Bayesian Text Analysis
Comptext 2025

Petro Tolochko

University of Vienna

What Is Bayesian Text Analysis?

- Applying Bayesian inference to model text data with explicit probability distributions
- Treats model parameters (e.g., topic weights, sentiment mixtures) as random variables
- Combines prior knowledge (lexicons, hierarchical structure) with observed word counts
- Full posterior distributions $\rightarrow$ credible intervals & uncertainty quantification

Motivation

- Uncertainty quantification in text analytics (full posterior distribution)
- Natural incorporation of statistical models
- Incorporation of prior knowledge
- Smoothing (with priors)
- Hierarchical models

Bayes' Theorem

$$P(\theta \mid D) = \frac{P(D \mid \theta)\, P(\theta)}{P(D)}$$

- Prior $P(\theta)$ prior distribution of the parameters
- Likelihood $P(D \mid \theta)$ represents the probability of observing the data given the parameters θ
- Posterior $P(\theta \mid D)$ Posterior distribution of the model parameters θ given the data D
- Marginal Likelihood $P(D)$ Probability of observing the data

Bayes' Theorem

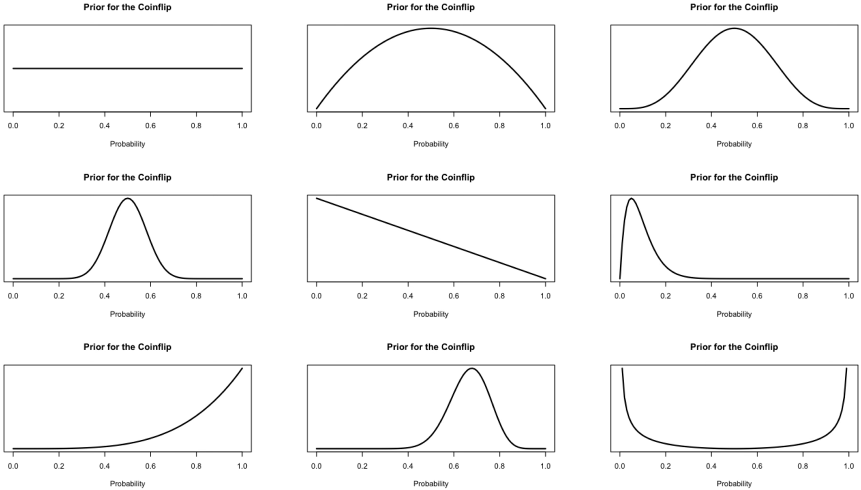
$$P(\theta \mid D) = \frac{P(D \mid \theta)\, P(\theta)}{P(D)}$$

- Prior $P(\theta)$ encodes domain knowledge
- Likelihood $P(D \mid \theta)$ measures data fit
- Posterior $P(\theta \mid D)$ captures uncertainty
- Posterior $P(D)$ normalisation constant

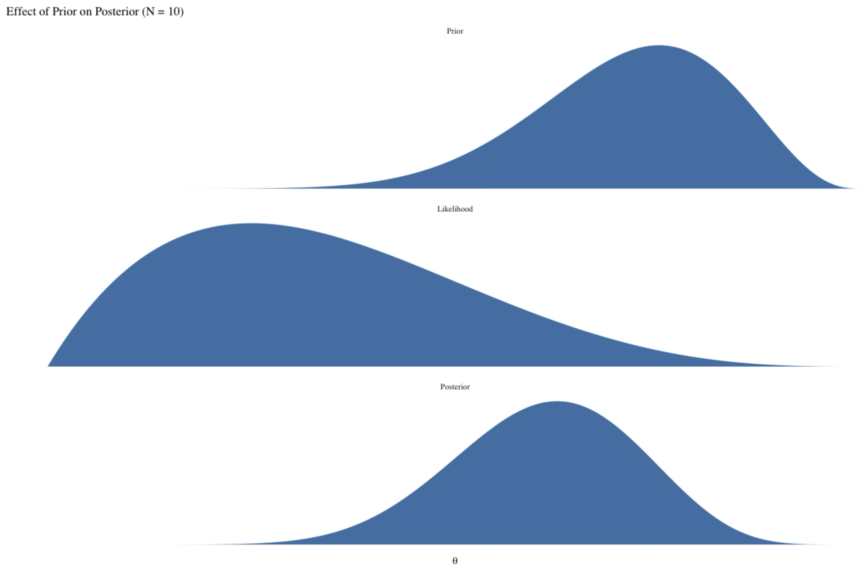
Bayesian Inference (priors)

- We need to assign priors for the model to work
- Priors are our beliefs about the event *prior* to seeing the data
- We can be very certain or very uncertain (or everything in between)

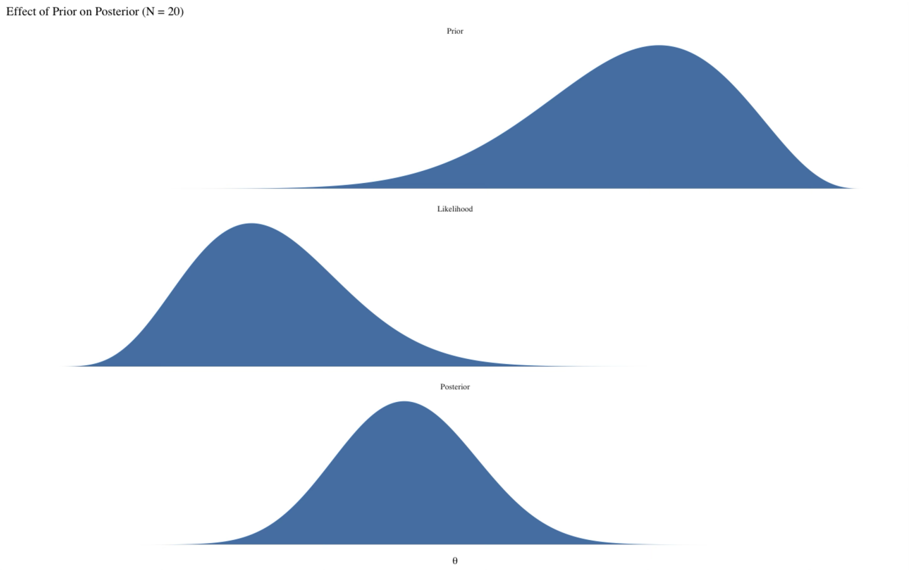
Priors (coinflip)



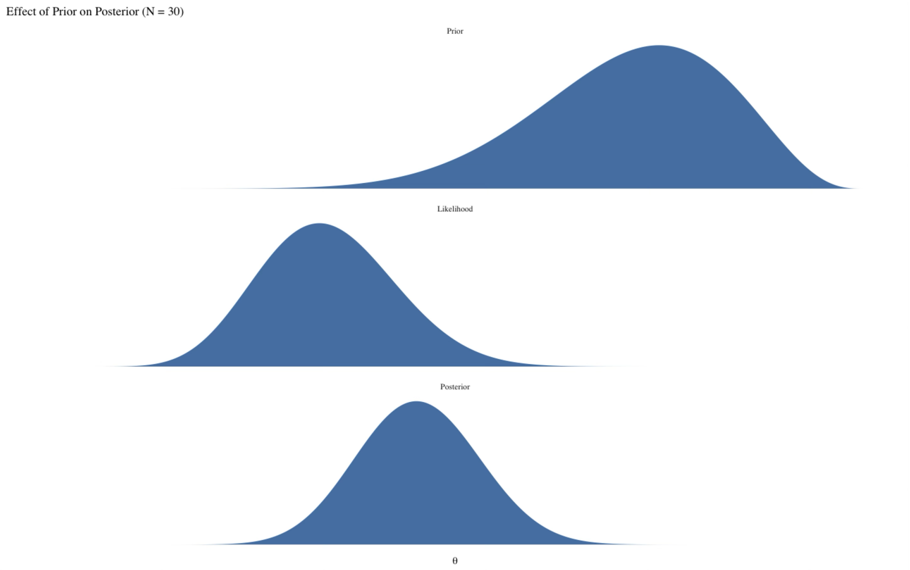
Effect of Prior on Posterior



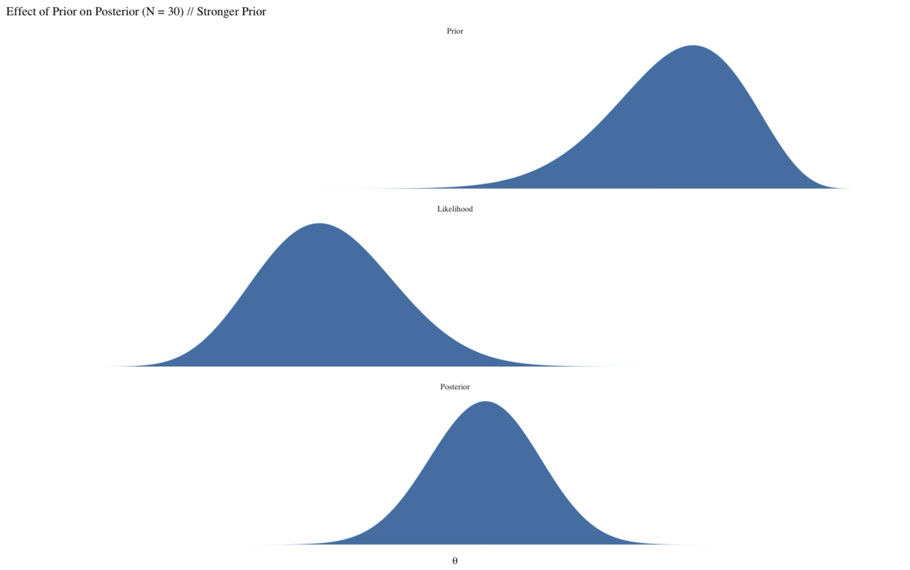
Effect of Prior on Posterior



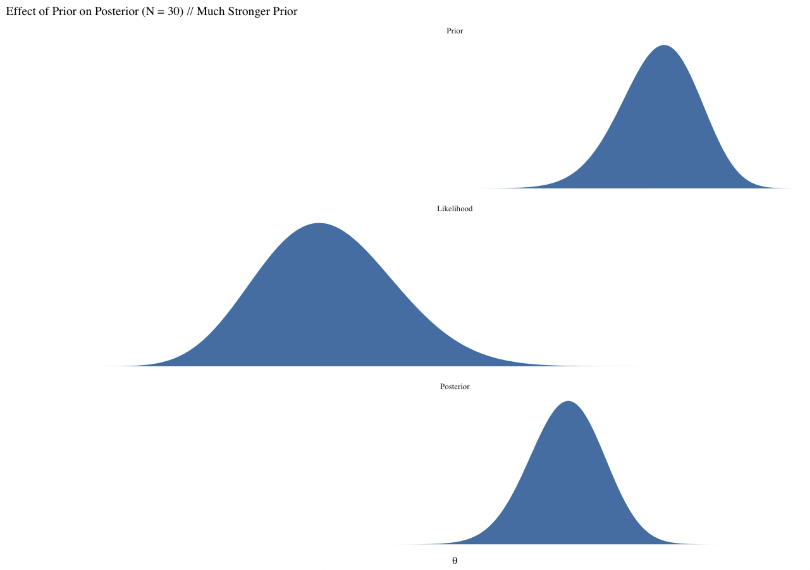
Effect of Prior on Posterior



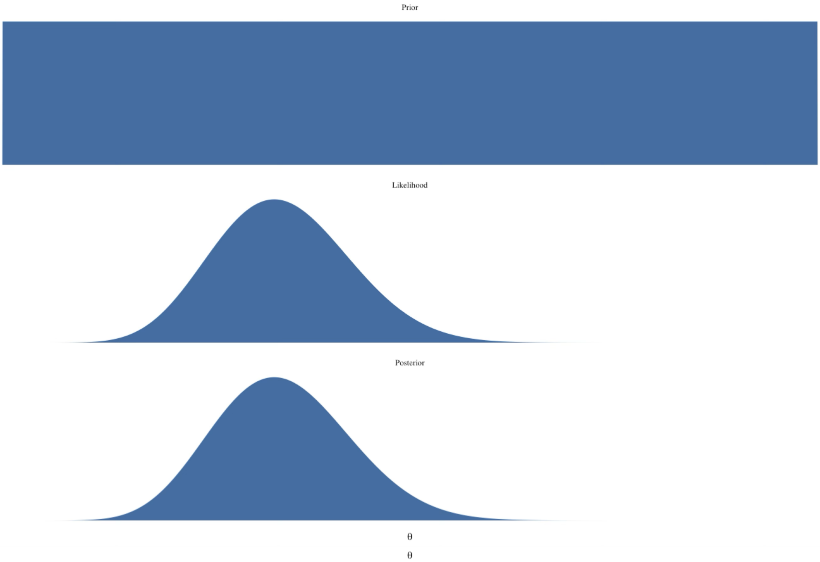
Effect of Prior on Posterior



Effect of Prior on Posterior



Effect of Prior on Posterior



Statistical Model

$$y = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k + \epsilon$$

Statistical Model: Components Explained

$$y \sim \text{Normal}(\mu,\ \sigma)$$

$$\mu = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

$$\beta_j \sim \text{Normal}(0,\ 5) \quad (j = 0, \ldots, k)$$

$$\sigma \sim \text{Exponential}(1)$$

Statistical Model: Components Explained

$$y \sim \mathrm{Normal}(\mu,\ \sigma)$$
$$\mu = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$
$$\beta_j \sim \mathrm{Normal}(0,\ 5) \quad (j = 0, \ldots, k)$$
$$\sigma \sim \mathrm{Exponential}(1)$$

- **Likelihood** $y \sim Normal(\mu, \sigma)$ How the data arise given μ, σ.
- **Linear predictor** $\mu = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$ Maps covariates X_j to the mean response.
- **Priors on coefficients** $\beta_j \sim \mathrm{Normal}(0, 5)$ Each β_j centered at 0 with moderate spread.
- **Prior on scale** $\sigma \sim \mathrm{Exponential}(1)$ Belief that $\sigma > 0$, mean 1.

Parameter Interpretation: Bayesian vs. Frequentist

Bayesian Inference

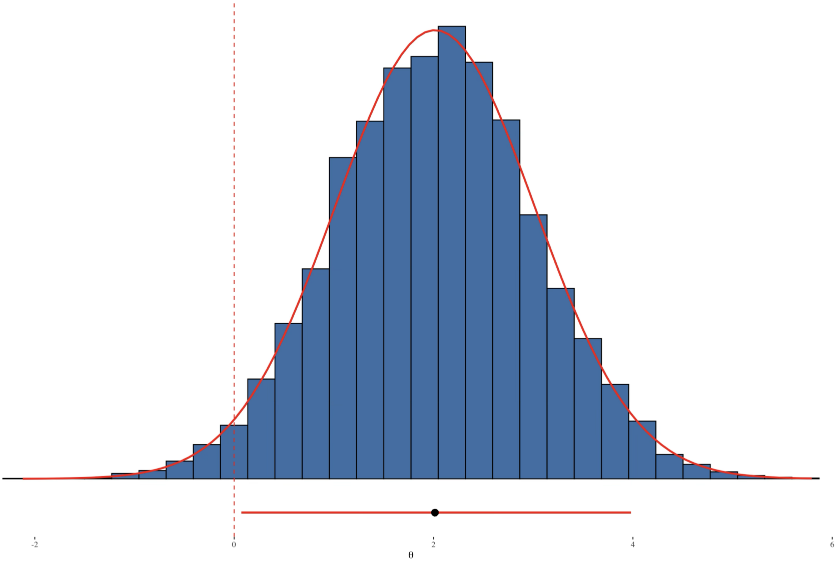
- *Parameters are random variables* with a distribution.
- Encode prior beliefs via a **prior** $P(\theta)$.
- Update beliefs with data via the **posterior** $P(\theta \mid D)$
- Uncertainty expressed by **credible intervals**: $\Pr(\theta \in C \mid D) = 0.95$
- Interpret intervals directly as "there's a 95% chance θ lies in C"

Frequentist Inference

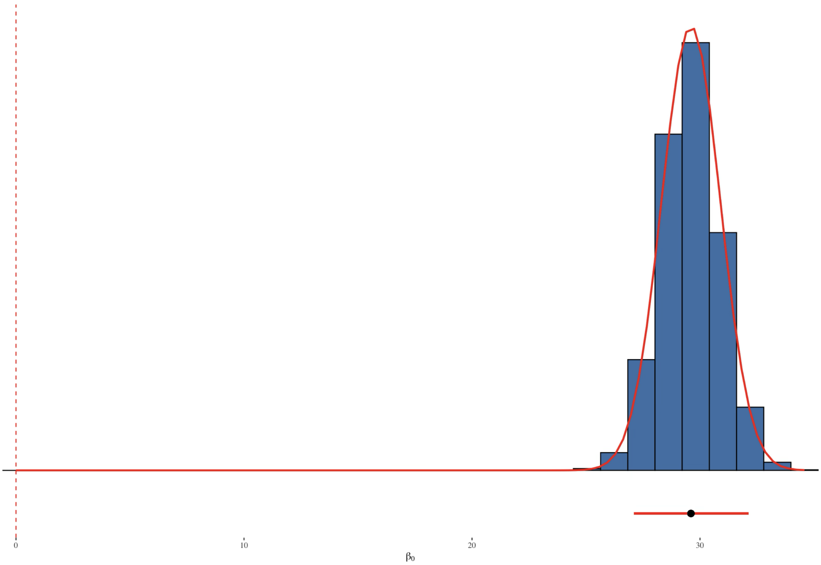
- *Parameters are fixed* but unknown constants.
- No prior; inference derives from the **likelihood** $P(D \mid \theta)$
- Estimation via point estimators $\hat{\theta}$ (e.g. MLE).
- Uncertainty quantified by the **sampling distribution** of $\hat{\theta}$.
- **Confidence intervals**: Under repeated sampling, 95% of CIs will cover the true θ.
- Cannot say "θ has 95% probability of lying in this interval."

Confidence Interval

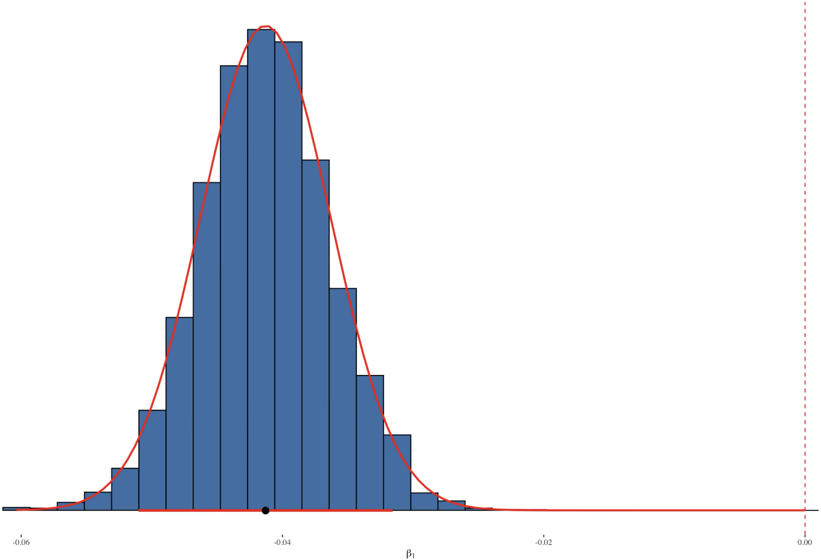
Posterior (Credibility Interval)



Posterior (Credibility Interval)



Posterior (Credibility Interval)



Probabilistic Programming: Overview

- **Definition:** A *probabilistic program* is code for random sampling and conditioning.
- Automates the core steps of Bayesian analysis:
  - Specifying generative models
  - Defining priors and likelihoods
  - Performing posterior inference
- "Intuitive" approach to complex hierarchical and latent-variable models.

Why Bayesian for Text Analysis?

- **Full uncertainty quantification** We get the full posterior over parameters (topic weights, embedding coefficients, etc.) instead of just point estimates.
- **Principled smoothing** Dirichlet- or Normal-priors naturally smooth sparse word counts or regression weights.
- **Prior knowledge incorporation** Encode beliefs (e.g. lexicon strengths, topic correlations) via informative priors.
- **Probabilistic Predictions** E.g., Posterior Predictive distributions

Explicit Statistical Modeling

- **Generative story** "Words $\sim$ Multinomial(topic mixture), topic mixture $\sim$ Dirichlet(α)" makes assumptions explicit.
- **Model comparison & checking** Bayes factors, WAIC/LOO, and posterior predictive checks.
- **Extensibility**

Hierarchical & Latent-Variable Benefits

- **Document-level random effects** Capture author or genre variability via hierarchical priors on topic proportions.
- **Word-level latent structure** Bayesian embedding models (e.g. neural variational LDA) recover nuanced semantic representations.
- **Cross-document coupling** Share statistical strength across documents, corpora, languages, or time via multi-level priors.
- **Uncertainty propagation** Downstream tasks (classification, prediction) inherit posterior uncertainty automatically.

Take-home, Flexible, Interpretable

- **Flexible:** Easily swap likelihoods (e.g. Poisson for counts, negative-binomial for overdispersion) or build new hierarchies
- **Interpretable:** Each parameter has a clear probabilistic role; credible intervals and posteriors speak directly to belief
- **Reproducible:** Full generative code (in Stan/Pyro/PyMC)

Laplace Smoothing vs Dirichlet Priors

Laplace Smoothing

$$\hat{\phi}_{k,j} = \frac{n_{k,j} + 1}{\sum_{j'}(n_{k,j'} + 1)}$$

Fixed κ, point estimates only.

Dirichlet Prior

$$\phi_k \sim \mathrm{Dirichlet}(\alpha)$$

Posterior: $\mathrm{Dir}(\alpha + n)$, with variance.

Latent Dirichlet Allocation: Model Specification

$$
\begin{aligned}
\theta_d &\sim \mathrm{Dirichlet}(\alpha), & d &= 1, \ldots, D \\
\phi_k &\sim \mathrm{Dirichlet}(\beta), & k &= 1, \ldots, K \\
z_{d,n} &\sim \mathrm{Categorical}(\theta_d), & n &= 1, \ldots, N_d \\
w_{d,n} &\sim \mathrm{Categorical}(\phi_{z_{d,n}}), & n &= 1, \ldots, N_d
\end{aligned}
$$

- α: concentration hyperparameter for document–topic Dirichlet prior
- β: concentration hyperparameter for topic–word Dirichlet prior
- θ_d: topic proportion vector for document d
- ϕ_k: word probability vector for topic k
- $z_{d,n}$: latent topic assignment of the nth word in document d
- $w_{d,n}$: observed nth word in document d, drawn from topic $z_{d,n}$

Naive Bayes Classifier

- For a document $d = (w_1, \ldots, w_N)$, posterior over class c:

$$P(c \mid d) \;\propto\; P(c) \prod_{i=1}^{N} P(w_i \mid c)$$

- Conditional independence: words w_i independent given class c
- Priors $P(c)$: frequency of each class in training data
- Likelihoods $P(w \mid c)$: estimated from word counts with Laplace (add-1) smoothing
- Prediction: choose $\hat{c} = \arg\max_c P(c \mid d)$
- Not *fully* Bayesian, relies on MLE estimates

Fightin' Words Methodology (Monroe et al., 2008)

- Objective: Identify words that most distinguish two corpora (A vs. B)
- Model counts with Dirichlet–Multinomial for each corpus:

$$\mathbf{n}^{(X)} \sim \mathrm{Multinomial}(N^{(X)}, \mathbf{p}^{(X)}), \quad \mathbf{p}^{(X)} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}),\ X \in \{A, B\}$$

- Define log-odds difference for each word:

$$\delta_j = \log \frac{p_j^{(A)}}{1 - p_j^{(A)}} - \log \frac{p_j^{(B)}}{1 - p_j^{(B)}}.$$

- Approximate posterior of δ_j as Gaussian via Beta marginals
- Rank words by standardized effect: $z_j = \hat{\delta}_j / \mathrm{sd}(\delta_j)$

Fighting Words vs. Raw Log-Odds

- **Raw Log-Odds:**

$$\log \frac{n_j^{(A)}/N^{(A)}}{n_j^{(B)}/N^{(B)}}$$

    - No smoothing $\rightarrow$ infinite or undefined for zero counts
    - Lacks uncertainty quantification
- **Fighting Words:**
    - Bayesian smoothing via Dirichlet prior
    - Provides posterior mean and variance of δ_j
    - Enables credible intervals and z-scores
- **Key difference:** principled shrinkage + uncertainty vs. unstable point estimates

Example: Unsupervised Sentiment Model

Building Sentiment-Informed Priors

- For each vocabulary term v, extract its AFINN score s_v (zero if absent).
- Define two Dirichlet concentration vectors:
  - $\alpha_v^+ = 1 + \kappa \cdot \max(s_v, 0)$, boosting words with positive scores.
  - $\alpha_v^- = 1 + \kappa \cdot \max(-s_v, 0)$, boosting words with negative scores.
- The scaling constant κ (e.g. 15) controls how strongly lexicon scores influence the prior.
- These priors encourage our "positive" word distribution to favor lexicon-positive words, and similarly for "negative."

Mixture-of-Multinomials Sentiment Model

- We posit two latent word distributions: ϕ_{neg} and ϕ_{pos}, each drawn from their Dirichlet priors.
- A global mixing probability $\pi \sim \mathrm{Beta}(1, 1)$ governs the overall chance a document is "positive."
- Each document's counts C_d arise by first flipping a latent component (neg/pos) then drawing words from the corresponding multinomial.
- This captures the idea that each document is either predominantly negative or positive in tone, with uncertainty.

Example: Semi-Supervised Model

- Goal: leverage both labeled and unlabeled data in a single generative model

Model Priors

- $\pi \sim \mathrm{Beta}(1, 1)$: prior on the probability a document is "positive."
- Two sentiment-specific word distributions:

$$\phi_{\text{neg}} \sim \mathrm{Dirichlet}(\alpha^{-}), \quad \phi_{\text{pos}} \sim \mathrm{Dirichlet}(\alpha^{+}).$$

- Dirichlet priors $\alpha^{\pm}$ are constructed from AFINN scores to bias word probabilities.
- These priors encode our lexicon-informed beliefs before seeing any documents.

Labeled-Data Likelihood

- For each labeled document d with $y_d = 0$ (negative):

$$C_d \sim \mathrm{Multinomial}\big(n_d,\ \phi_{\text{neg}}\big).$$

- For each labeled document with $y_d = 1$ (positive):

$$C_d \sim \mathrm{Multinomial}\big(n_d,\ \phi_{\text{pos}}\big).$$

- We "fix" the latent component to the observed label, effectively conditioning on true sentiment.
- This uses our labeled subset to anchor the two word distributions.

Unlabeled-Data Mixture Likelihood

- For each unlabeled document d, we don't know its sentiment component.
- We model its counts as a mixture:

$$C_d \sim (1-\pi)\,\mathrm{Mult}(n_d, \phi_{\text{neg}}) \;+\; \pi\,\mathrm{Mult}(n_d, \phi_{\text{pos}}).$$

- The mixture weight π links labeled and unlabeled parts of the model.
- This lets unlabeled documents inform both the global sentiment frequency and word distributions.

Drawbacks of Bayesian Framework for Text Analysis

- Computational cost
- Sensitivity to priors
- Scalability
- Implementation complexity

Questions?