

Lecture 6: The Bayesian Approach

What Did We Do Up to Now?

- We are given a model
 - Log-linear model, Markov network, Bayesian network, etc.
- This model induces a distribution $P(X)$
- Learning: estimate a set of parameters

Bayesian Reasoning

- We have a model
 - This model encodes beliefs about the parameters as well
- We observe data
- We update our beliefs about the parameters according to the observed data

Bayesian vs. Non-Bayesian

- The Non-Bayesian model:

$$P(X \mid \theta)$$

- The Bayesian model:

$$P(X, \theta) = P(\theta)P(X \mid \theta)$$

Why Be Bayesian?

- This issue is the subject of a longstanding debate in statistics (Berger, 2010)
- Subjective probability vs. probability in a repeated experiment

Main Bayesian Procedure

- Learning stays within “probability theory:”

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{\int_{\theta} P(X \mid \theta)P(\theta)d\theta}$$

- In NLP, we would usually have an additional hidden structure:

$$P(\theta, Z \mid X) = \frac{P(X \mid Z, \theta)P(Z \mid \theta)P(\theta)}{\int_{\theta} \sum_Z P(X \mid Z, \theta)P(Z \mid \theta)P(\theta)d\theta}$$

e.g. tree e.g. sentence



Simple Example

- Coin toss: $X \sim \text{Bernoulli}(\theta)$

- Let's assume that

$$\theta \sim \text{Beta}(\alpha_H, \alpha_T)$$

$$P(\theta) \propto \theta^{\alpha_H - 1} (1 - \theta)^{\alpha_T - 1}$$



- What is the posterior?
- How is that linked to smoothing?

Coin Toss Example

Nice Properties

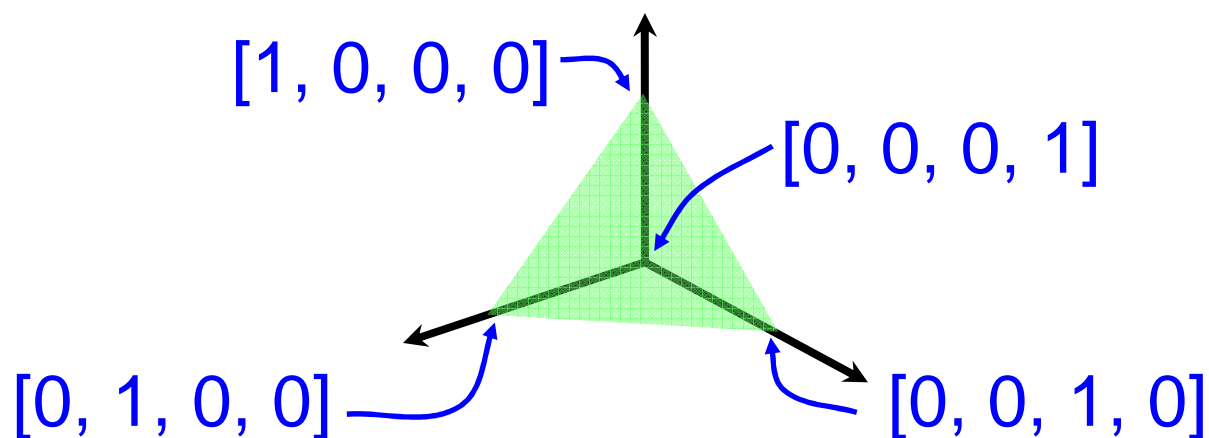
- Posterior over θ : Beta as well
 - Why? The prior is conjugate
- Smoothing: pseudo counts
- What happens when our data size grows?
- What else can we do with the posterior?
 - Another version of pseudo counts

General Multinomials

- The HMM model keeps track of two sets of multinomials:
 - Transition
 - Emission
- Bayesian networks' CPTs are also multinomials
- In the more general case, we need a prior over multinomials
- Multinomials are the building blocks of NLP models

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

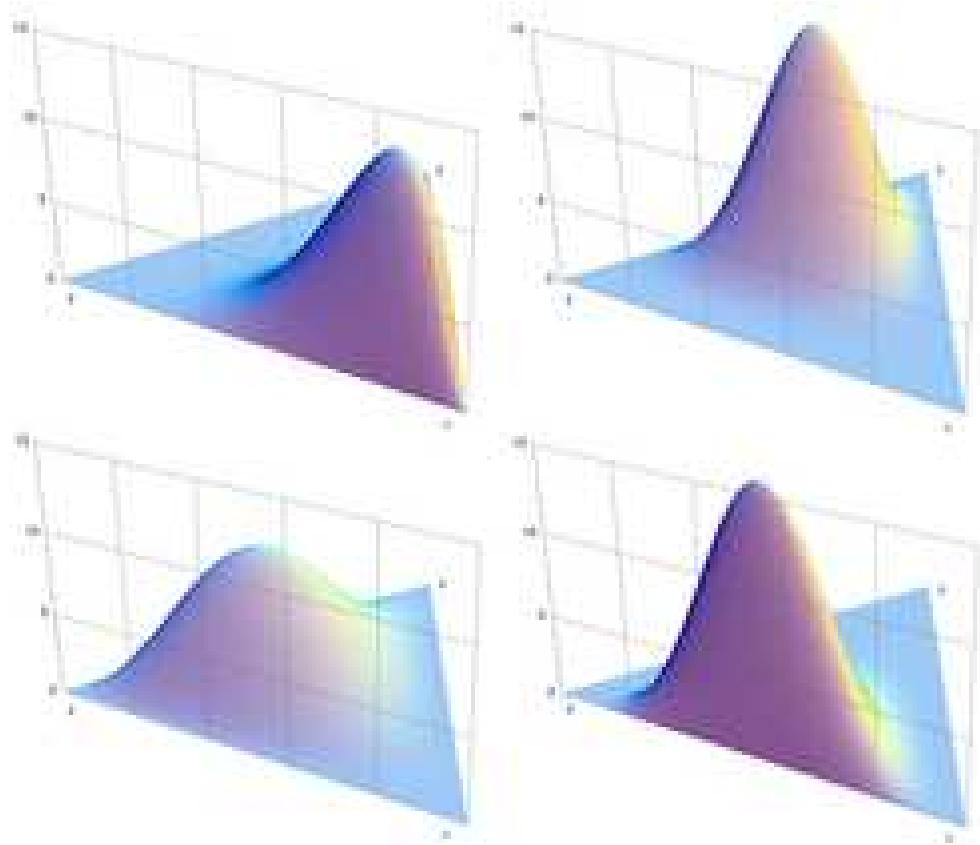
- A distribution over the d-event probability simplex.
- Parameters: α , a vector of positive values.
- Beta function:
- Gamma function (generalized factorial):

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

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^d \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^d \alpha_i\right)}$$

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

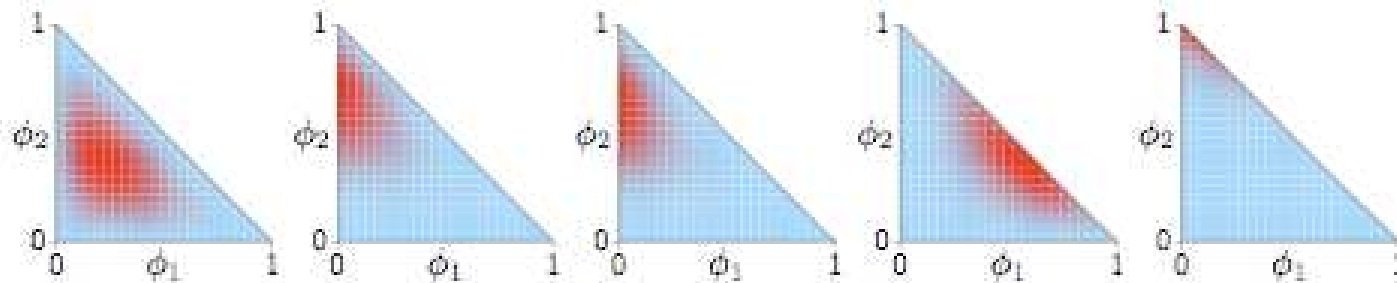
Dirichlet, $d=3$
(various parameter settings)



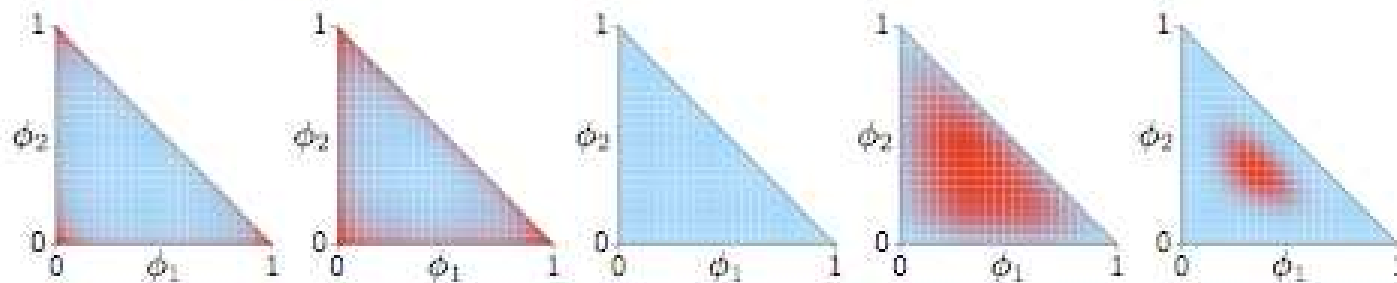
- from answers.com

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

Different means:



Different variances:



- from Liang and Klein, 2007

Why Dirichlet?

- Note that the Beta distribution is a specific case of a Dirichlet
- The Dirichlet distribution is conjugate to the multinomial distribution
- This means that identifying the posterior will again be easy, just like with the coin toss example

Posterior for Dirichlet

$X = (0, \dots, 1, \dots, 0)$ representing a certain outcome

- Same derivation as we had with the Beta prior:

$$p(\theta \mid \alpha) \propto \prod_{i=1}^d \theta_i^{\alpha_i - 1} \qquad p(X \mid \theta) \propto \prod_{i=1}^d \theta_i^{X_i}$$

- Multiply those together:

$$p(\theta \mid X) \propto \left(\prod_{i=1}^d \theta_i^{X_i} \right) \times \left(\prod_{i=1}^d \theta_i^{\alpha_i - 1} \right) = \prod_{i=1}^d \theta_i^{X_i + \alpha_i - 1}$$

- This is a Dirichlet with parameters $X + \alpha$

Until now...

- We assumed that we observe all of the events in the multinomials
- Therefore, deriving the posterior was quite easy
- What if we wanted to move to the unsupervised case?

Part II: Unsupervised Learning with the Bayesian Approach

Posterior in the Unsupervised Case

Latent Dirichlet Allocation (Blei et al., 2003)

- Given: M (# documents), α (prior over topic distributions), β (per-topic unigram distributions)
- For $i = 1 \dots M$:
 - Choose N , the number of words (Poisson or whatever).
 - Choose a distribution θ_i over topics (Dirichlet(α)).
 - For $j = 1 \dots N$:
 - Choose a topic z_{ij} according to θ_i .
 - Choose a word w_{ij} according to $\beta[z_{ij}, *]$.

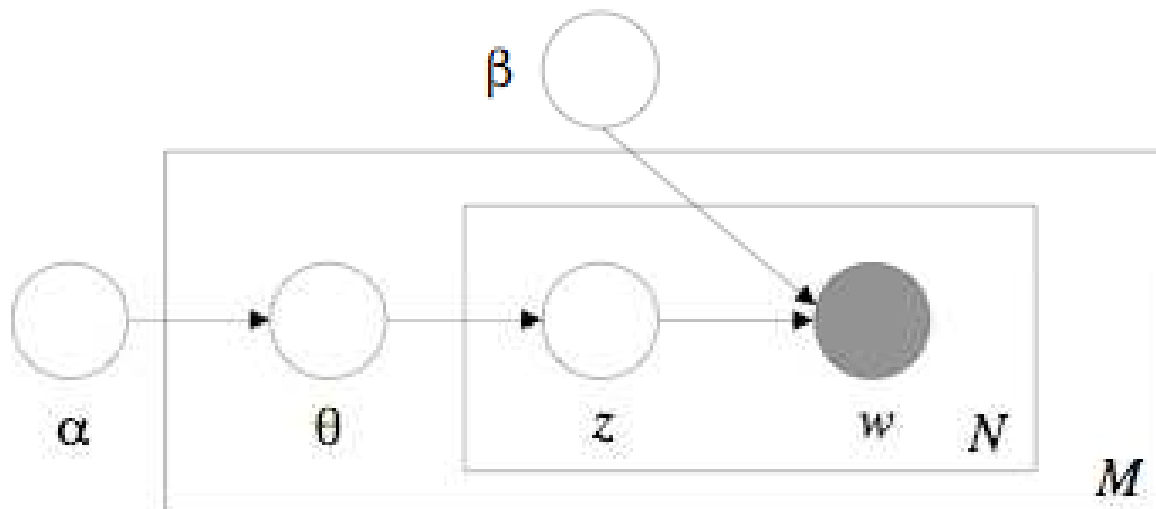
“Arts”	“Budgets”	“Children”	“Education”
--------	-----------	------------	-------------

NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Blei et al. (2003)

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

Graphical Model (LDA)

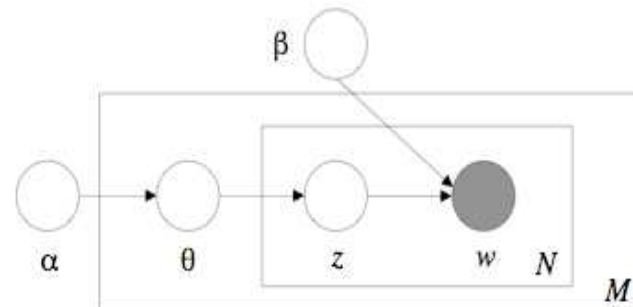


Posterior Inference with the LDA model

- Need to identify:

$$p(\theta, z \mid w, \beta, \alpha)$$

- Same problem as before
- The coupling between theta and z makes identifying the posterior hard



Posterior Inference

- The problem is exacerbated with structured models
 - Hidden markov models
 - Probabilistic context-free grammars
 - etc.

Two Main Techniques for Bayesian Inference

- Randomization
 - Use random sampling to approximate distributions.
 - Example: Gibbs sampling.
- Variational methods
 - Define a simpler, more factored model and fit that model to the true one.
 - Turns an intractable calculation into an optimization problem!
 - Example: mean-field approximation

Part III: Approximate Inference with Sampling

Gibbs Sampling

- Sampling directly from a complicated posterior distribution (always “given the evidence, \mathbf{x} ”) is hard.
- Consider all hidden variables (structure, parameters, etc.) as a set $\{y_1, y_2, \dots, y_n\}$. Initialize everything.
- A single iteration:
 - For $i = 1 \dots n$:
 - Randomly sample $y_i \sim p(y_i \mid \mathbf{x}, y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n)$.
- After “enough iterations,” you will be sampling from the joint distribution over all y_i .

Markov Chain Monte Carlo

- Think of every possible assignment to \mathbf{y} as a state in a huge (maybe infinite) Markov model.
- Gibbs sampling (and other MCMC methods, such as Metropolis-Hastings) equate to random walks on this highly connected graph.
- The theory of Markov chains tells us that under certain conditions, we will eventually reach the stationary distribution (which is the joint $p(\mathbf{y} \mid \mathbf{x})$ we want)
- “Monte Carlo” = randomized algorithm whose probability of error can be bounded arbitrarily by repeated (randomized) runs. (Cf. “Las Vegas” = always correct, but runtime varies.)

A Bayesian HMM for POS Tagging (Goldwater and Griffiths, 2007)

- Given: Dirichlet distribution with hyperparameter α (prior over tag trigram distributions), Dirichlet distribution with hyperparameter β (prior over emission distributions)
- Pick trigram tag parameters γ according to α
- Pick emission parameters η according to β
- Sample (\mathbf{t}, \mathbf{w}) from the HMM defined by (γ, η)
- Goldwater and Griffiths (2007) fix α and β to be symmetric (single scalar parameter).
- For now, assume α and β are fixed; just want to do inference over the tags for our corpus.

Gibbs Sampling for HMM


Gibbs Sampling for the Bayesian HMM

- Think of each tag as a hidden variable.
- We could think of \mathbf{y} and $\mathbf{\eta}$ as hidden variables, too, but there is a nice way to collapse them out.
- The reason for doing this: when we do not have to sample the parameters, our sampler “mixes” faster

Integrating Out $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$

- Assume a multinomial $\boldsymbol{\theta}$ with symmetric Dirichlet prior β and observables \mathbf{x}

$$\begin{aligned} p(X_i = x \mid \mathbf{x}_{-i}, \beta) &= \int p(x \mid \theta) p(\theta \mid \mathbf{x}_{-i}, \beta) d\theta \\ &= \frac{\text{count}(x; \mathbf{x}_{-i}) + \beta}{n - 1 + M\beta} \end{aligned}$$

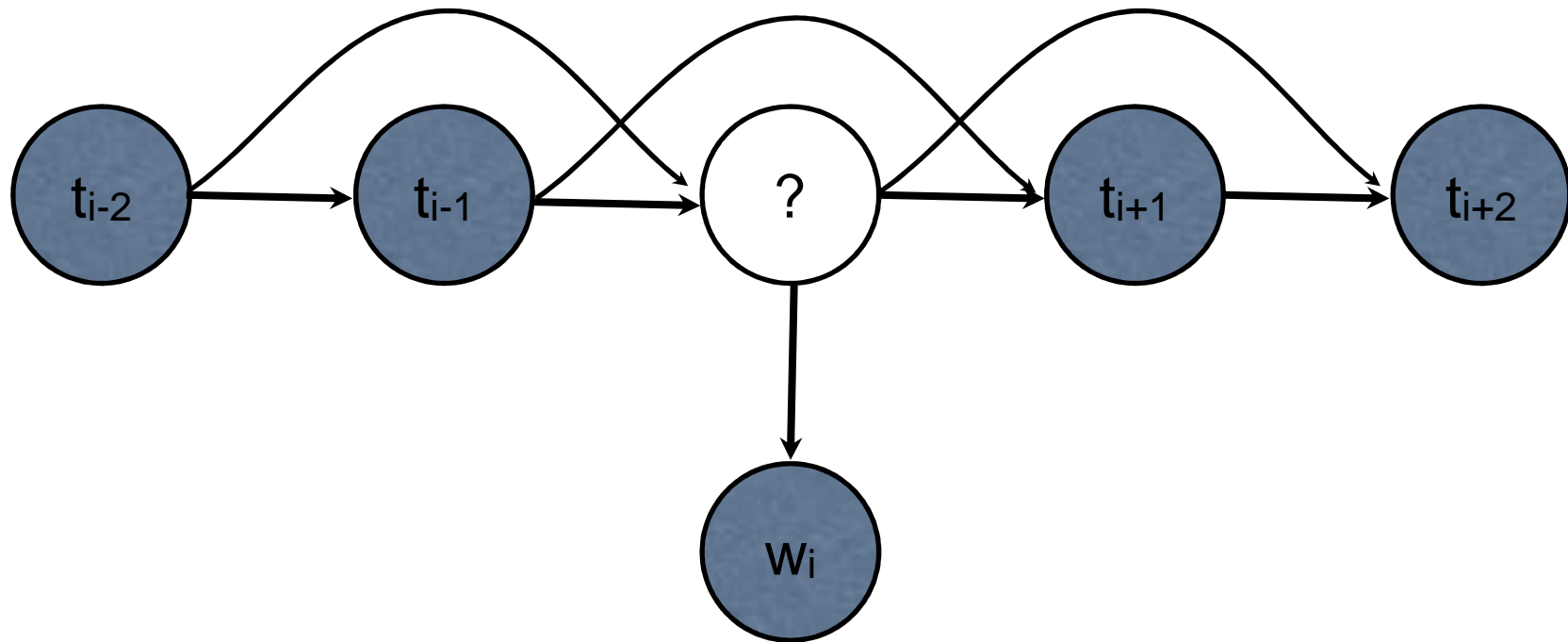
 “all \mathbf{x} excluding x_i ”

- Derivation in MacKay and Peto (1995)
- The property of exchangeability lets us pretend any x_i is last!

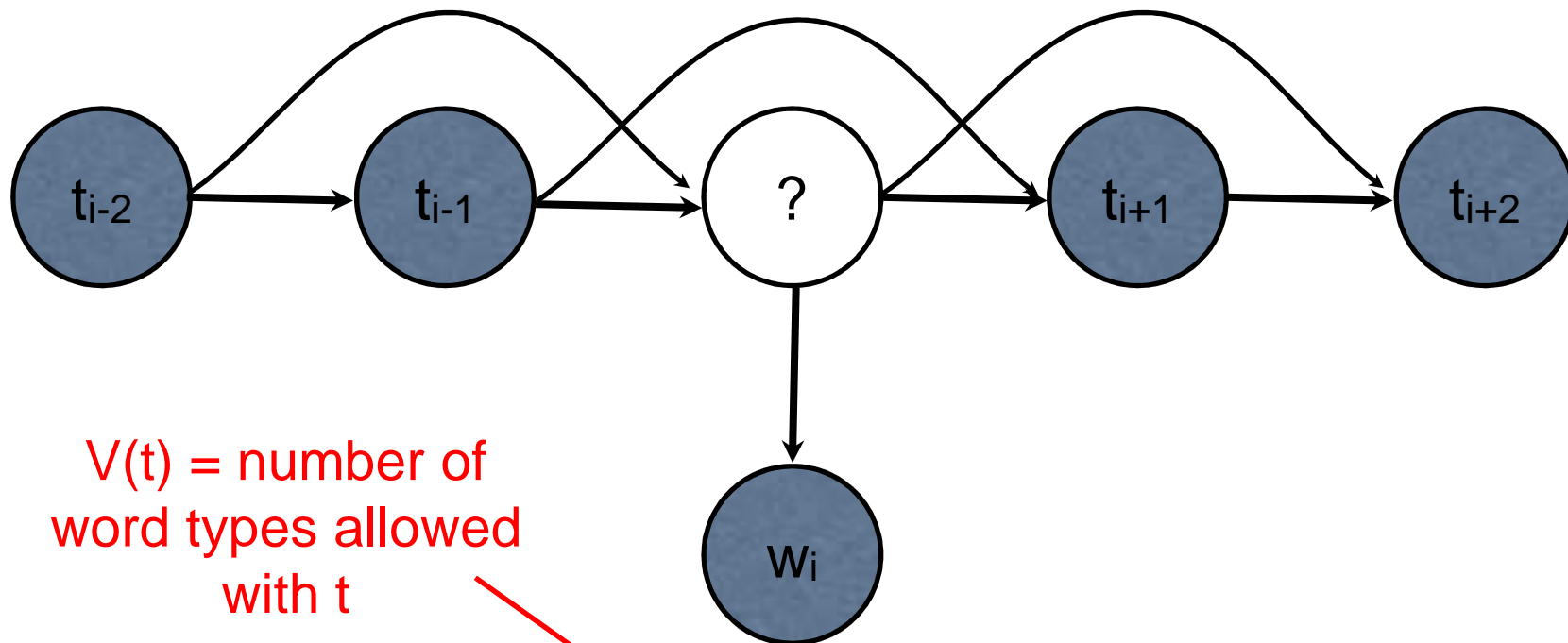
Gibbs Sampling for the Bayesian HMM

- Think of each tag as a hidden variable.
- We could think of \mathbf{y} and $\mathbf{\eta}$ as hidden variables, too, but there is a nice way to collapse them out.
- What's always fixed are the words and the prior: \mathbf{w} , $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$
- We want to sample from $p(\mathbf{t} \mid \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
- To do this, we iteratively sample from $p(t_i \mid \mathbf{t}_{-i}, \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
- Key issue: the probability that $T_i = t_i$ depends on the number of times the trigram $t_{i-2}t_{i-1}t_i$ has appeared elsewhere!
- This is because we are implicitly integrating out \mathbf{y} and $\mathbf{\eta}$.

Sampling 1 Tag



Sampling 1 Tag



$$p(t \mid \mathbf{t}_{-i}, \mathbf{w}, \alpha, \beta) \propto \frac{\text{count}(t, w_i; \mathbf{t}_{-i}, \mathbf{w}) + \beta}{\text{count}(t; \mathbf{t}_{-i}) + V(t)\beta} \times \frac{\text{count}(t_{i-2}, t_{i-1}, t; \mathbf{t}_{-i}) + \alpha}{\text{count}(t_{i-2}, t_{i-1}; \mathbf{t}_{-i}) + L\alpha} \\ \times \frac{\text{count}(t_{i-1}, t, t_{i+1}; \mathbf{t}_{-i}) + \alpha}{\text{count}(t_{i-1}, t; \mathbf{t}_{-i}) + L\alpha} \times \frac{\text{count}(t, t_{i+1}, t_{i+2}; \mathbf{t}_{-i}) + \alpha}{\text{count}(t, t_{i+1}; \mathbf{t}_{-i}) + L\alpha}$$

Metropolis-Hastings

- Sometimes it is not clear how to find the conditionals to do Gibbs sampling
 - Or they could be intractable to compute
 - Especially the normalization constant of the distribution
- Metropolis-Hastings is a way to do MCMC sampling when the normalization constant is hard to compute

Metropolis-Hastings

- Main idea: sample from a proposal distribution and correct by accepting/rejecting the samples to get samples from the real distribution
- To traverse the Markov chain:
 - At time step t , sample y' from a proposal distribution
 - Sample α from uniform $[0,1]$
 - Set $y_{t+1} = y'$ if

$$\alpha < \frac{p(y)q(y_t; y')}{p(y_t)q(y'; y_t)}$$

Gibbs Sampling and Metropolis-Hastings

- Gibbs sampling can actually be thought of as a specific case of Metropolis-Hastings
- The proposal distributions are the conditional distributions
- Samples are “always accepted”

More on Sampling in Grammars

- Johnson, Griffiths, and Goldwater (2007) explore samplers for Bayesian PCFGs (more general).
- Gibbs sampler that samples the probabilities (not collapsed)
- Metropolis-Hastings sampler (much faster; useful when posteriors are really tricky)
- Inside algorithm is a subroutine (they sample whole trees all at once)

Not Just for Unsupervised Learning!

- Gibbs sampling has also been used to approximate inference in **supervised models**, such as CRFs, when the “feature window” is not very local.
- The problem there is different, in that you’re not trying to integrate anything out ...
- ... but similar, because you’re dealing with non-local dependencies between the random variables!
- See work by Finkel and Manning.

Convergence

- Sampling is notoriously slow
- In fact, you cannot necessarily be certain when the Markov chain has “mixed”
- There are several heuristics:
 - If you can evaluate the likelihood, check if it stabilizes
 - If there is an extrinsic evaluation, you can evaluate on that measure
 - Run several Markov chains, and compare a scalar parameter across all these chains
- In NLP, in many cases, we run a sampler for a fixed number of iterations that is chosen ahead

Sampling Pros and Cons

- If you run it long enough, you will be drawing samples from the correct posterior.
- You can get approximate MAP inference, too: gradually make the distributions more and more “peaked” by *annealing* (raise each probability to a power)
- It can be very, very slow.
- You don’t know when your sampler has “mixed” or “burned in.”

Part IV: Variational Approximation

The Basic Idea

- An alternative to MCMC sampling
- Reminder: we are looking for the posterior

$$p(\theta, z \mid x)$$

- Variational approximation estimates this posterior by solving (for some family of distributions Q):

$$\arg \min_{q \in Q} \text{KL}(q \parallel p(\theta, z \mid x))$$

Variational Approximation

- The KL divergence is “the variational bound”

$$\arg \min_{q \in \mathcal{Q}} \text{KL}(q || p(\theta, z | x))$$

- Variational inference done this way is a specific instantiation of a principle in which we represent a log partition function (such as marginalized likelihood) as a maximization problem
- See Wainwright and Jordan (2008) for more details

Mean Field Approximation

- We need to decide what is the family Q of distributions we use
- One approximation that in many cases leads to tractable solutions is the mean-field approximation

$$Q = \{q(\theta, z) = q(\theta)q(z)\}$$

- Assumes a factorized posterior distribution

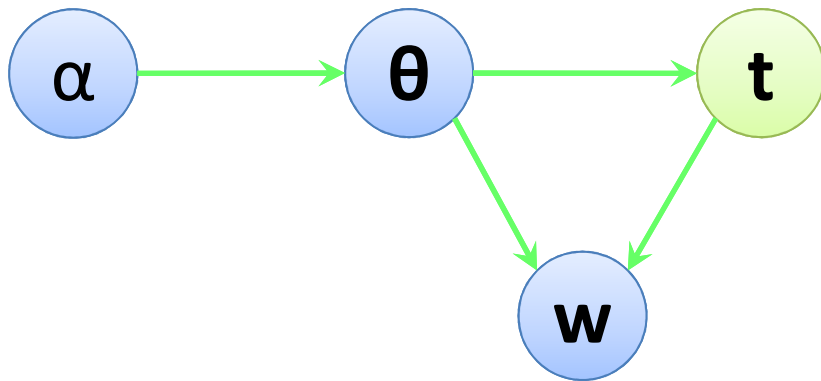
Finding the Approximate Posterior

variational
inference:

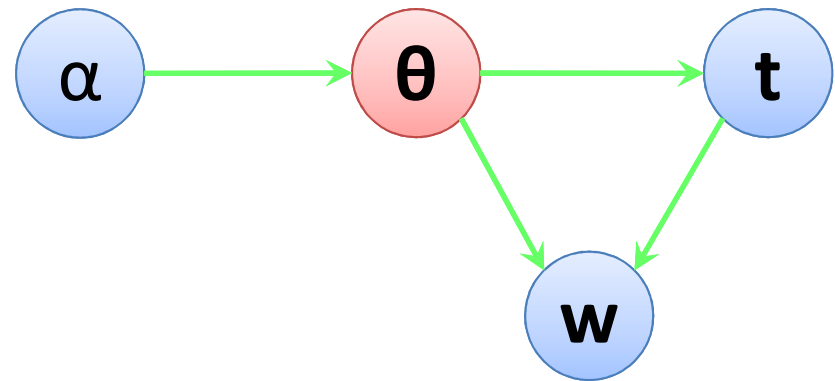
$$\arg \min_{q \in \mathcal{Q}} \text{KL}(q || p(\theta, z | x))$$
$$\mathcal{Q} = \{q(\theta, z) = q(\theta)q(z)\}$$

- Finding the approximate posterior involves alternating between two steps:
 - E-step: identifies $q(z)$
 - M-step: identifies $q(\theta)$
- Very similar to EM (for a good reason)

Reminder: EM

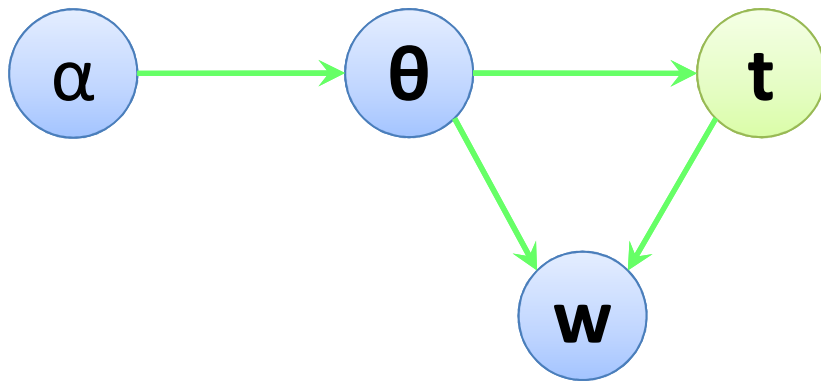


E step

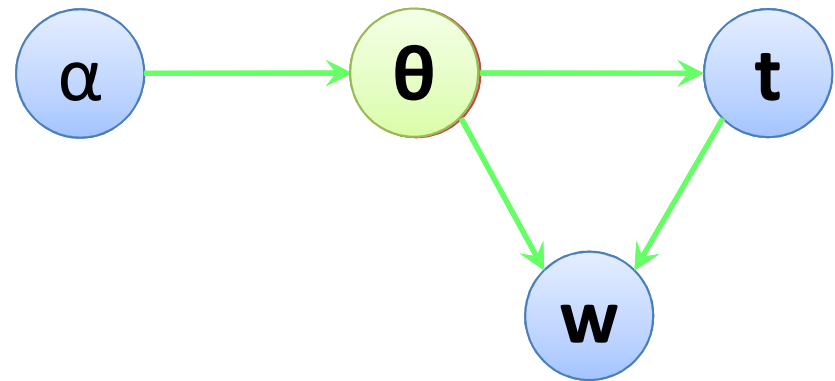


M step

Variational EM



E step



M step

Connection to EM

- We can choose the family of approximate posteriors over the parameters $q(\theta)$ to be distributions that put probability mass 1 on a single parameter
- Reduces variational inference to the EM algorithm

Test case: HMMs

$$\mathcal{Q} = \{q(t, \gamma, \eta) = q_1(t)q_2(\gamma, \eta)\}$$

- “E step”: get posterior over model events (transitions and emissions in the HMM), $q_1(\mathbf{t})$
 - Most convenient: q_1 consists of an unnormalized HMM
- “M step”: get posterior over model parameters, $q_2(\boldsymbol{\gamma}, \boldsymbol{\eta})$
 - Most convenient: q_2 consists of Dirichlets
- This is really just a coordinate ascent algorithm that can be derived from the variational bound!

An EM-like algorithm (M step)

- EM:

$$\theta_i \leftarrow \frac{\mathbb{E}[\# \text{ event } i]}{\mathbb{E}[\# \text{ events where } i \text{ was possible}]}$$

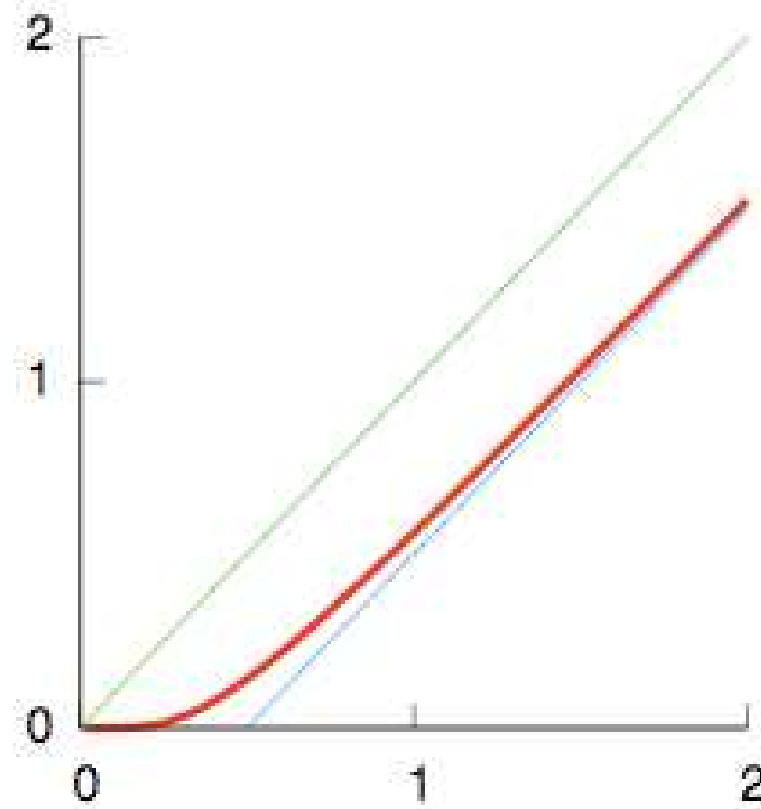
- EM with prior:

$$\theta_i \leftarrow \frac{\mathbb{E}[\# \text{ event } i] + \alpha_i - 1}{\mathbb{E}[\# \text{ events where } i \text{ was possible}] + \sum_j \alpha_j - d}$$

- Variational inference:

$$w_i \leftarrow \frac{\exp \Psi(\mathbb{E}[\# \text{ event } i] + \alpha_i)}{\exp \Psi(\mathbb{E}[\# \text{ events where } i \text{ was possible}] + \sum_j \alpha_j)}$$

 use as model weights on E step



exp-digamma function

from Johnson (2007)

Empirical Bayes

- Until now, we focused on the problem of identifying the posterior
- We also have to decide what would be the hyperparameters
- For example, what would be α when using a Dirichlet?
- Estimating these kinds of hyperparameters is called “empirical Bayes”

Variational EM

- Derive a variational bound again
- This bound depends on the hyperparameters
- Variational EM:
 - E-step: Maximize the bound by finding $q(z)$
 - M-step: Maximize the bound by finding $q(\theta)$
- Outer M-step: Maximize the bound by changing the hyperparameters

Note

- The problem of deriving the variational bound and optimizing it are two separate problems
- In principle, you could also use gradient descent algorithms to optimize the variational bound
- Or any other type of optimization technique

Bayesian

- Parameters are part of the hidden variables
- Mathematically elegant
- Can be computationally complex
- Can help overfitting
- Many frequentist techniques can be described as a Bayesian with a special prior

Frequentist

- Parameters are estimated with a point estimate
- Needs new notions to talk about “goodness” of estimator, convergence, etc.
- Has the interpretation of “repeated experiments”

Things We Omitted

- Nonparametric methods based on the Dirichlet process
 - Generally helps deciding on “number of clusters” automatically
 - Can be used to decide on the number of topics for LDA
 - Can be used for latent annotation for symbols in a PCFG (Liang et al., 2007; Finkel et al., 2007)
 - Adaptor grammars (Johnson et al., 2007)
- Non-conjugate priors
 - Priors which do not satisfy the property of Dirichlet with respect to the multinomials
 - Richer priors, and can prove to be much more useful than the Dirichlet distribution

Summary

- The Bayesian approach helps introduce some bias into the learning process
- This is done by setting a prior on the parameters
- In this lecture, we focused mostly on the Dirichlet prior because it is conjugate
- There is work that shows we can go beyond that
- Sampling and variational inference are the main tools to do inference in the Bayesian setting