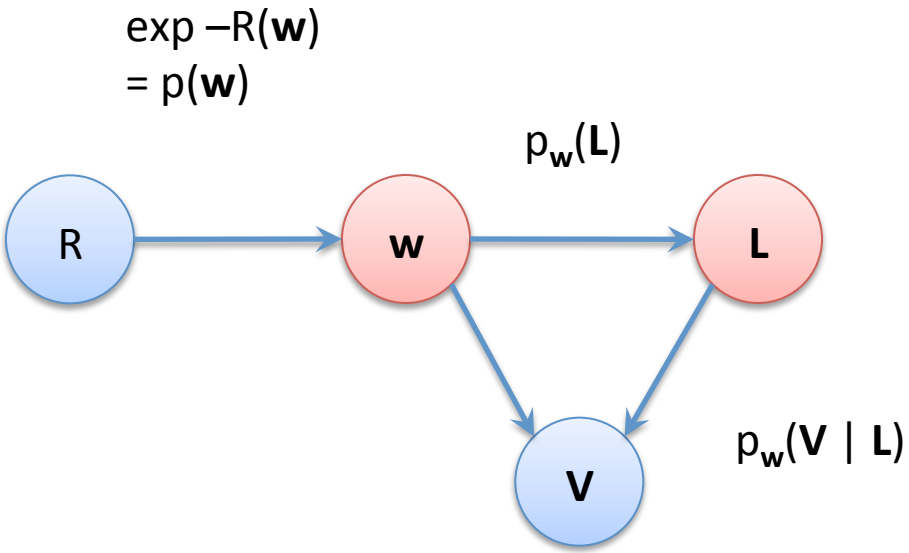
No Active Trail in All-Visible Case



Why Latent Variables Make Learning Hard

- New intuition: parameters are now interdependent that were not interdependent in the fully-visible case.
- It all goes back to the structural properties of the graphical model (in this case, active trails).

"Viterbi" Learning

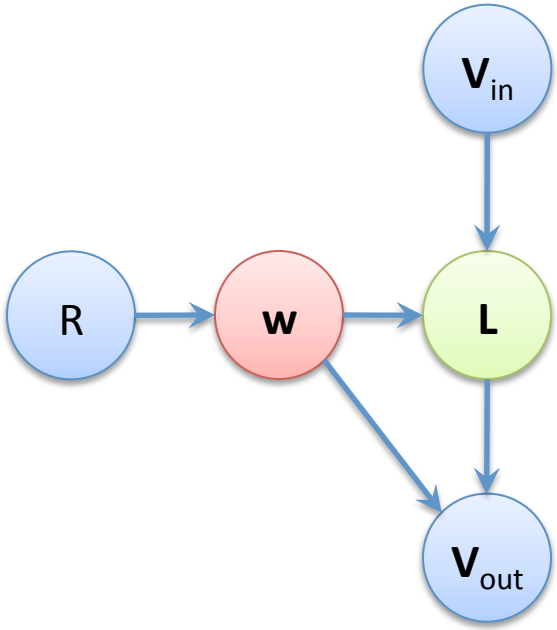


- Approximate joint MAP inference over **w** and **L** (most probable explanation inference).
- Loss function: $loss(\boldsymbol{v}; h_{\mathbf{w}}) \quad = \quad -\max_{\boldsymbol{\ell}} \log p_{\mathbf{w}}(\boldsymbol{v}, \boldsymbol{\ell})$

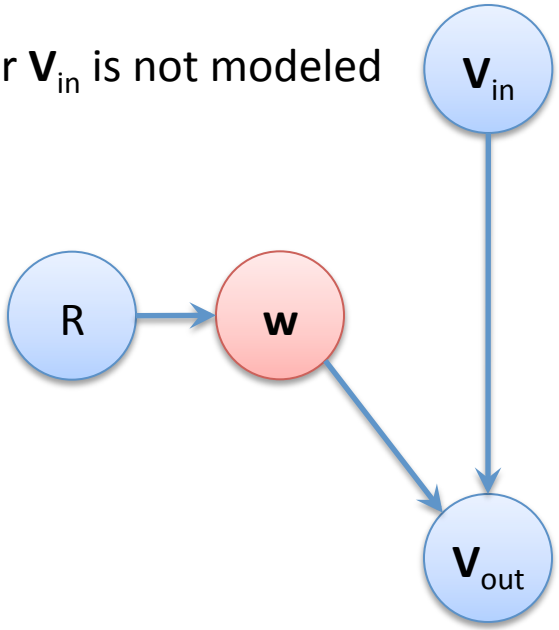
Conditional Models

- EM is usually closely associated with fully generative approaches.
- It generalizes to log-linear models and with conditional models.
  - Locally normalized models give flexibility without requiring global inference (Eisner, 2002).
  - Hidden variable CRFs (Quattoni et al., 2007) are very powerful.

Learning Conditional Hidden Variable Models



distribution over $\mathbf{V}_{in}$ is not modeled



standard conditional model (e.g., CRF)

Optimization for Hidden Variables

- We've described hidden variable learning as *inference* problems.
- It is more practical to think about this as *optimization*.
- EM can be understood from an optimization framework, as well.

EM and Likelihood

$$\Phi(\mathbf{w}) \quad = \quad \sum_i \log \sum_{\boldsymbol{\ell}} p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell})$$

- The connection between the goal above and the EM procedure is not immediately clear.

Optimization View of EM

$$\sum_i \left(-\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log q_i(\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\boldsymbol{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \log p_{\mathbf{w}}(\boldsymbol{v}_i) \right)$$

- A function of **w** and the collection of q_i.
- Claim: EM performs *coordinate ascent* on this function.

Optimization View of EM

$$\sum_i \left(-\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log q_i(\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\boldsymbol{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \log p_{\mathbf{w}}(\boldsymbol{v}_i) \right)$$

$\Phi(\boldsymbol{w})$

- The third term is our actual goal, Φ. It only depends on **w** (not the q_i).

Optimization View of EM

$$\Phi(\boldsymbol{w})$$

$$\sum_i \left(-\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log q_i(\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\boldsymbol{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \log p_{\mathbf{w}}(\boldsymbol{v}_i) \right)$$

$$\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell})$$

- The latter two terms together are precisely what we maximize on the M step, given the current q_i.
  - This is a concave problem and we solve it exactly.

Optimization View of EM

$$\sum_i \left(-\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log q_i(\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\boldsymbol{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \log p_{\mathbf{w}}(\boldsymbol{v}_i) \right)$$

$\Phi(\boldsymbol{w})$

$$\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell})$$

- Concern: is the M step improving term 2 at the expense of Φ (term 3)?
  - No.

The M Step

$$\Phi(\mathbf{w}) = \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell}) - \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i)$$

- Last term is also not getting any worse from iteration to iteration:

$$\begin{aligned}
&- \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log p_{\mathbf{w}^{(t+1)}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log p_{\mathbf{w}^{(t)}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) \\
&= - \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log p_{\mathbf{w}^{(t+1)}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \sum_i \sum_{\boldsymbol{\ell}} q_i^{(t)}(\boldsymbol{\ell}) \log q_i^{(t)}(\boldsymbol{\ell}) \\
&= \sum_i D(q_i^{(t)}(\cdot) \| p_{\mathbf{w}^{(t+1)}}(\cdot \mid \boldsymbol{v}_i)) \\
&\geq 0
\end{aligned}$$

The M Step

- Each M step, once q_i is fixed, maximizes a bound on the log-likelihood Φ.
  - For fixed q_i, this is a concave problem we can solve in closed form in many cases.
- What about the E step?

Optimization View of EM

$$\sum_i \left(-D(q_i(\cdot) \| p_{\mathbf{w}}(\cdot \mid \boldsymbol{v}_i)) + \Phi(\boldsymbol{w}) \right)$$

$$-\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log q_i(\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\boldsymbol{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \log p_{\mathbf{w}}(\boldsymbol{v}_i)$$

$$\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell})$$

- E step considers the first two terms.
- Sets each q_i to be equal to the posterior under the current model.

Coordinate Ascent

$$-D(q_i(\cdot)\|p_{\mathbf{w}}(\cdot \mid \boldsymbol{v}_i))$$

$$\sum_i \left(-\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log q_i(\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\boldsymbol{w}}(\boldsymbol{\ell} \mid \boldsymbol{v}_i) + \log p_{\mathbf{w}}(\boldsymbol{v}_i) \right)$$

$$\sum_{\boldsymbol{\ell}} q_i(\boldsymbol{\ell}) \log p_{\mathbf{w}}(\boldsymbol{v}_i, \boldsymbol{\ell})$$

- E step fixes **w** and solves for the q_i.
- M step fixes all q_i and solves for **w**.

Things People Forget About EM

- Multiple random starts (or non-random starts), select final model using likelihood on development data.
- Variants may help avoid local optima …

Variants of EM

- "Online" variants where we do an E step on one or a mini-batch of examples are still coordinate ascent (Neal and Hinton, 1998).
- Deterministic annealing: flatten out the q_i, making the function closer to concave.
- Stochastic variant: use randomized approximate inference for E step.
- "Generalized" EM: improve **w** but don't bother optimizing completely.

Direct Optimization

- An alternative to EM: apply stochastic gradient ascent or quasi-Newton methods directly to Φ.
- Typically done for MN-like models with features, e.g., latent-variable CRFs.
  - Gradient is a difference of feature expectations.
  - Requires marginal inference.

Summary

- EM: many ways to understand it.
  - The guarantee: each round will improve the likelihood.
  - That's about as much as we can say.
- Sometimes it works.
  - Smart initializers
  - Lots of bias inherent in the model structure/ assumptions
- Better to understand the more general problem of optimizing a model given data.

Bayesian Modeling

In Statistics …

- Better to assign F. and B. to analyses, not people.

- **Frequentist** analysis (most of science today): parameters are fixed and unknown; we gain information by repeated experiments
  - Point estimates, standard errors, confidence intervals ("in P% of experiments, the interval will cover the true θ"), hypothesis tests with α fixed in advance, reason about p(data | H_0)
- **Bayesian** analysis: treat unknown parameters probabilistically; update beliefs as evidence arrives
  - Start with p(θ) and infer p(θ | data), means and quantiles of the posterior over θ, intervals corresponding to "P% belief" that θ is in the interval

The Attraction of Bayesian Thinking

- Write down your model declaratively, worry later about how to fit it from data.
- Prior encodes *prior* knowledge.
  - We have lots of this when it comes to language, or at least we think we do!
- Manage uncertainty about the model the same way we manage uncertainty about the data.
- Bayesian methods are strongly associated with:
  - *unsupervised* (and latent variable) learning
  - generative models

Evolving Definitions

- MLE (not Bayesian): $\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\text{data})$
- Maximum *a posteriori* estimation:

$$\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\text{data}) p_{\boldsymbol{\alpha}}(\boldsymbol{\theta})$$

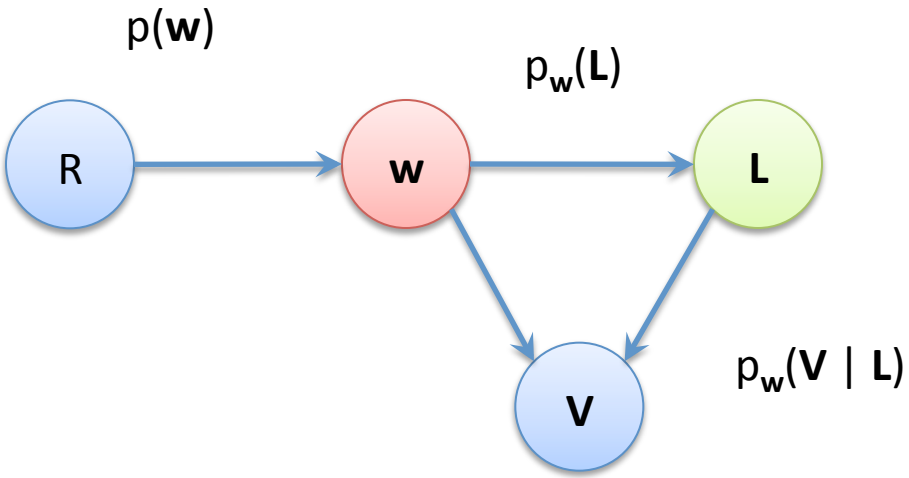
- Computing the posterior over the parameters (fully Bayesian):

$$p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}, \text{data}) = \frac{p_{\boldsymbol{\theta}}(\text{data}) p_{\boldsymbol{\alpha}}(\boldsymbol{\theta})}{\int p_{\boldsymbol{\theta}'}(\text{data}) p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}') d\boldsymbol{\theta}'}$$

- Empirical Bayesian:

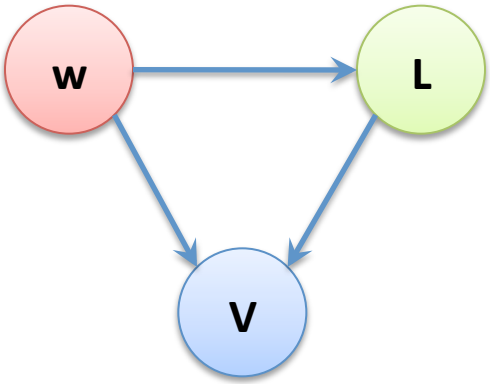
$$\max_{\boldsymbol{\alpha}} \int p_{\boldsymbol{\theta}}(\text{data}) p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

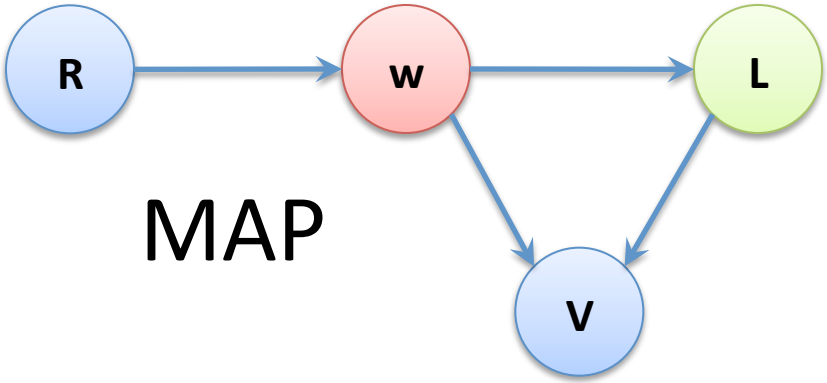
MAP Learning as a Graphical Model



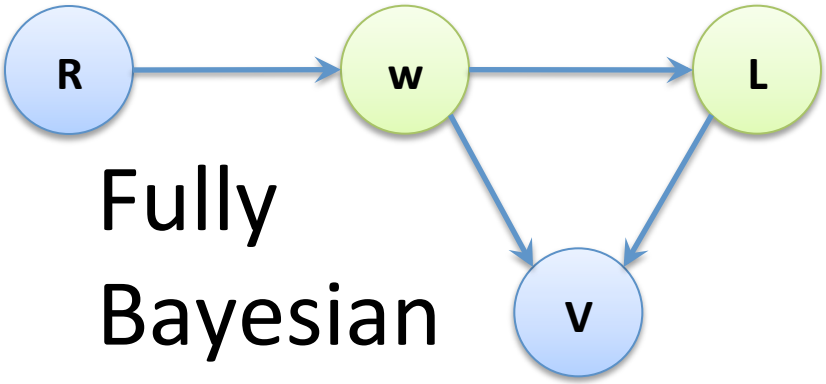
- Combined inference (max over **w**, sum over **L**) is very hard.
  - If **w** were fixed, getting the posterior over **L** wouldn't be so bad.
  - If **L** were fixed, maximizing over **w** wouldn't be so bad.
- "Standard EM" doesn't have p(**w**); it's very simple to add and useful in practice.

MLE

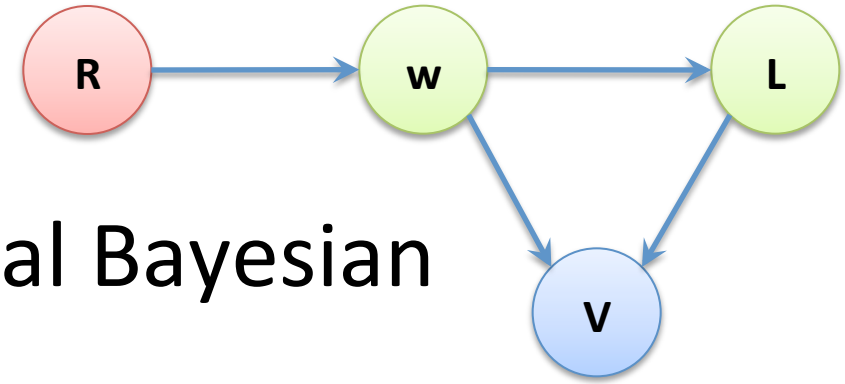




MAP



Fully Bayesian



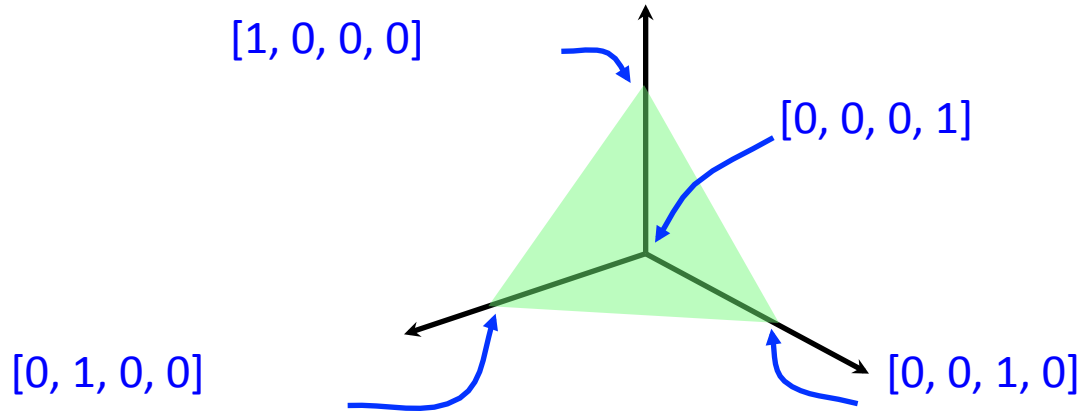
Empirical Bayesian

Multinomials

- Let's assume discrete distributions that simply assign probabilities to finite sets of events.
  - n-gram models, HMMs, PCFGs, …

Distributions over Multinomials

- You can think of a multinomial distribution over d events as a point in the (d-1) simplex.



- To randomly pick a point in this space, we need a **continuous** distribution over the simplex.

Dirichlet Distribution

- A distribution over the d-event probability simplex.
- Parameters: **ρ**, the mean of the Dirichlet, and α, the concentration around that mean (large α means smaller variance).
- Beta function:
- Gamma function (generalized factorial):

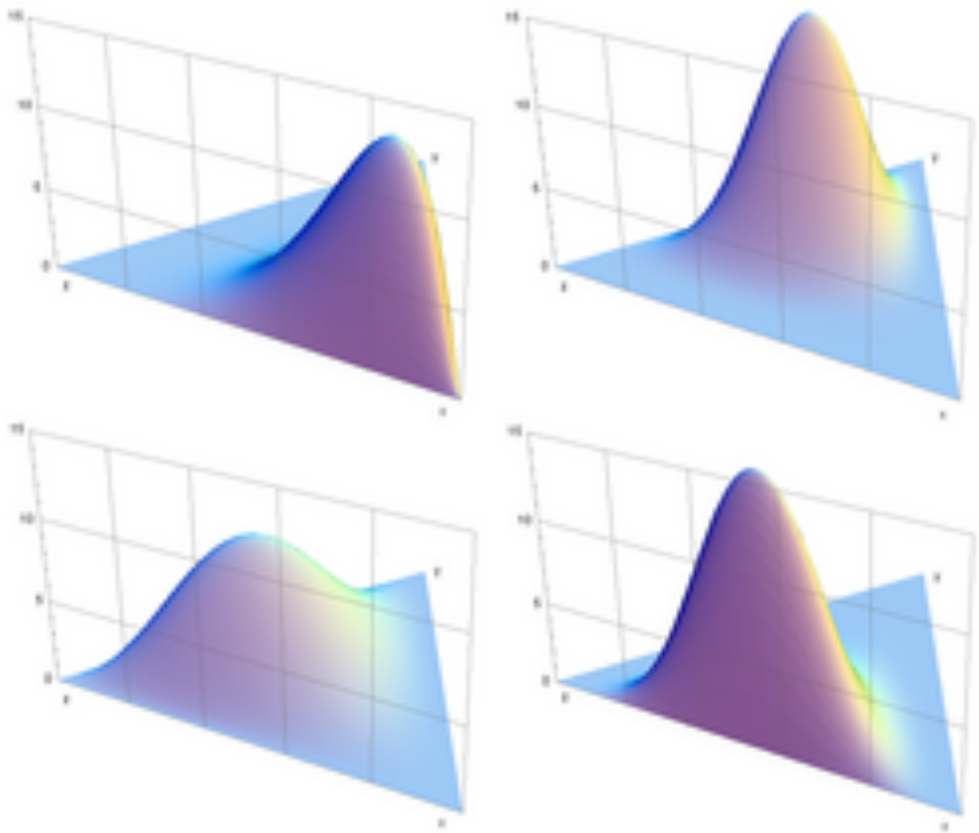
$$p(\boldsymbol{\theta} \mid \alpha, \boldsymbol{\rho}) = \frac{1}{\mathrm{B}(\alpha\boldsymbol{\rho})} \prod_{i=1}^{d} \theta_i^{\alpha\rho_i - 1}$$

$$\mathrm{B}(\alpha\boldsymbol{\rho}) = \frac{\prod_{i=1}^{d} \Gamma(\alpha\rho_i)}{\Gamma(\alpha)}$$

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-a}\, dt$$

Dirichlet, d=3
(various parameter settings)

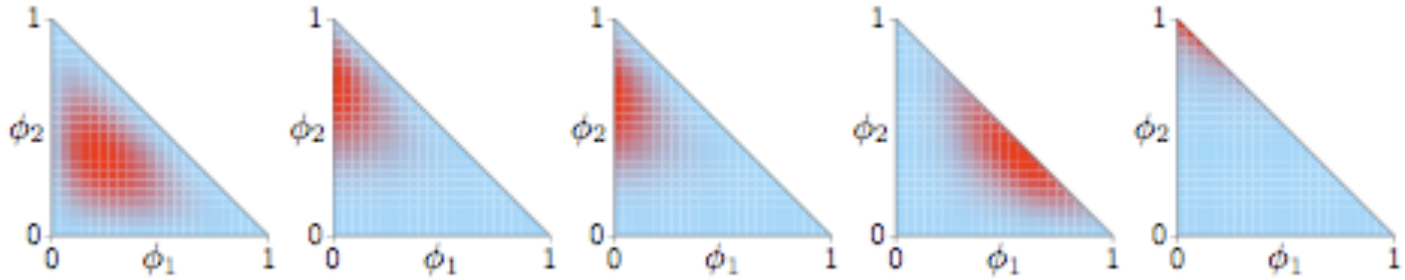
from answers.com



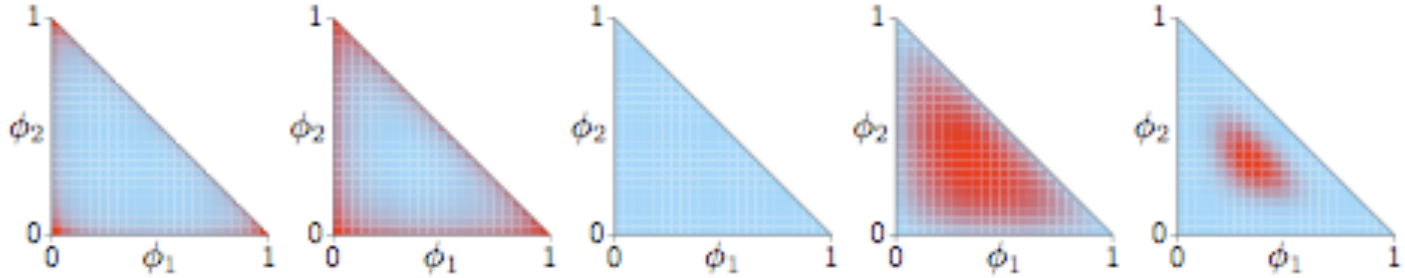
Dirichlet, d= 3
(different “means” and “variances”)

- from Liang and Klein, 2007

Different means:



Different variances:



Sampling from a Dirichlet

- For i from 1 to do, sample v_i from a gamma distribution with shape $\alpha\rho_i$ and scale 1.
- Renormalize the vector $\mathbf{v}$ to obtain $\boldsymbol{\theta}$.

MAP with a Dirichlet

- Recall that we can use a prior to "smooth" an estimate.
- For a multinomial $\boldsymbol{\theta}$ with Dirichlet prior $\alpha\boldsymbol{\rho} > \mathbf{1}$, this equates to adding *pseudocounts* to the vector of observed counts.

$$\hat{\theta}_i = \frac{N_i + \alpha\rho_i - 1}{N + \alpha - d}$$

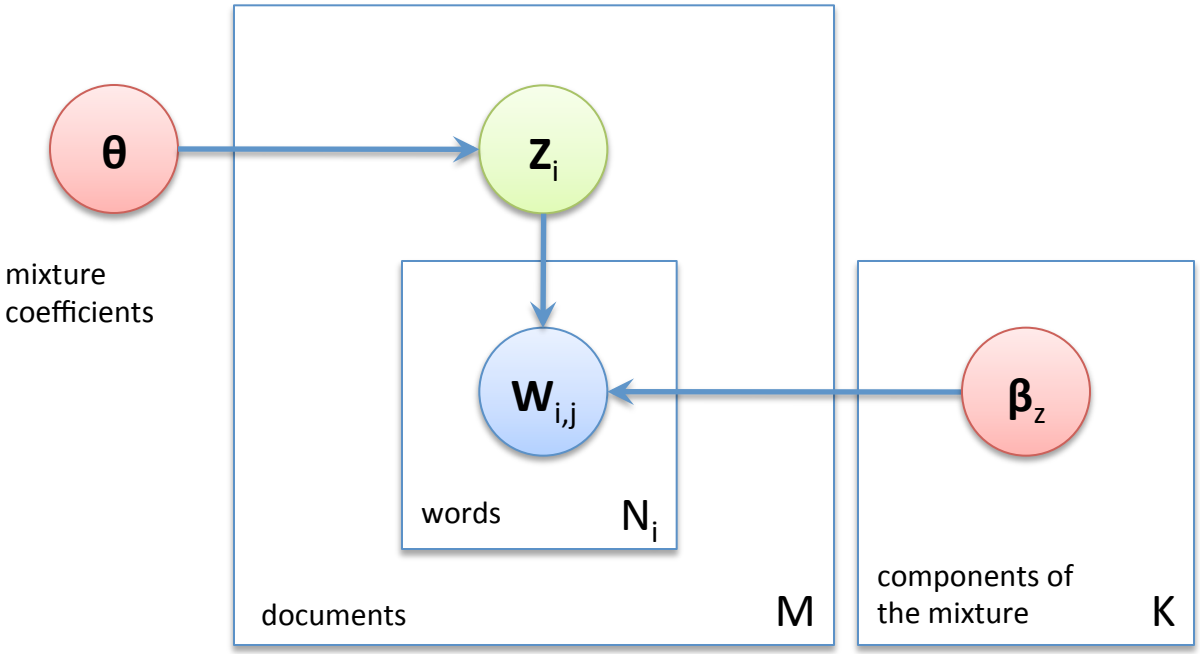
  - As counts become large, prior matters less.
  - Closed form!
  - Regularizer view: $R(\boldsymbol{\theta}) = -\sum_{i=1}^{d} (\alpha\rho_i - 1) \log \theta_i$

- Flat prior: $\alpha = d$, all $\rho_i = 1/d$ (equates to MLE)
- Sparse prior (encourages most θ_i to go to zero), but now it's not closed form.

Mixture of Unigrams

- The generative story for a classical document-clustering model would be something like this (Nigam et al., 2000):
- For i = 1 ... M (number of documents):
  - Draw a document length N_i from some distribution.
  - Draw a topic z_i for the document from a multinomial over topics, $\boldsymbol{\theta}$.
  - For j = 1 ... N_i:
    - Draw word w_{ij} from the multinomial $\boldsymbol{\beta}_z$.
- Nigam et al. learned this using EM.

Mixture of Unigrams

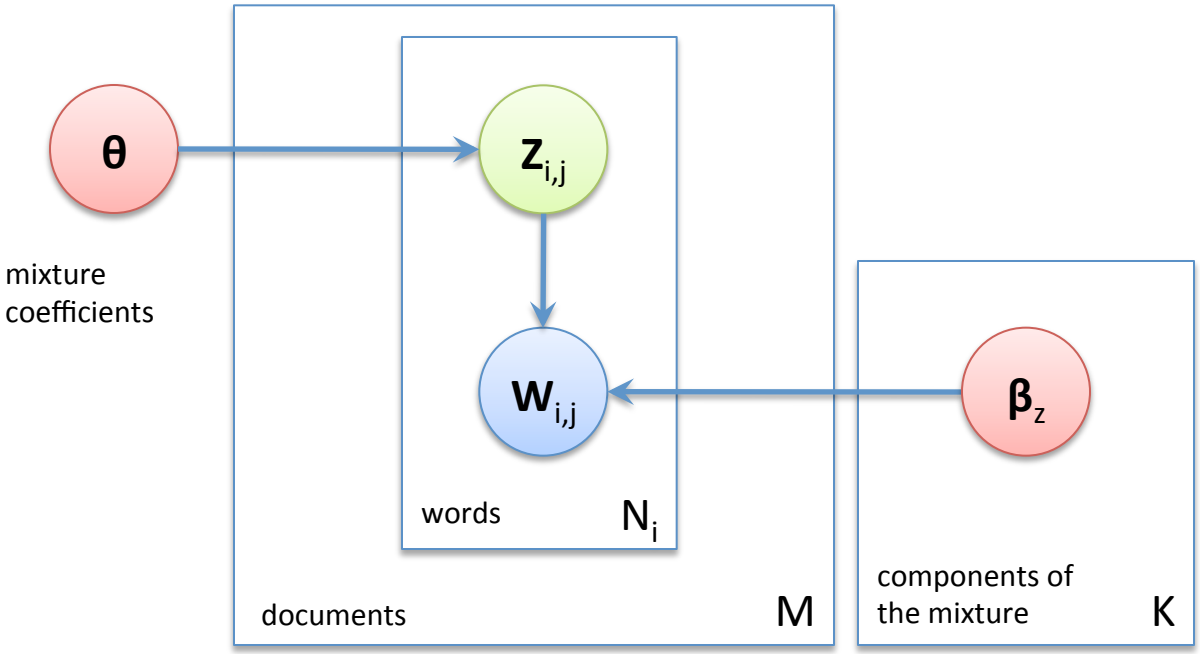


Z_i is sometimes called the **topic** of document i.

A topic z is defined by a unigram distribution $\boldsymbol{\beta}_z$.

M = number of documents
N_i = number of words in document i
K = number of mixture components

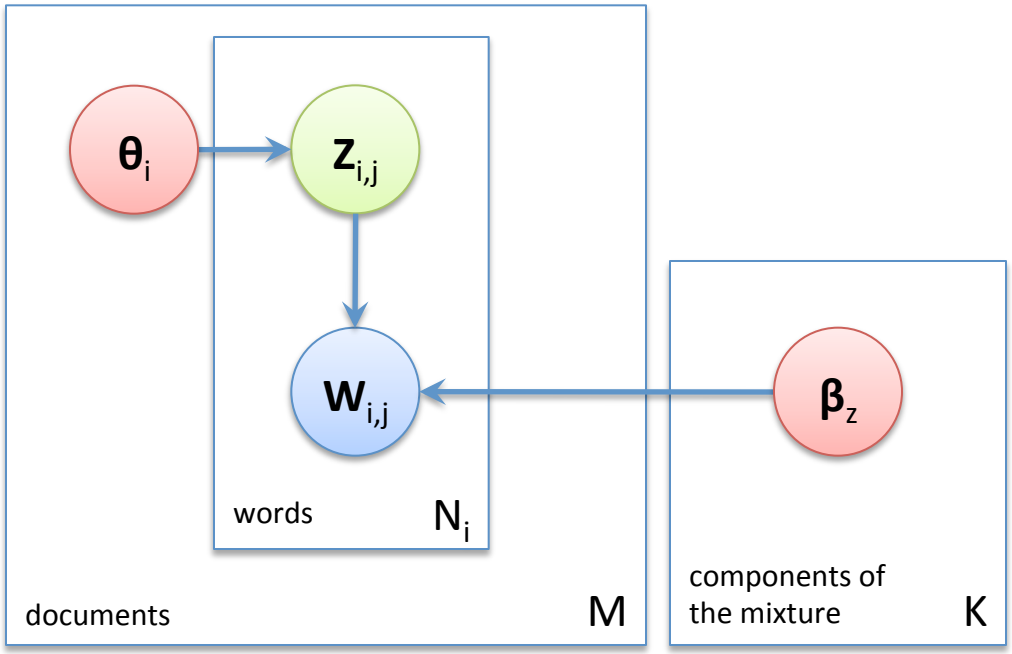
A *Word* Clustering Model



Problem: all words are the same; document information is irrelevant. (This is exactly a zero-order HMM.)

M = number of documents
N_i = number of words in document i
K = number of mixture components

Probabilistic LSI (Hofmann, 1996)



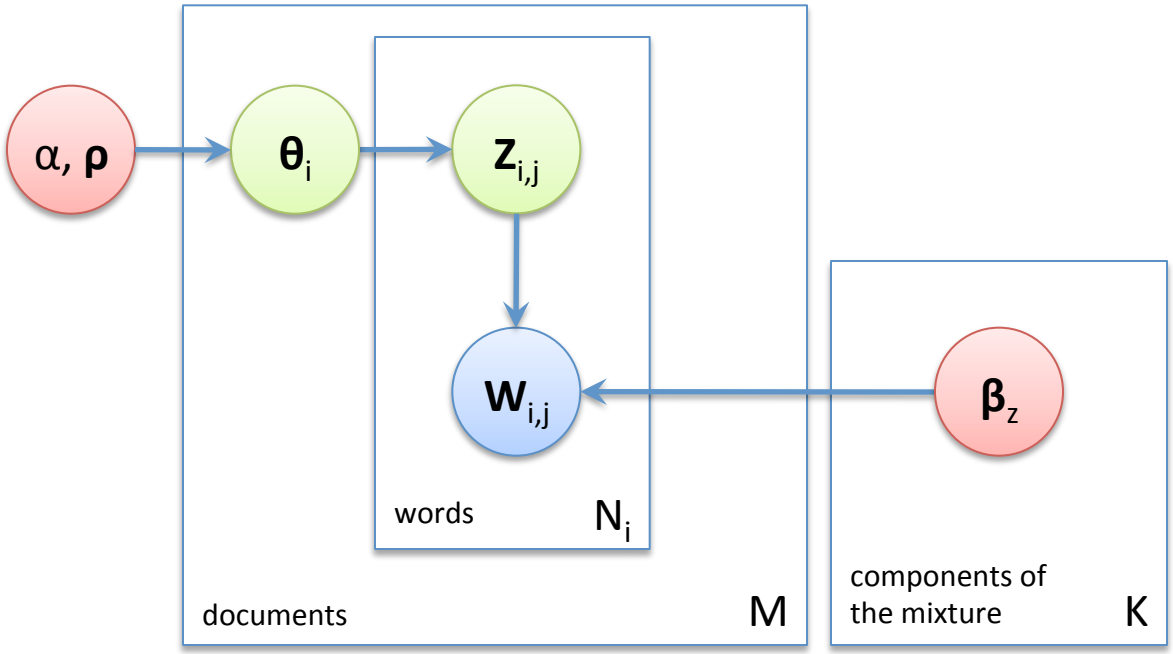
This has very little to do with **latent semantic indexing**, except that it's a probabilistic model trying to perform a similar task.

Word clustering; documents correspond to *distributions* over topics.

Problem: can't describe new documents!

M = number of documents
N_i = number of words in document i
K = number of mixture components

Latent Dirichlet Allocation (Blei et al., 2003)



Documents are mixtures of topics, but a prior over those mixtures lets us reason about new documents, too.

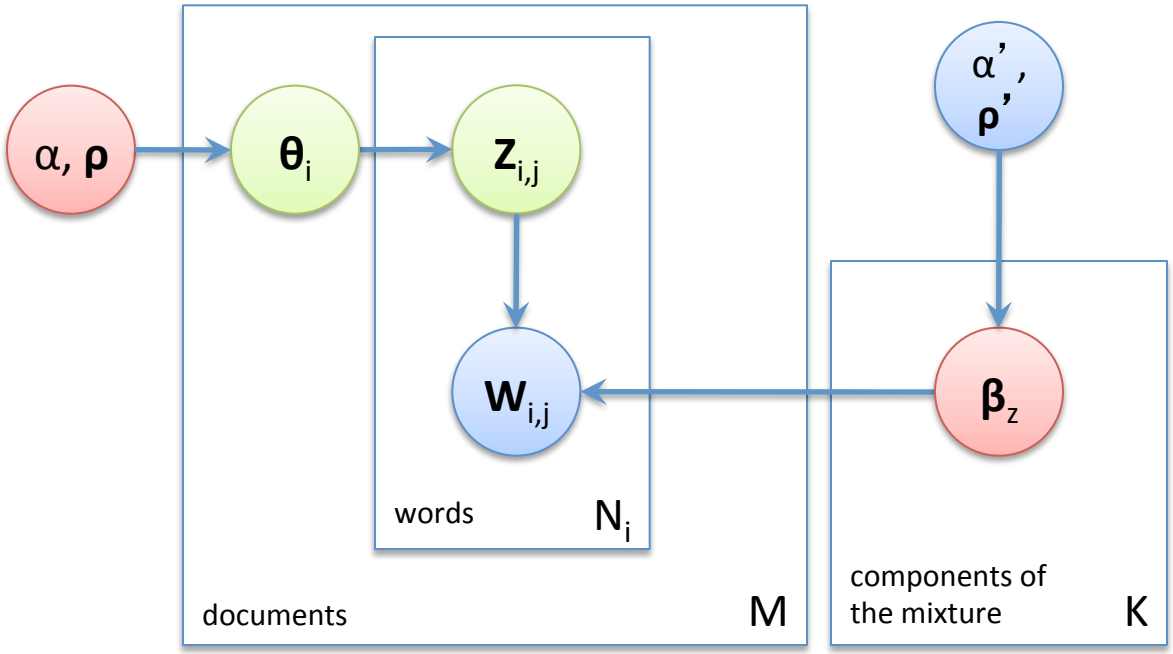
M = number of documents
N_i = number of words in document i
K = number of mixture components

LDA on ACL Papers (Gimpel, 2006)

"POS tagging"	"Information Retrieval"	"Parsing"	"MT"	"Speech Recognition"	"Probabilistic Modeling"	"Experiments"	"Syntax"
pos	document	parsing	translation	speech	model	corpus	verb
tags	terms	parser	alignment	recognition	models	results	verbs
tagging	query	parse	word	spoken	probability	data	noun
sequence	term	treebank	english	language	training	number	case
tag	documents	accuracy	source	asr	data	table	syntactic
information	retrieval	parses	target	error	word	frequency	phrase
chunk	information	penn	translations	errors	language	test	clause
label	web	trees	machine	speaker	probabilities	average	structure
hmm	text	empty	phrase	utterances	set	found	phrases
learning	search	section	words	results	words	values	nouns
sequences	queries	wsj	language	turns	distribution	total	english
labels	system	proceedings	bilingual	rate	statistical	cases	subject
crf	collections	results	parallel	table	parameters	distribution	lexical

Figure 9: Selected topics resulting from executing the Gibbs sampler for inference in the LDA model for the collection of ACL papers from 1999 up to 2006. The top 13 words for each topic are shown, in decreasing order of their probability. The topics have been given titles by hand based on their most probable words.

Smoothed Latent Dirichlet Allocation (Blei et al., 2003)



M = number of documents
N_i = number of words in document i
K = number of mixture components

Topic Models Beyond LDA

- Small industry in variations on topic models, usually adding more evidence to be explained by the topics.
- Examples:
  - Supervised LDA (Blei and McAuliffe, 2007) adds an observed document category.
  - Link LDA (Erosheva et al., 2004) adds citations to the document, explained by more draws from $\boldsymbol{\theta}_i$
  - Author topic model (Rosen-Zvi et al., 2004) adds authors.
  - Correlated topic model (Blei and Lafferty, 2006) lets different topics correlate more flexibly through a different prior.
  - Comment LDA (Yano et al., 2009) generates comments from a different set of unigram models but the same topics.

Where is the Structure?

- Through the topics and θ_i, words in a document become interdependent.
  - Kind of a joint document/word clustering.

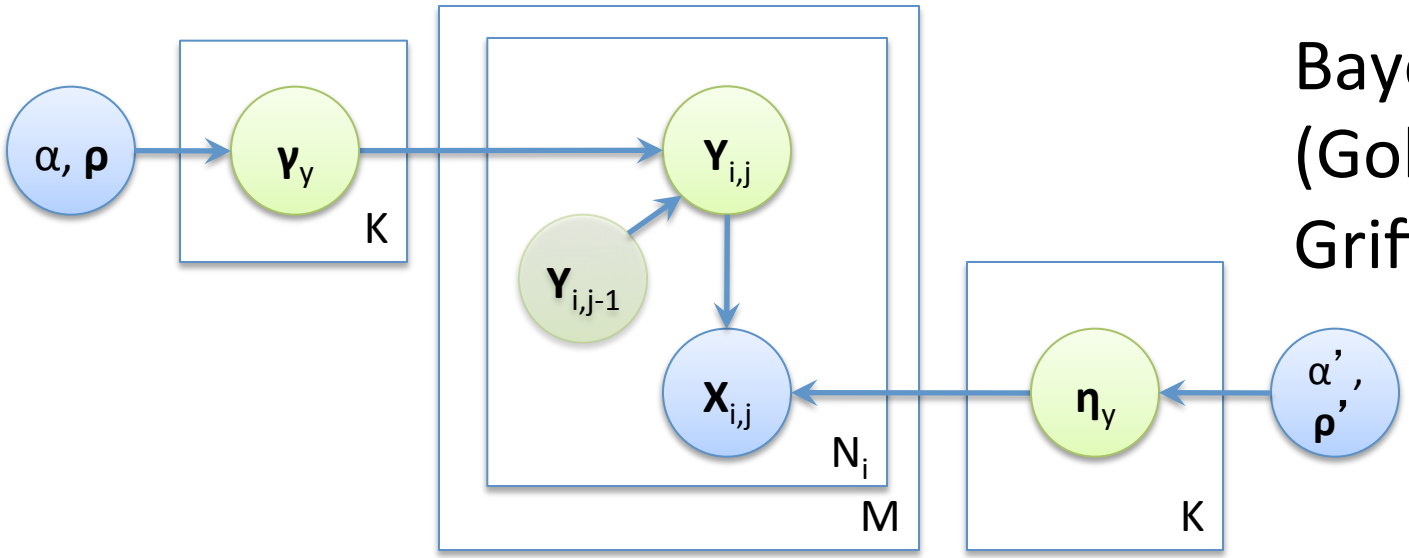
- Not really discrete structure the way we've mostly discussed in this class, though.

- LDA is a "Bayesian zero-order HMM."

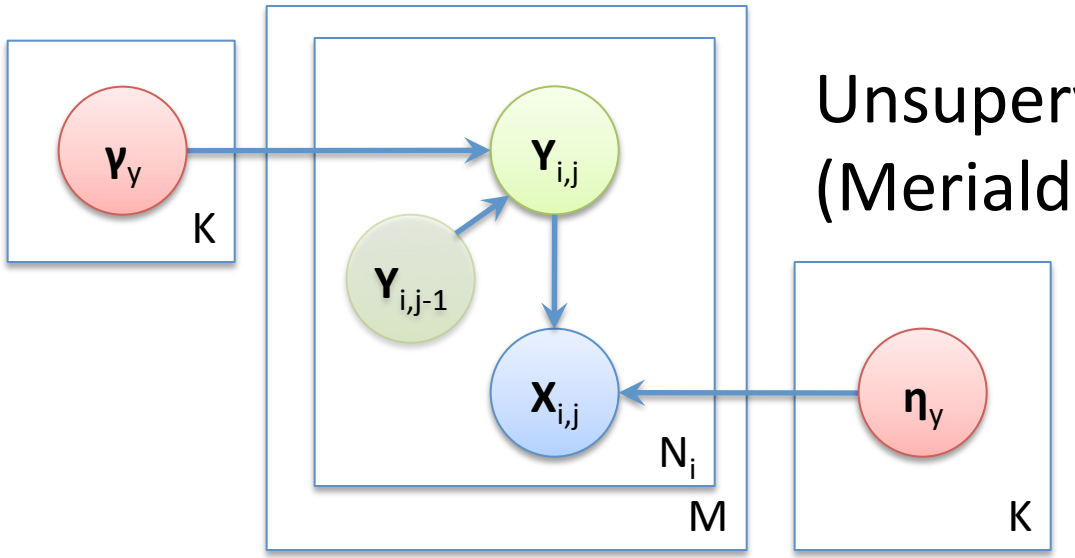
Where is the Prediction?

- Topics are hard to evaluate; no gold standard.
- This is either an open problem or the nail in the coffin, depending on your point of view.

Hidden Markov Models



Bayesian HMM (Goldwater and Griffiths, 2007)



Unsupervised HMM (Merialdo, 1994)

M = number of sequences
N_i = number of words in sequence i
K = number of states

The Engineering Part

- Typically approximate inference is required.
  - Markov chain Monte Carlo (e.g., Gibbs sampling)
  - Variational inference (e.g., mean-field)
- Graphical model view is *really* helpful when designing inference algorithms for your Bayesian model!
- Learning: approximate inference + optimization of hyperparameters (for LDA, usually α and $\boldsymbol{\beta}$; ρ is often assumed uniform).
  - Stochastic or variational EM, depending on your choice of approximate inference.
- Full Bayesian: fix the prior and do inference on all your data.
  - Implications for train/test methodology?

Sketch of Gibbs Sampling

- MCMC: design (on paper) a graph where each configuration from Val(**V**) is a node.
  - Transitions in the graph designed to give a Markov *chain* whose stationary distribution is the posterior.
- Simulate a random walk in the graph.
- If you walk long enough, your position is distributed according to P(**V**).

Transitions in Gibbs Sampling

- A transition in the Markov chain equates to changing a subset of the random variables.
- Gibbs: resample V_i's value according to $P(V_i \mid \mathbf{V} \setminus \{V_i\})$.
  - Only need the local factors that affect V_i: take product, marginalize, and randomly choose new value.

- Simply lock evidence variables **X**.
- Maximizing version gradually shifts sampler in favor of most probable value for V_i.

Sketch of Mean Field Variational Inference

- Inference with our distribution P is hard.
- Choose an "easier" distribution family, $\mathcal{Q}$. Then find:

$$\arg\min_{Q \in \mathcal{Q}} D(Q \| P)$$

- Usually iterative methods are required to "fit" Q to P.
  - These often resemble familiar learning algorithms like EM!

Energy Functional

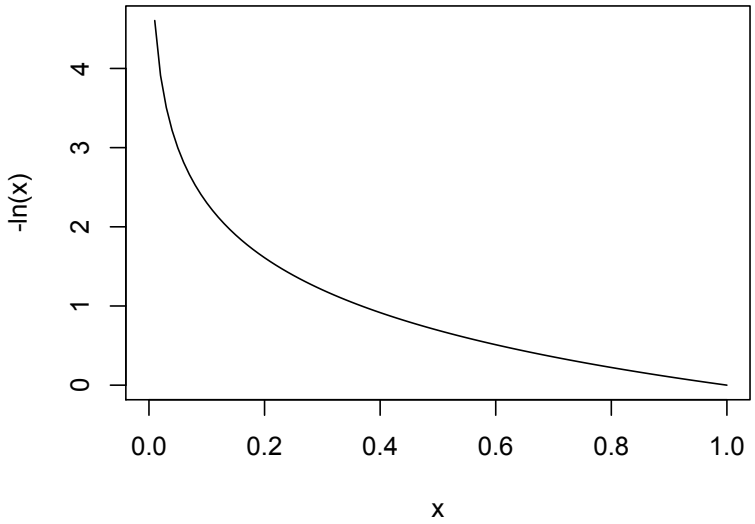
$$
\begin{aligned}
D(Q(Y)\|P(Y \mid X=x)) &= \mathbb{E}_Q[\log Q(Y)] - \mathbb{E}_Q[\log P(Y \mid X=x)] \\
&= -H(Q(Y)) - (\mathbb{E}_Q[\log P(X=x,Y)] - \log P(X=x)) \\
&= -H(Q(Y)) - \left(\sum_\phi \mathbb{E}_Q[\log \phi_{|x}] - \log P(X=x)\right) \\
\underbrace{\log P(X=x)}_{\text{constant}} &= D(Q(Y)\|P(Y \mid X=x)) + \underbrace{H(Q(Y)) + \sum_\phi \mathbb{E}_Q[\log \phi_{|x}]}_{\text{maximize this}}
\end{aligned}
$$

- Expectations under simpler distribution family, $\mathcal{Q}$.
  - Every element of $\mathcal{Q}$ is an approximate solution.
  - We try to find the best one.

Variational Methods

- This is a simple example.
- For any λ and any x:

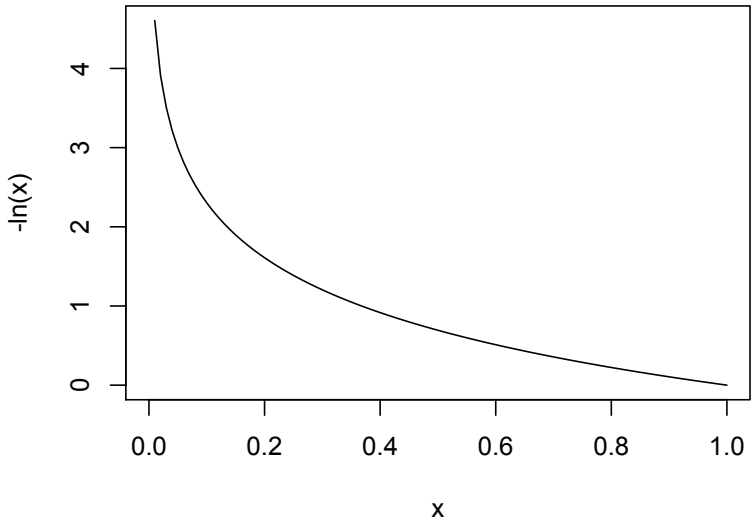
$$-\ln(x) \quad \geq \quad \underbrace{-\lambda x + \ln(\lambda) + 1}_{\text{family of functions } g_\lambda(x)}$$



Variational Methods

- This is a simple example.
- For any λ and any x:

$$-\ln(x) \quad \geq \quad -\lambda x + \ln(\lambda) + 1$$

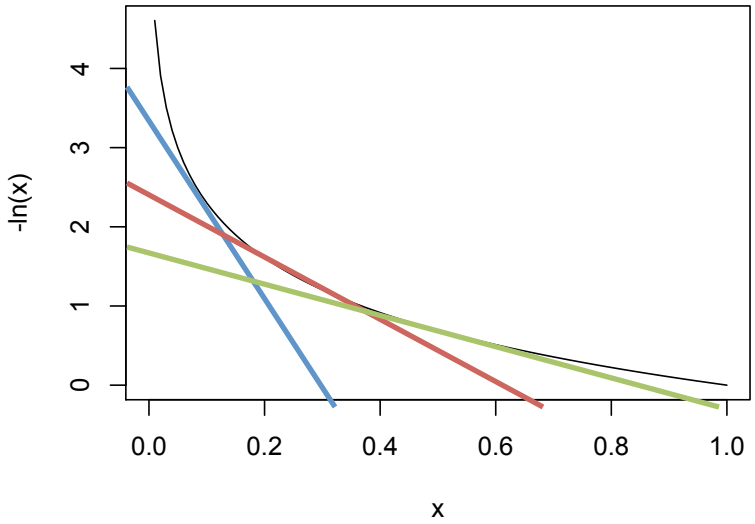


- Further, for any x, there is some λ where the bound is tight.
  - λ is called a **variational parameter**.

Tangent: Variational Methods

- This is a simple example.
- For any λ and any x:

$$-\ln(x) \quad \geq \quad -\lambda x + \ln(\lambda) + 1$$

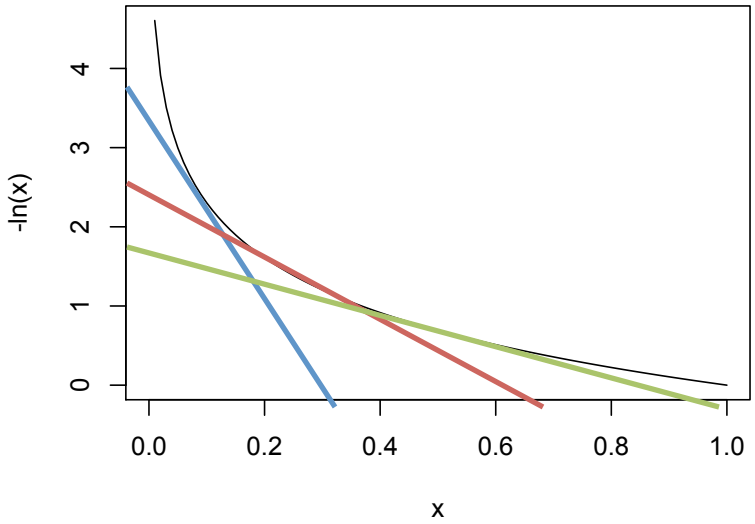


- Further, for any x, there is some λ where the bound is tight.
  - λ is called a **variational parameter**.

Tangent: Variational Methods

- This is a simple example.
- For any λ and any x:

$$-\ln(x) \quad \geq \quad -\lambda x + \ln(\lambda) + 1$$



- Further, for any x, there is some λ where the bound is tight.
  - λ is called a **variational parameter**.
- For us, log P(X = x) is like -ln(x), and Q is like λ.

Structured Variational Approach

- Maximize the energy functional over a family $\mathcal{Q}$ that is well-defined.
  - A graphical model!
  - Probably not an I-map for P. (Bound isn't tight.)
- Simpler structures lead to easier inference.
  - Mean field is the simplest:

$$Q(\boldsymbol{V}) \quad = \quad \prod_i Q_i(V_i)$$

Going Nonparametric

- How many topics or states?
- Nonparametric: let the data decide.
  - Not necessarily Bayesian or even probabilistic!
  - More data justify more parameters.
- Most common nonparametric and Bayesian tools in NLP are based on the Dirichlet process.
  - DP is not the same as the Dirichlet *distribution*.
- You can be Bayesian without being nonparametric, and you can be nonparametric without being Bayesian!

Remember

- “Bayesian” describes your model, not you!
- Approximate inference is necessary in Bayesian modeling, but useful elsewhere, too!