Models of Annotation (II)

Bob Carpenter, *LingPipe, Inc.*

Massimo Poesio, *Uni. Trento*

LREC 2010 (Malta)

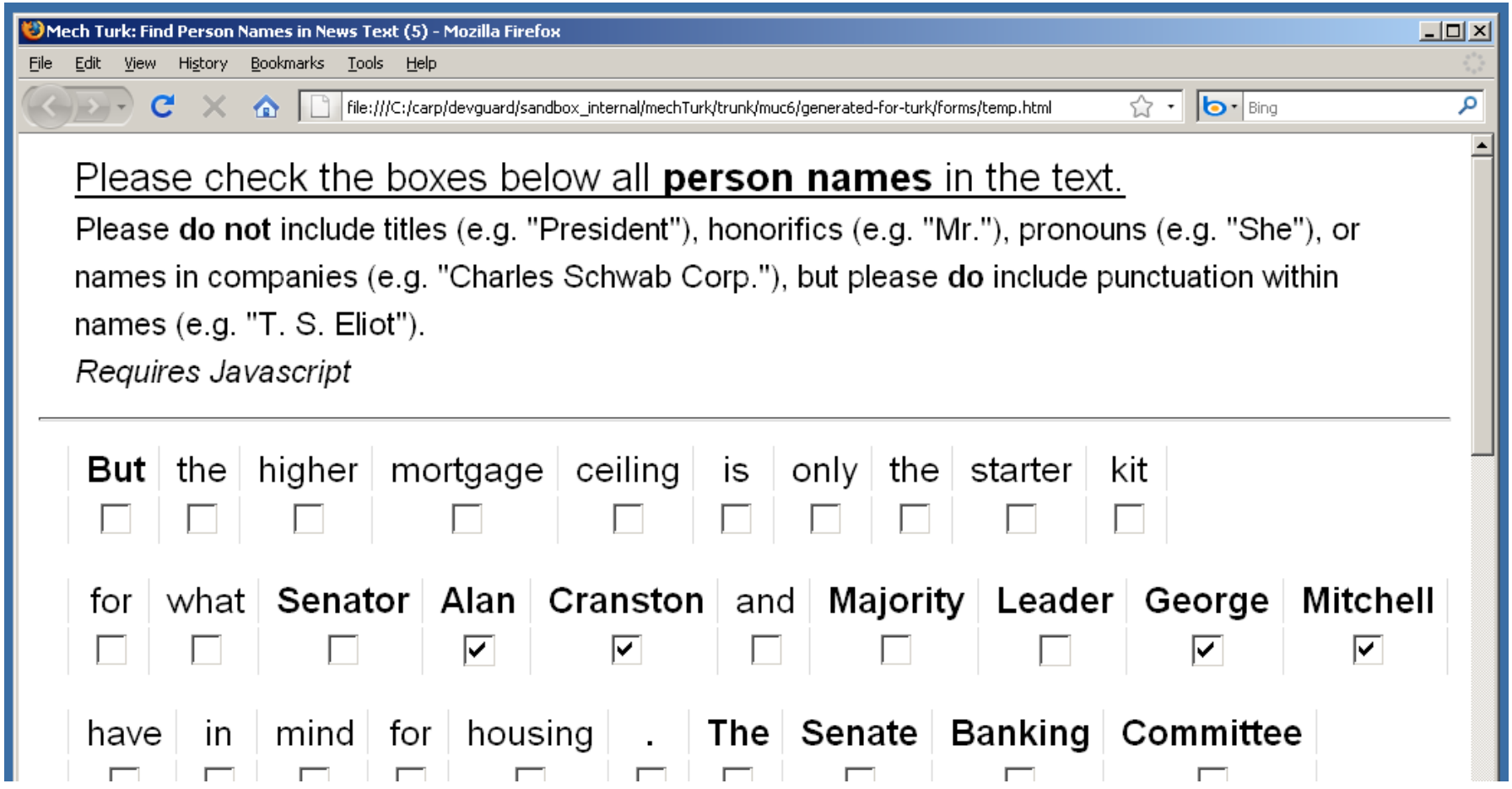
Mechanical Turk Examples

(Carpenter, Jamison and Baldwin, 2008)

Amazon's Mechanical Turk

- "Crowdsourcing" Data Collection (Artificial AI)
- We provide web forms to Turkers (through REST API)
- We may give Turkers a qualifying/training test
- Turkers choose tasks to complete
  - We have no control on assignment of tasks
  - Different numbers of annotations per annotator
- Turkers fill out a form per task and submit
- We pay Turkers through Amazon
- We get results from Amazon in a CSV spreadsheet

Case 1: Named Entities



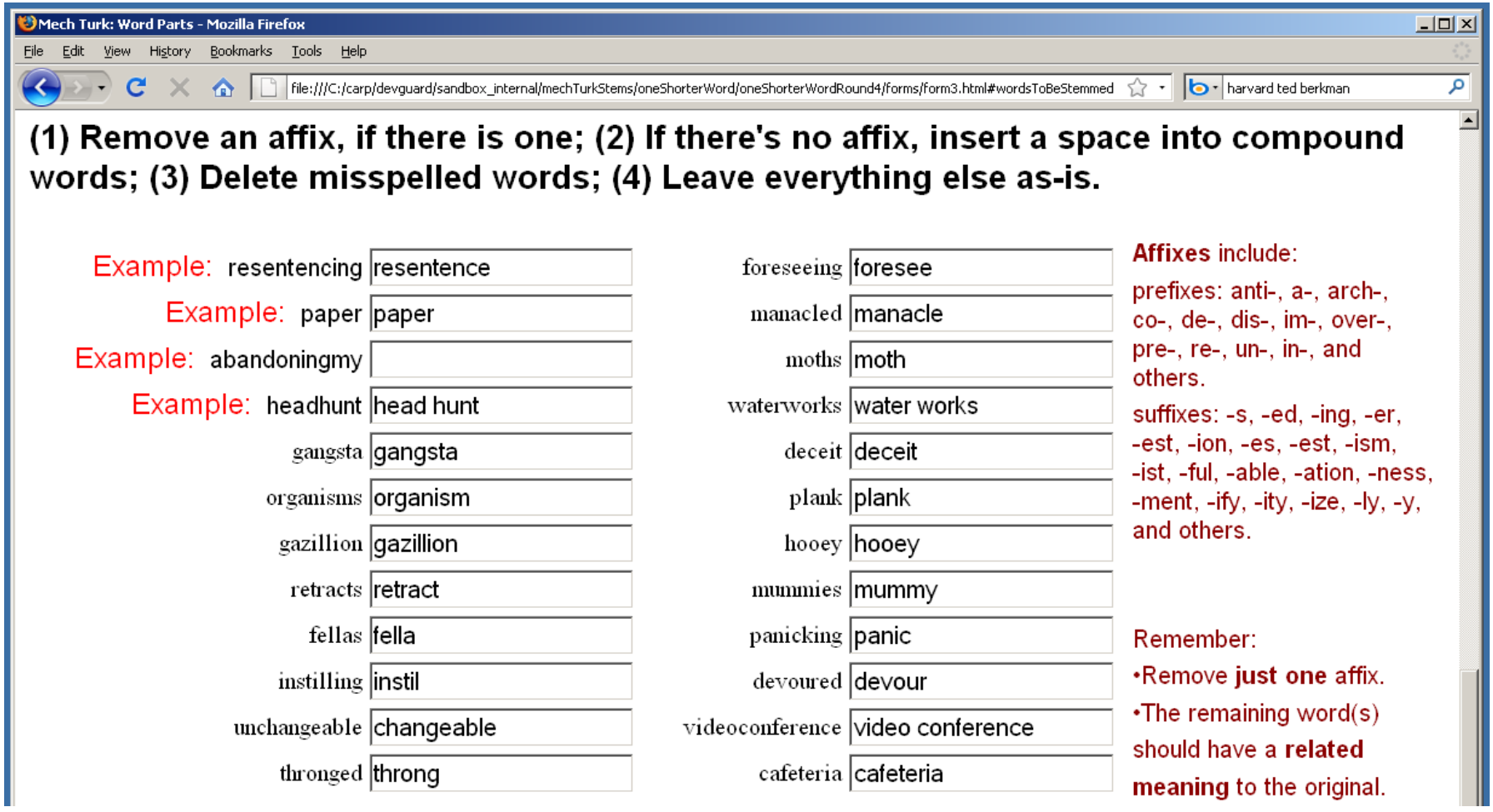
Named Entities Worked

- Conveying the coding standard
  - official MUC-6 standard dozens of pages
  - examples are key
- Fitts's Law
  - time to position cursor inversely proportional to target size
  - highlighting text: fine position + drag + position
  - pulldown menus for type: position + pulldown + select
  - checkboxes for entity at a time: fat target click

Discussion: Named Entities

- 190K tokens, 64K capitalized, 4K person name tokens
  - 4K / 190K = 2.1% prevalence of entity tokens
- 10 annotators per token
- 100+ annotators, varying numbers of annotations
- Less than a week at 2 cents/400 tokens (US$95)
- Aggregated Turkers better than LDC data
  - Correctly Rejected: `Webster's, Seagram, Du Pont, Buick-Cadillac, Moon, erstwhile Phineas Foggs`
  - Incorrectly Accepted: `Tass`
  - Missed Punctuation: `J E. ''Buster'' Brown`

Case 2: Morphological Stemming



Morphological Stemming Worked

- Coded and tested by intern (Emily Jamison of OSU)
  - Less than one month to code, modify and collect
- Three iterations of coding standard, Four of instructions
  - began with full morphological segmentation (too hard)
  - simplified task to one stem with full base (more “natural”)
  - added previously confusing examples and sample affixes
- Added qualifying test
- 60K (50K frequent Gigaword, 10K random) tokens
- 5 annotators / token

Generative Labeling Model

(Dawid and Skene 1979; Bruce and Wiebe 1999)

Assume Binary Labeling for Simplicity

- 0 = “FALSE”, 1 = “TRUE” (arbitrary for task)
  - e.g. Named entities: Token in Name = 1, not in Name = 0
  - e.g. RTE-1: entailment = 1, non-entailment = 0
  - e.g. Information Retrieval: relevant=1, irrelevant=0
- Models generalize to more than two categories
  - e.g. Named Entities: PERS, LOC, ORG, NOT-IN-NAME
- Models generalize to ordinals or scalars
  - e.g. Paper Review: 1-5 scale
  - e.g. Sentiment: 1-100 scale of positivity

Prevalence

- Assumes binary categories (0 = “FALSE”, 1 = “TRUE”)
- Prevalence π is proportion of 1 labels
  - e.g. RTE-1 400/800 = 50% [artificially “balanced”]
  - e.g. Sports articles (among all news articles): 15%
  - e.g. Bridging anaphors (among all anaphors): 6%
  - e.g. Person named entity tokens 4K / 190K = 2.1%
  - e.g. Zero (tennis) sense of “love” in newswire: 0.5%
  - e.g. Relevant docs for web query [Malta LREC]:
    500K/1T = 0.00005%

Gold-Standard Estimate of Prevalence

- Create gold-standard labels for a subset of data
  - Choose the subset randomly from all unlabeled data
  - Otherwise, may result in biased estimates
- Use proportion of 1 labels for prevalence π [MLE]
- More data produces more accurate estimates
  - For N examples with prevalence π, 95% interval is

    $$\pi \pm 2\sqrt{\frac{\pi(1-\pi)}{N}}$$
  - e.g. 100 samples, 20 positive, $\pi = 0.20 \pm 0.08$
  - Given fixed prevalence, uncertainty inversely proportional to $\sqrt{N}$
  - The law of large numbers in action

Accuracies: Sensitivity and Specificity

- Assumes binary categories (0 =“FALSE”, 1 = “TRUE”)
- Reference is gold standard, Response from coder
- Contingency Matrix

	Resp=1	Resp=0
Ref=1	TP	FN
Ref=0	FP	TN

- Sensitivity = θ_1 = TP/(TP+FN) = Recall
  - Accuracy on 1 (true) items
- Specificity = θ_0 = TN/(TN+FP) $\neq$ Precision = TP/(TP+FP)
  - Accuracy on 0 (false) items

Gold-Standard Estimate of Accuracies

- Choose random set of positive (category 1) examples
- Choose random set of negative (category 0) examples
- Does not need to be balanced according to prevalence
- Have annotator label the subsets
- Use agreement on negatives for specificity θ_0 [MLE]
- Use agreement on positives for sensitivity θ_1 [MLE]
- Again, more data means more accurate estimates

Generative Labeling Model

- Item i's category $c_i \in \{0, 1\}$
- Coder j's specificity $\theta_{0,j} \in [0, 1]$; sensitivity $\theta_{1,j} \in [0, 1]$
- Coder j's label for item i: $x_{i,j} \in \{0, 1\}$
- If category $c_i = 1$,
  - $\Pr(x_{i,j} = 1) = \theta_{1,j}$ [correctly labeled]
  - $\Pr(x_{i,j} = 0) = 1 - \theta_{1,j}$
- If category $c_i = 0$,
  - $\Pr(x_{i,j} = 1) = 1 - \theta_{0,j}$
  - $\Pr(x_{i,j} = 0) = \theta_{0,j}$ [correctly labeled]
- $\Pr(x_{i,j} = 1 | c, \theta) = c_i \theta_{1,j} + (1 - c_i)(1 - \theta_{0,j})$

Calculating Category Probabilities

- Given prevalence π, specificities θ_0, sensitivities θ_1, and annotations x
- Bayes's Rule

$$\begin{aligned} p(a|b) &= p(b|a)\, p(a)/p(b) \\ &\propto p(b|a)\, p(a) \end{aligned}$$

- Applied to Category Probabilities

$$\begin{aligned} p(c_i|x_i,\theta,\pi) &\propto p(x_i|c_i,\theta,\pi)\, p(c_i|\theta,\pi) \\ &= p(x_i|c_i,\theta)\, p(c_i|\pi) \\ &= p(c_i|\pi)\ \textstyle\prod_{j=1}^{J} p(x_{i,j}|c_i,\theta) \end{aligned}$$

Calculating Cat Probabilities: Example

- Prevalence: $\pi = 0.2$
- Specificities: $\theta_{0,1} = 0.60$; $\theta_{0,2} = 0.70$; $\theta_{0,3} = 0.80$
- Sensitivities: $\theta_{1,1} = 0.75$; $\theta_{1,2} = 0.65$; $\theta_{1,3} = 0.90$
- Annotations for item i: $x_{i,1} = 1,\ x_{i,2} = 1,\ x_{i,3} = 0$

$$\begin{aligned}\Pr(c_i = 1|\theta, x_i) &\propto \pi \Pr(x_i = \langle 1, 1, 0\rangle|\theta, c_i = 1) \\ &= 0.2 \cdot 0.75 \cdot 0.65 \cdot (1 - 0.90) = 0.00975\end{aligned}$$

$$\begin{aligned}\Pr(c_i = 0|\theta, x_i) &\propto (1 - \pi) \Pr(x_i = \langle 1, 1, 0\rangle|\theta, c_i = 0) \\ &= (1 - 0.2) \cdot (1 - 0.6) \cdot (1 - 0.7) \cdot 0.8 = 0.0768\end{aligned}$$

$$\Pr(c_i = 1|\theta, x_i) = \frac{0.00975}{0.00975 + 0.0768} = 0.1126516$$

Bayesian Estimates

Example

Estimates Everything

- What if you don't have a gold standard?
- We can estimate everything from annotations
  - True category labels
  - Prevalence
  - Annotator sensitivities and specificities
  - Mean sensitivity and specificity for pool of annotators
  - Variability of annotator sensitivities and specificities
  - (Individual item labeling difficulty)

Analogy to Epidemiology and Testing

- Commonly used models for epidemiology
  - Tests (e.g. blood, saliva, exam, x-ray, MRI, biopsy) like annotators
  - Prevalence of disease in population
  - Diagnosis of individual patient like item labeling
- Commonly used models in educational testing
  - Annotators like test takers
  - Items like test questions
  - Accuracies like test scores; error patterns are confusions
  - Interesed in difficulty and discriminativeness of questions

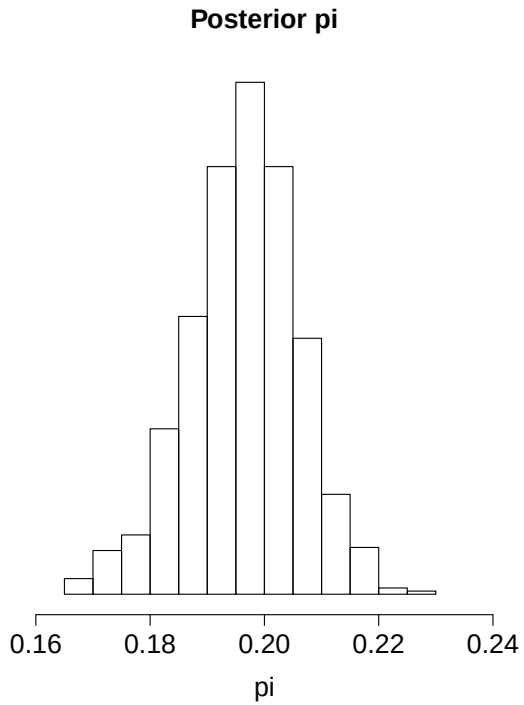
Five Dentists Diagnosing Caries

Dentists	*Count*	*Dentists*	*Count*	*Dentists*	*Count*
00000	1880	10000	22	00001	789
10001	26	00010	43	10010	6
00011	75	10011	14	00100	23
10100	1	00101	63	10101	20
00110	8	10110	2	00111	22
10111	17	01000	188	11000	2
01001	191	11001	20	01010	17
11010	6	01011	67	11011	27
01100	15	11100	3	01101	85
11101	72	01110	8	11110	1
01111	56	11111	100		

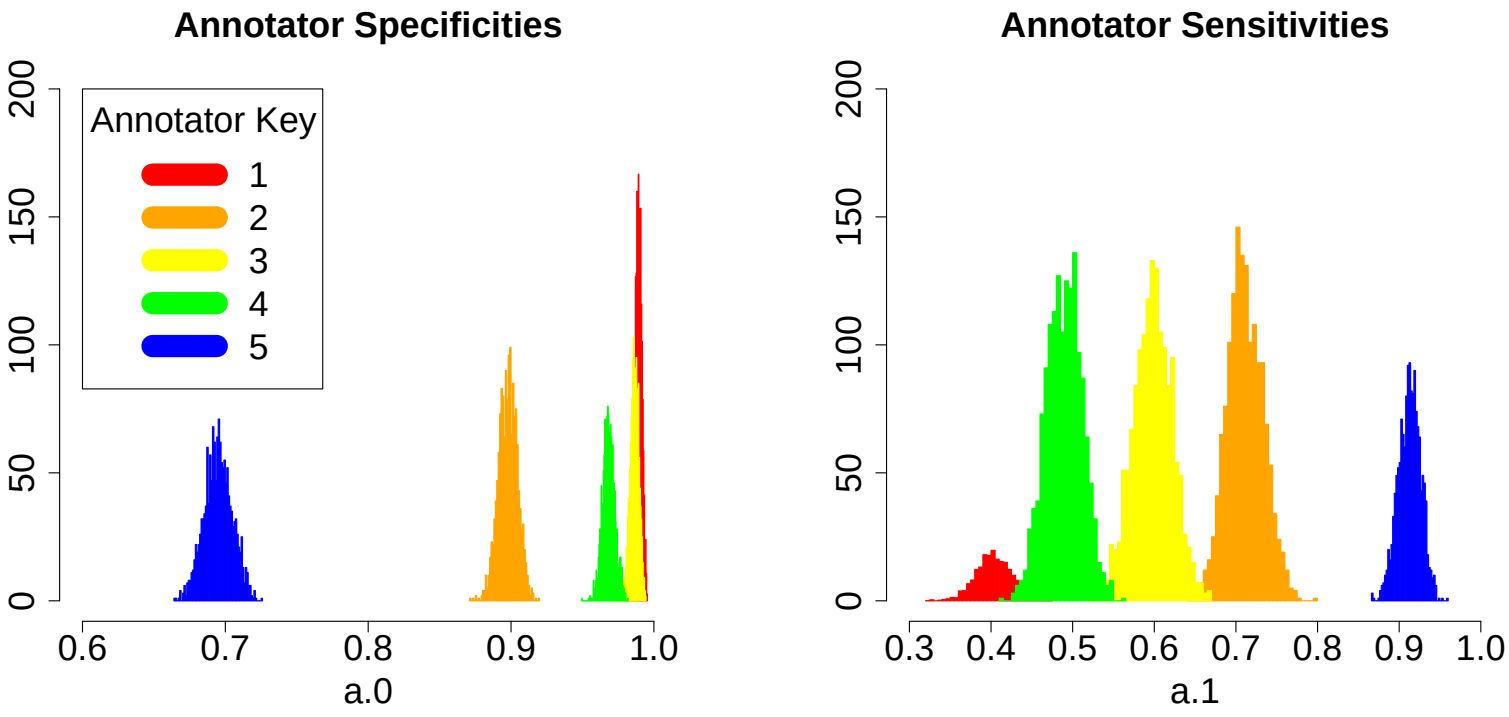
- Caries is a type of tooth pitting preceding a cavity
- Can imagine it's a binary NLP tagging task

Posterior Prevalence of Caries π

- Histogram of Gibbs samples approximates posterior
- 95% interval (0.176, 0.215); Bayesian estimate 0.196
- Consensus estimate (all 1s) 0.026; Majority estimate (≥ 3 1s), 0.13

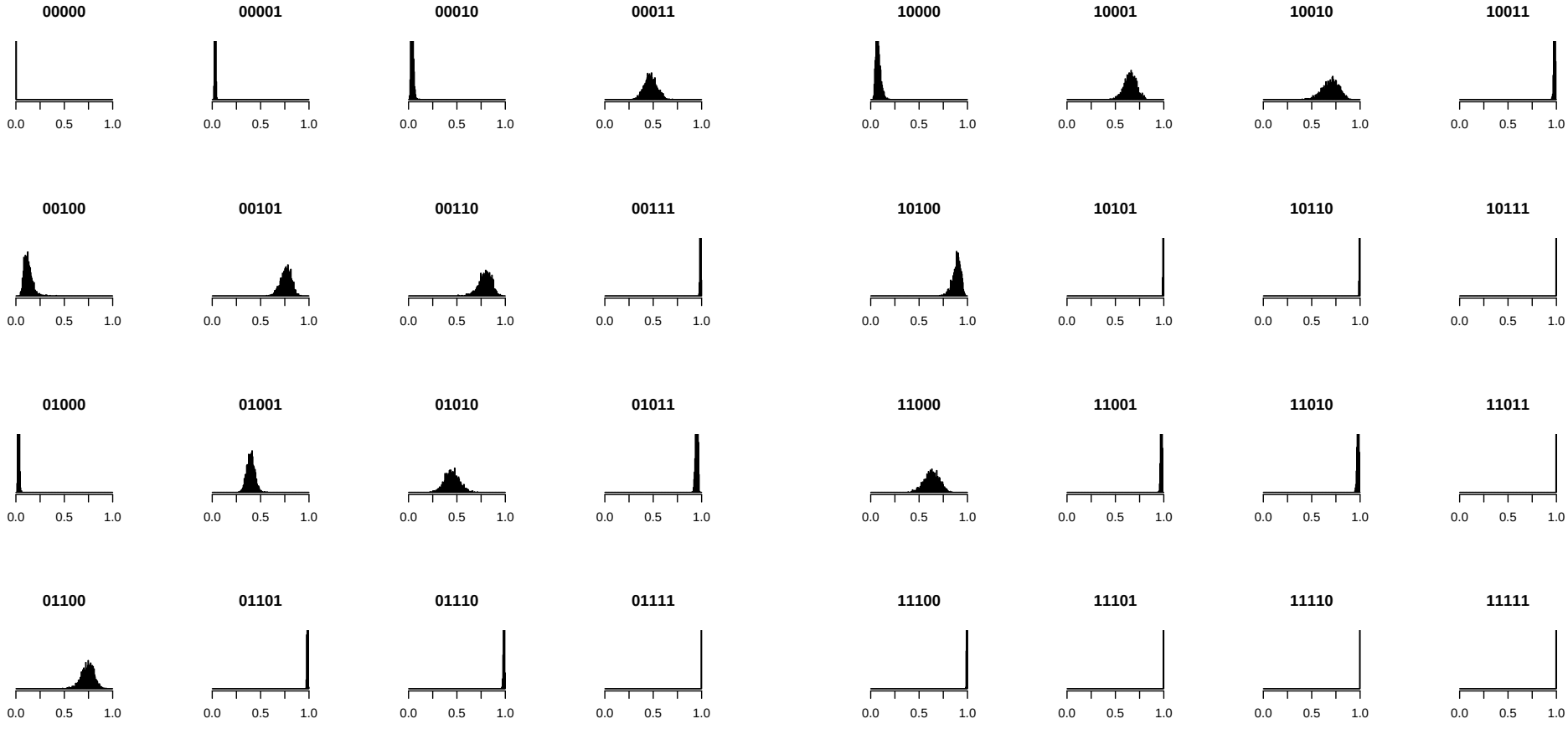


Posteriors for Dentist Accuracies



- Posterior densities useful for downstream inference
- Mitigates overcertainty of point estimates

Posteriors for Dentistry Data Items



- Accuracy adjustment results are very different than simple vote

Marginal Evaluation

- Common evaluation in epidemiology uses χ^2 on marginals

Positive Tests	*Frequency*	*Posterior Quantiles* .025	.5	.975
0	1880	1818	1877	1935
1	1065	1029	1068	1117
2	404	385	408	434
3	247	206	227	248
4	173	175	193	212
5	100	80	93	109

- Simpler models (all accuracies equal) are underdispersed (not enough all-0 or all-1 results)
- Better marginal eval is over all items (e.g. 00100, 01101, . . .)
- Accounting for item difficulty provides even tighter fit

Applications

Is the Truth Out There?

- Or are the "gold standards" just fool's gold?
- Evaluate uncertainty in item category with $\Pr(c_i = 1)$
- Do all items even have true categories?
  - Coding standard may be vague (e.g. "Mars" as location [MUC-6])
  - Distinguishing author/speaker intent from interpretation
- Items often don't have clear interpretation
  - Some items hard to distinguish categorically (esp. metonymy)
  - e.g. "New York" as team or location or political entity
  - e.g. "p53" as gene/protein, wild/mutant, human/mouse

Evaluating Annotators

- Not all annotators have the same accuracy
- Ideally, annotators have high sensitivity and specificity
- Ideally annotators are unbiased
  - Bias indicated by sensitivity $>>$ specificity or vice-versa
- Provide feedback to help annotators improve
- Filter annotators to contribute to coding
- Deciding how many annotators needed for an item

Evaluating “Gold” Labels

- Get probabilities $\Pr(c_i = 1)$
- Middling probabilities mean annotators uncertain
- Find items for which annotators having difficulty
  - Refine coding standard by adjudicating examples
- (Alternative to explicitly modeling item difficulty)

Speed versus Accuracy

- Assume all we have goal for “gold standard” accuracy
- May be beneficial to trade accuracy for speed
  - e.g. 150 items at 80% accuracy vs. 100 items at 90%
  - Former may be better with voting with adjustment for accuracies
- Less-than-perfect gold standard acceptable for some tasks
  - Many machine learning procedures robust to noise
  - More problematic for evaluating “state of the art”

New Items versus New Labels

- Evaluate whether to generate
  - A new label for an uncertainly labeled item, or
  - a new label for currently unlabeled item
  - (Sheng, Provost and Ipeirotis 2009)
- Choose which annotator to label item
  - Can measure expected gain in certainty given annotator accuracy
  - Like active learning, only for annotators rather than items

Evaluating Coding Standard Difficulty

- Replacement for κ with predictive power for new annotators
- Allows inference on correctness of gold standard
- Sensitivity and Specificity priors (α, β) model:
  - Mean annotator accuracy
  - Annotator variation
  - Annotator bias
- Low mean accuracy indicates a problematic coding standard

Probabilistic Training and Testing

- Use probabilistic item posteriors for training
  - Easy to generalize most probabilistic models
  - e.g. naive Bayes or HMMs: proportional train (as for EM)
  - e.g. logistic regression or CRFs: modify log loss
  - Generalize arbitrary model with posterior samples (e.g. SVMs)
- Use probabilistic item posteriors for testing
  - Penalizes overconfidence of models on uncertain items
  - Easy generalization with log loss evaluation
  - Not so clear with first-best accuracy or F-measure
- Demonstrated theoretical effectiveness (Smyth 1995)

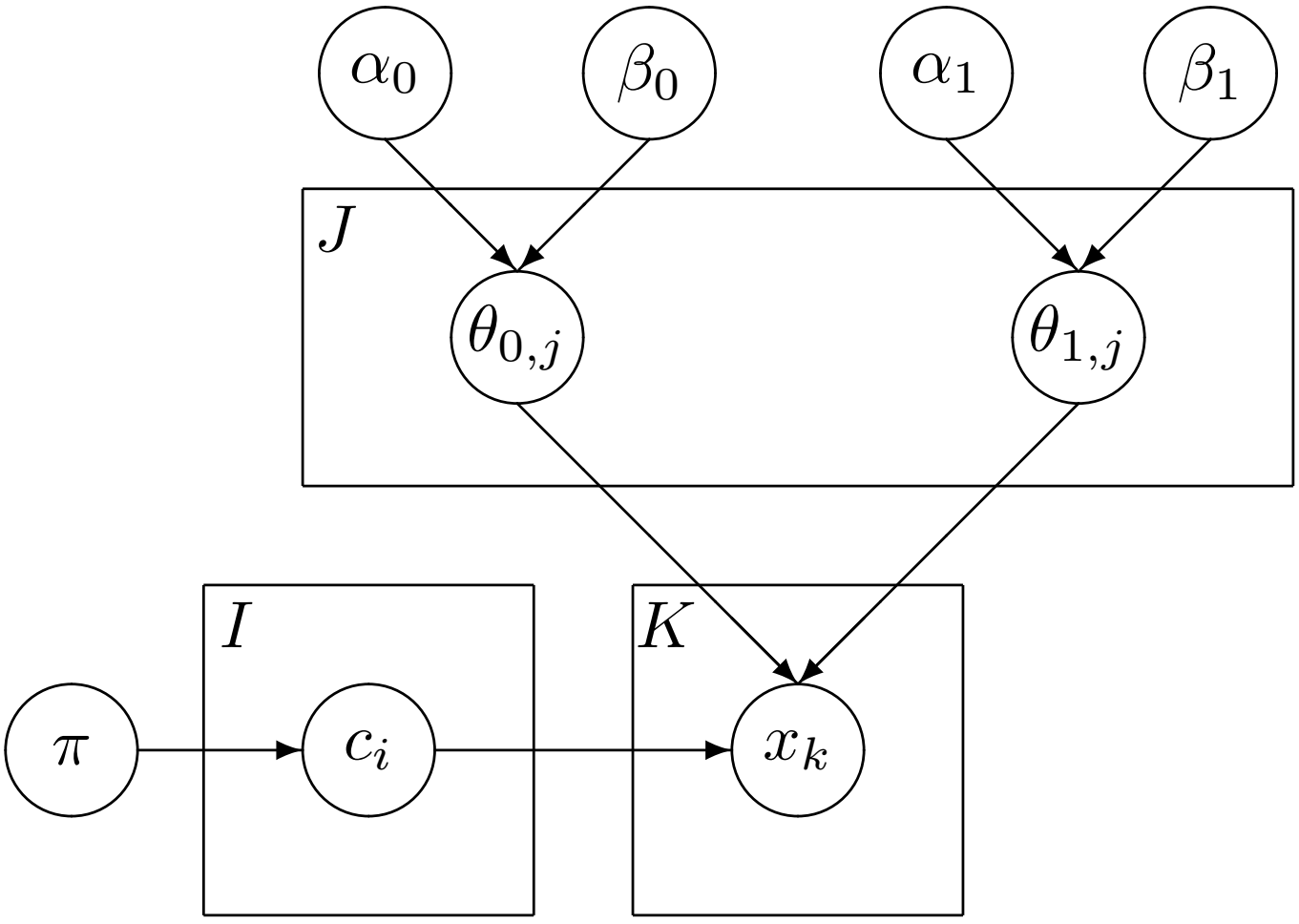
Bayesian κ Estimates

- Given estimated sensitivity, specificity and prevalence:
  - Calculate expected κ for two annotators
  - Don't even need to annotate common items
  - Calculate expected κ for two new annotators
  - Calcluate confidence/posterior uncertainty of κ
  - May formulate hypothesis tests
  - e.g. κ for given standard above 0.8
  - e.g. κ for coding standard 1 higher than for standard 2
- Always a good idea to measure posterior uncertainty
- May estimate Bayesian posteriors (or frequentist confidence intervals) without annotation model

Hierarchical Bayesian Annotation Model

(Carpenter 2008)

Generative Annotation Model Sketch



Generative Model of Annotation Process

- Models all random variables given constant (hyper)priors
- Annotators don't all label all items
- Label x_k by annotator i_k for item j_k

$$
\begin{aligned}
\pi &\sim \mathsf{Beta}(1,1) = \mathsf{Unif}([0,1]) \\
c_i &\sim \mathsf{Bernoulli}(\pi) \\
\theta_{0,j} &\sim \mathsf{Beta}(\alpha_0, \beta_0) \\
\theta_{1,j} &\sim \mathsf{Beta}(\alpha_1, \beta_1) \\
x_k &\sim \mathsf{Bernoulli}(c_{i_k}\theta_{1,j_k} + (1 - c_{i_k})(1 - \theta_{0,j_k}))
\end{aligned}
$$

- Same annotation model as before for labels x given accuracies θ and category c
- Additionally model generation of prevalence π, categories c, and accuracies θ

Hierarchical Generation of Hyperpriors

- Models annotator population: mean accuracies and variability
- Infers appropriate smoothing for low counts
- Prior mean accuracy: $\alpha/(\alpha+\beta)$
- Prior mean scale (inverse variability): $(\alpha+\beta)$

$$
\begin{aligned}
\alpha_0/(\alpha_0+\beta_0) &\sim \text{Beta}(1,1) \\
\alpha_0+\beta_0 &\sim \text{Pareto}(1.5) \\
\alpha_1/(\alpha_1+\beta_1) &\sim \text{Beta}(1,1) \\
\alpha_1+\beta_1 &\sim \text{Pareto}(1.5)
\end{aligned}
$$

- $\text{Beta}(1,1)$ uniform prior on mean accuracies $\alpha/(\alpha+\beta) \in [0,1]$
- $\text{Pareto}(x|1.5) \propto x^{-2.5}$ a diffuse prior for scales $\alpha+\beta \in [0,\infty)$

Sampling Notation Defines Joint Density

$$
\begin{aligned}
&p(c, x, \theta_0, \theta_1, \pi, \alpha_0, \beta_0, \alpha_1, \beta_1) \\
&\quad = \textstyle\prod_{i=1}^{I} \mathsf{Bern}(c_i|\pi) \\
&\qquad \times \textstyle\prod_{k=1}^{K} \mathsf{Bern}(x_k|c_{i_k}\theta_{1,j_k} + (1 - c_{i_k})(1 - \theta_{0,j_k})) \\
&\qquad \times \textstyle\prod_{j=1}^{J} \mathsf{Beta}(\theta_{0,j}|\alpha_0, \beta_0) \\
&\qquad \times \textstyle\prod_{j=1}^{J} \mathsf{Beta}(\theta_{1,j}|\alpha_1, \beta_1) \\
&\qquad \times \mathsf{Beta}(\pi|1, 1) \\
&\qquad \times \mathsf{Beta}(\alpha_0/(\alpha_0 + \beta_0)|1, 1) \\
&\qquad \times \mathsf{Beta}(\alpha_1/(\alpha_1 + \beta_1)|1, 1) \\
&\qquad \times \mathsf{Pareto}(\alpha_0 + \beta_0|1.5) \\
&\qquad \times \mathsf{Pareto}(\alpha_1 + \beta_1|1.5)
\end{aligned}
$$

- Marginals: $p(x) = \int p(x, y)\, dy$; Conditionals: $p(y|x) = p(x, y)/p(x)$

Gibbs Sampling

Gibbs Sampling

- General purpose Markov Chain Monte Carlo (MCMC) method
  - States of Markov chain are samples of all variables

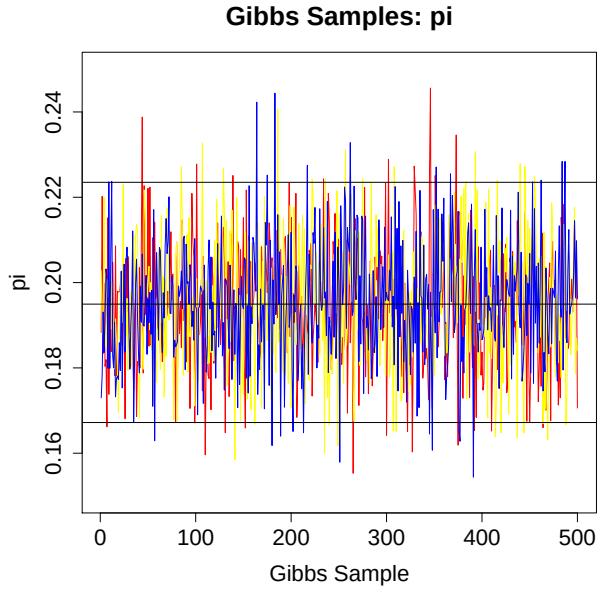
    e.g. $(c^{(n)}, x^{(n)}, \theta_0^{(n)}, \theta_1^{(n)}, \pi^{(n)}, \alpha_0^{(n)}, \beta_0^{(n)}, \alpha_1^{(n)}, \beta_1^{(n)})$
  - It's a continuous Markov process, unlike n-gram LMs or HMMs
- Typically randomly initialize with "reasonable" values
- Next state samples each var given current value of other vars
- Reduces sampling of joint model to conditionals
  - Requires sampler for each variable given all others
  - We explicitly calculated $p(c_i|x, \pi, \theta_0, \theta_1)$ as example

Gibbs Sampling (cont.)

- Works for any model where dependencies form directed acyclic graph
  - Such models called “directed graphical models”
  - Variables with no priors are hyperparameters
  - All otehr variables inferred
- BUGS automatically computes all conditional distributions
- Convergences to stationary process sampling from posterior
  - Typically sample from multiple chains to monitor convergence
  - Typically throw away initial samples before convergence
- Robust compared to Expectation Maximization [EM]

Gibbs Sample Traceplots

- Plots multiple chains overlaid with different colors (3 chains here)



- Want to see this kind of mixing of different chains
- Potential scale reduction statistic $\hat{R}$ characterizes mixing
- BUGS shows traceplots for all vars; Coda package in R calcs $\hat{R}$

Gibbs Samples for Bayesian Estimation

- Bayesian parameter estimate for variable ϕ given N samples $\phi^{(n)}$
- Approximate by averaging over collection of Gibbs samples

$$\begin{aligned}\hat{\phi} &= \mathbb{E}[\phi] \\ &= \int \phi \, p(\phi) \, d\phi \\ &\approx \frac{1}{N} \sum_{n=1}^{N} \phi^{(n)}\end{aligned}$$

- Provides unbiased estimate (equal to expected parameter value)
- Works for any marginal or joint distribution of parameters

Gibbs Samples For Inference

- Samples $\phi^{(n)}$ support plug-in inference
  - E.g. Predictive Posterior Inference

    $p(\tilde{y}|y) = \int p(\tilde{y}|\phi)\, p(\phi|y)\, d\phi \approx \frac{1}{N} \sum_{n=1}^{N} p(\tilde{y}|\phi^{(n)})$

  - E.g. (Multiple) Variable Comparisons

    $\Pr(\theta_{0,j} > \theta_{0,j'}) \approx \frac{1}{N} \sum_{n=1}^{N} \mathrm{I}(\theta_{0,j}^{(n)} > \theta_{0,j'}^{(n)})$

    $\Pr(j \text{ best specificity}) \approx \frac{1}{N} \sum_{n=1}^{N} \prod_{j'=1}^{J} \mathrm{I}(\theta_{0,j}^{(n)} \geq \theta_{0,j'}^{(n)})$

- Latter statistic can be used to compare systems (the ‘E’ in “LREC”)
- More samples provide more accurate approximations
- Plug-in estimates like (frequentist) bootstrap estimates

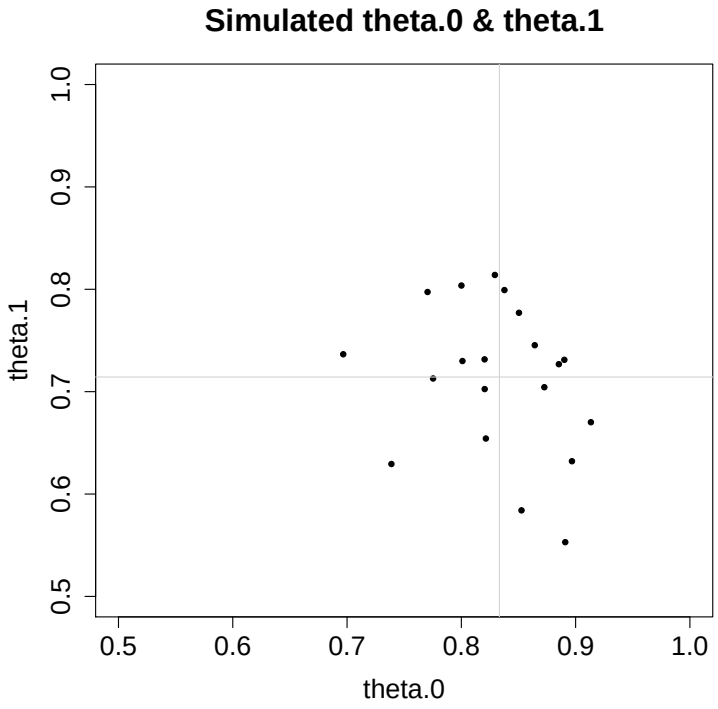
Simulation Study

Simulated Data Tests Estimators

- Simulate data (with reasonable model settings)
- Test sampler's ability to fit
- Simulation Parameter Values
  - $J = 20$ annotators, $I = 1000$ items
  - prevalence $\pi = 0.2$
  - specificity prior $(\alpha_0, \beta_0) = (40, 8)$ (83% accurate, medium var)
  - sensitivity prior $(\alpha_1, \beta_1) = (20, 8)$ (72% accurate, high var)
  - specificities θ_1 generated randomly given α_1, β_1
  - sensitivities θ_1 generated randomly given α_1, β_1
  - categories c generated randomly given π
  - annotations x generated randomly given θ_0, θ_1, c
  - 50% missing annotations removed randomly

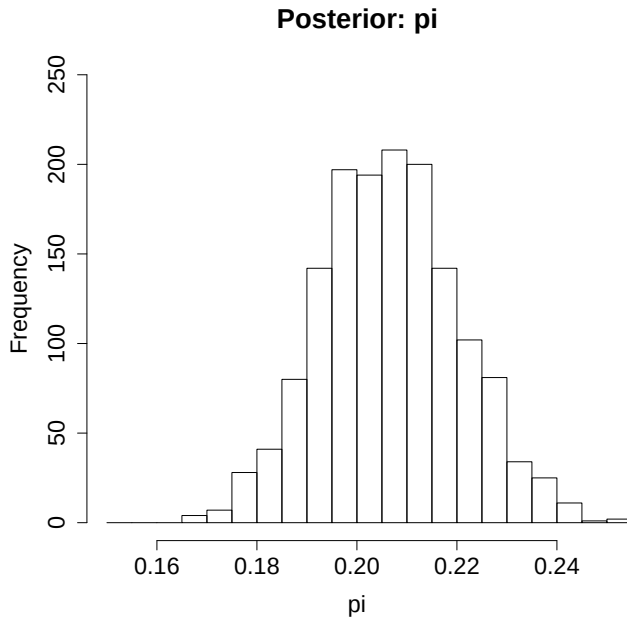
Simulated Sensitivities / Specificities

- Crosshairs at prior mean
- Realistic simulation compared to (estimated) real data

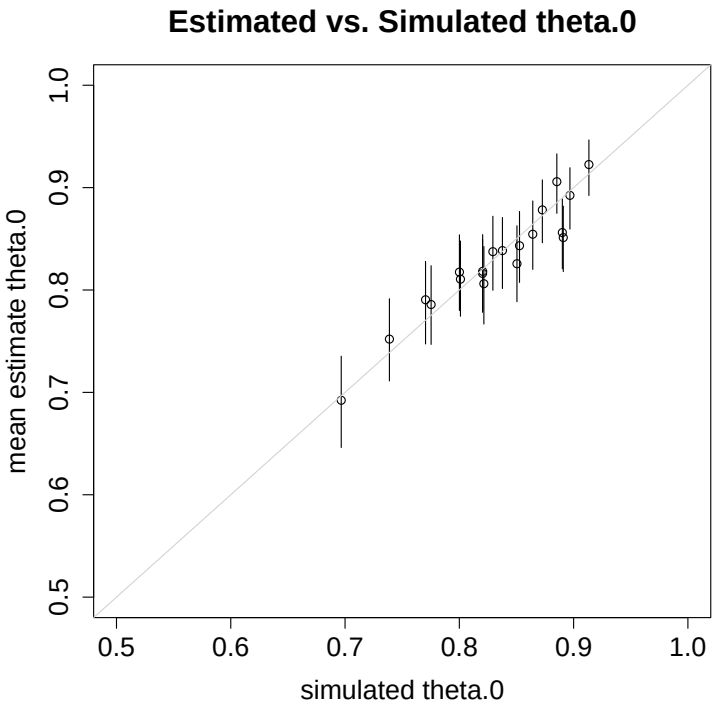


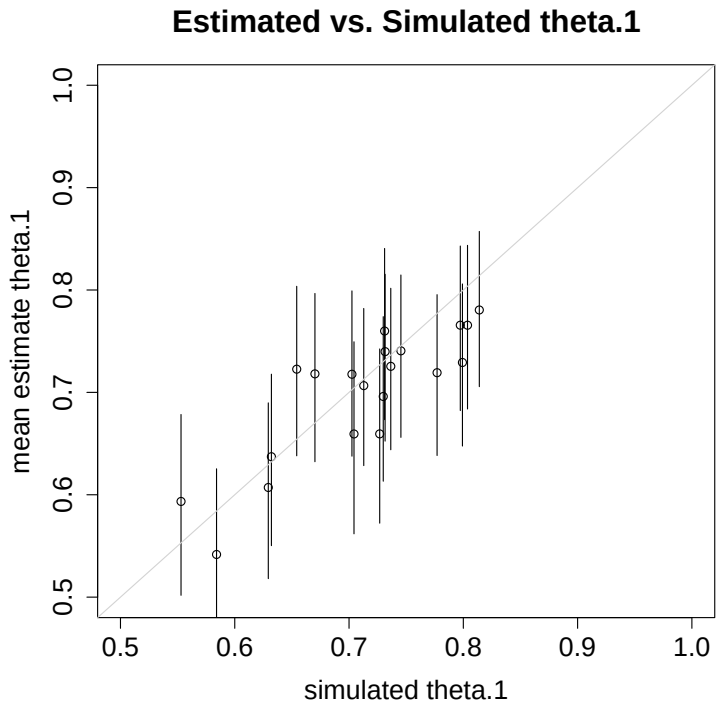
Prevalence Estimate

- Estimand of interest in sentiment (or epidemiology)
- Simulated with prevalence $\pi = 0.2$; sample prevalence 0.21
- Estimates match samples; more data produces tighter estimates
- Histogram of posterior Gibbs samples:



Sensitivity / Specificity Estimates

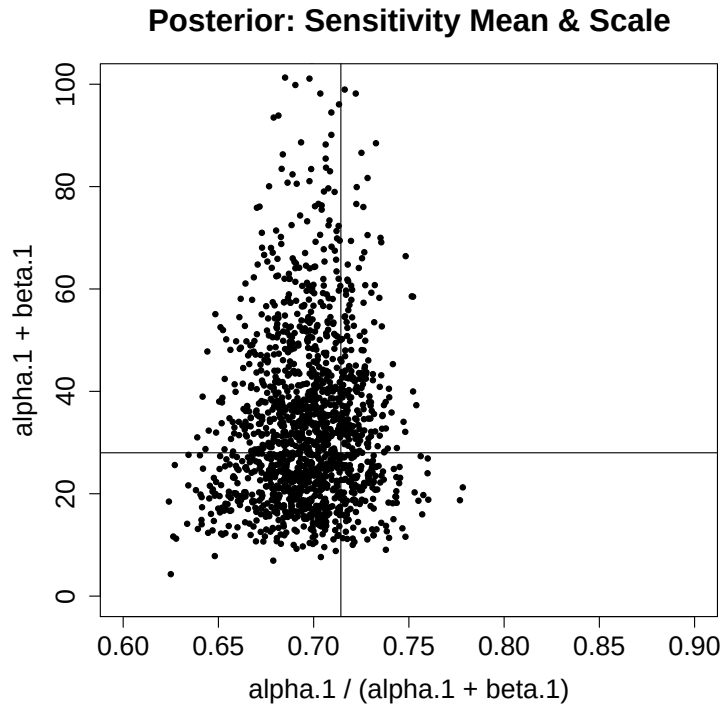


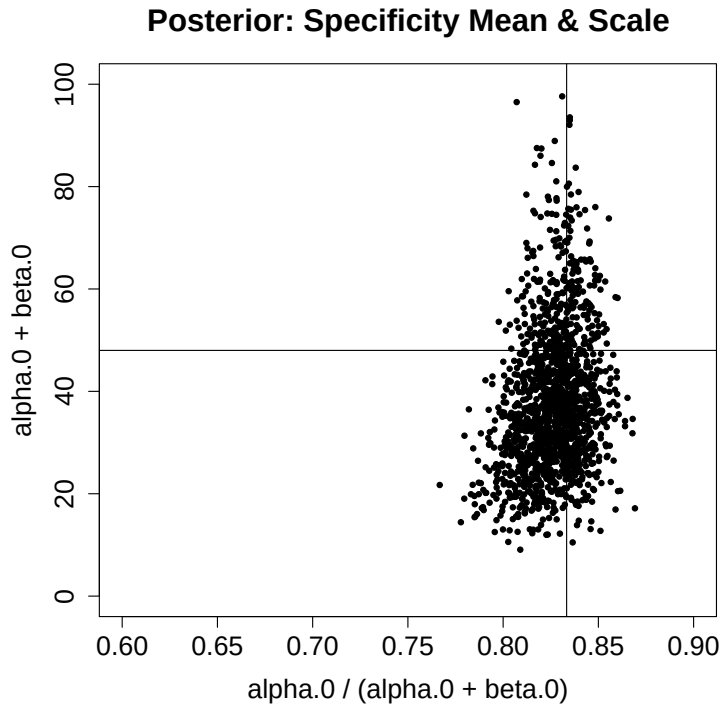


- Posterior mean and 95% intervals
- Diagonal is perfect estimation
- More uncertainty for sensitivity (more data w. $\pi = 0.2$)

Sens / Spec Hyperprior Estimates

- Posterior samples $(\alpha^{(n)}, \beta^{(n)})$ scatterplot
- Cross-hairs at simulated values; estimates at sample averages





- Observe typical skew to high scale (low variance)
- More variance on sensitivity (lower counts)

Another Mechanical Turk Example

(Snow, O'Connor, Jurafsky and Ng 2008)

Case 3: RTE-1

- Examples and gold standard by (Dagan, Glickman and Magnini 2006)
- 800 Items (400 true, 400 false in gold standard)
- Examples
  - *ID:* 56 *Gold Label:* TRUE
    Text: Euro-Scandinavian media cheer Denmark v Sweden draw.
    Hypothesis: Denmark and Sweden tie.

  - *ID:* 77 *Gold Label:* FALSE
    Text: Clinton's new book is not big seller here.
    Hypotheis: Clinton's book is a big seller.

RTE-1 Gold-Standard Procedure

- Each item labeled by two annotators
- Prevalence balanced at $\pi = 0.5$ by design
- Censoring Data
  - Censored 20% of data with disagreements
  - Censored another 13% authors found “questionable”
  - Censoring overestimates certainty and accuracy of evaluated systems on real data

RTE-1 Inter-Annotator Agreement

- Inter-annotator agreement was 80%
- Chance agreement = $0.5^2 + 0.5^2 = 0.5$
- $\kappa = \frac{0.8 - 0.5}{1 - 0.5} = 0.6$
- Assuming 2 annotators at 80% accuracy,

  expect 4% agreement on wrong label:

  $(1 - 0.8) \times (1 - 0.8) = 0.04$

Turker Annotations for RTE-1

- Collected by Dolores Labs
- Analyzed by Snow et al. in $EMNLP$ paper
- They also recreated 4 other NLP datasets:
  - word sense (multinomial)
  - sentiment (multi-faceted scalar 1–100)
  - temporal ordering (binary)
  - word similarity (ordinal 1–10)
- 2 items/task, 10 Turkers per item, 164 Turkers total
- All five tasks completed in a few days
- All five tasks cost under US$100

Turker Instructions for RTE-1

- Instructions

  Please state whether the second sentence (the Hypothesis) is implied by the information in first sentence (the Text), i.e., please state whether the Hypothesis can be determined to be true given that the Text is true.

  Assume that you do not know anything about the situation except what the Text itself says.

  Also, note that every part of the Hypothesis must be implied by the Text in order for it to be true.

- Plus, 2 true and 2 false examples

(Munged) Turker Data for RTE-1

	Item	Coder	Label		k	i	j	x
1	i[1]	j[1]	x[1]		1	1	1	1
2	i[2]	j[2]	x[2]		2	1	2	1
3	i[3]	j[3]	x[3]		3	1	3	1
4	i[4]	j[4]	x[4]		4	1	4	0
509	i[509]	j[509]	x[509]		509	51	22	0
510	i[510]	j[510]	x[510]		510	51	10	1
511	i[511]	j[511]	x[511]		511	52	4	1
512	i[512]	j[512]	x[512]		512	52	1	1
8000	i[8000]	j[8000]	x[8000]		8000	800	144	1

Gold-Standard Estimation (Again)

- Snow et al. used published gold standard as gold standard
- Inferred categories agreed closely with gold standard
- Snow et al. showed 3–5 Turkers as good as experts in most tasks
- Ten Turkers better than pair of “experts”
  - Turkers better matched coding standard on disagreements
  - Lots of random (spam) annotations from Turkers
  - Filtering out bad Turkers would have better ratio

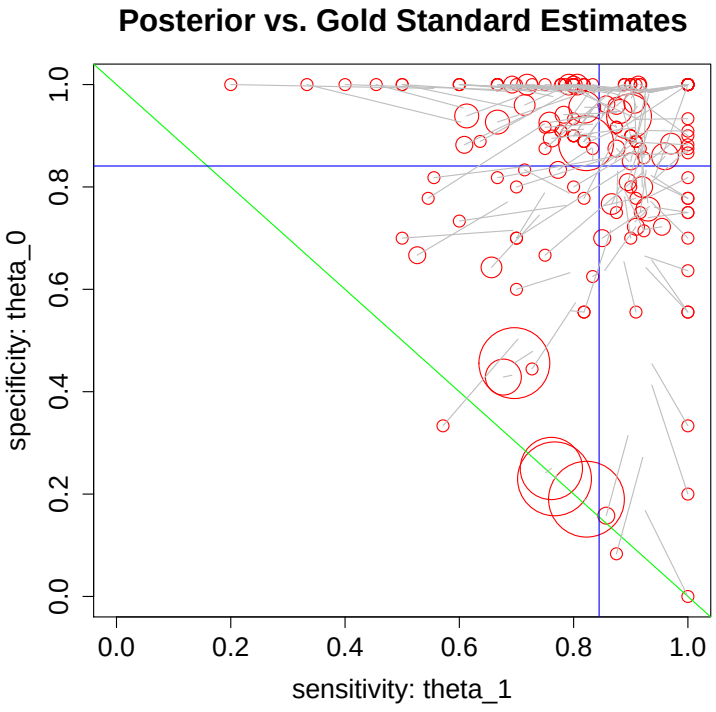
BUGS Code

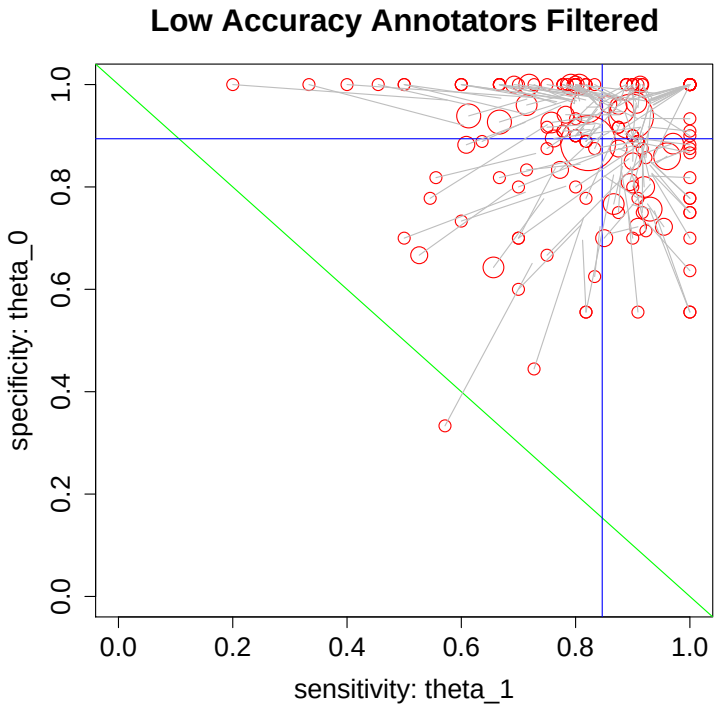
```
model {
  pi ~ dbeta(1,1)
  for (i in 1:I) {
    c[i] ~ dbern(pi)
  }
  for (j in 1:J) {
    theta.0[j] ~ dbeta(alpha.0,beta.0) I(.4,.99)
    theta.1[j] ~ dbeta(alpha.1,beta.1) I(.4,.99)
  }
  for (k in 1:K) {
    bern[k] <- c[ii[k]] * theta.1[jj[k]]
               + (1 - c[ii[k]]) * (1 - theta.0[jj[k]])
    xx[k] ~ dbern(bern[k])
  }
  acc.0 ~ dbeta(1,1)
  scale.0 ~ dpar(1.5,1)  I(1,100)
  alpha.0 <- acc.0 * scale.0
  beta.0 <- (1-acc.0) * scale.0
  acc.1 ~ dbeta(1,1)
  scale.1 ~ dpar(1.5,1) I(1,100)
  alpha.1 <- acc.1 * scale.1;
  beta.1 <- (1-acc.1) * scale.1
}
```

Calling BUGS from R

```
library("R2WinBUGS")

data <- list("I","J","K","xx","ii","jj")

parameters <- c("c", "pi","theta.0","theta.1",
                "alpha.0", "beta.0", "acc.0", "scale.0",
                "alpha.1", "beta.1", "acc.1", "scale.1")

inits <- function() {
  list(pi=runif(1,0.7,0.8),
       c=rbinom(I,1,0.5),
       acc.0 <- runif(1,0.9,0.9),
       scale.0 <- runif(1,5,5),
       acc.1 <- runif(1,0.9,0.9),
       scale.1 <- runif(1,5,5),
       theta.0=runif(J,0.9,0.9),
       theta.1=runif(J,0.9,0.9))  }

anno <- bugs(data, inits, parameters,
             "c:/carp/devguard/sandbox/hierAnno/trunk/R/bugs/beta-binomial-anno.bug",
              n.chains=3, n.iter=500, n.thin=5,
              bugs.directory="c:\\WinBUGS\\WinBUGS14")
```

Estimated vs. “Gold” Accuracies



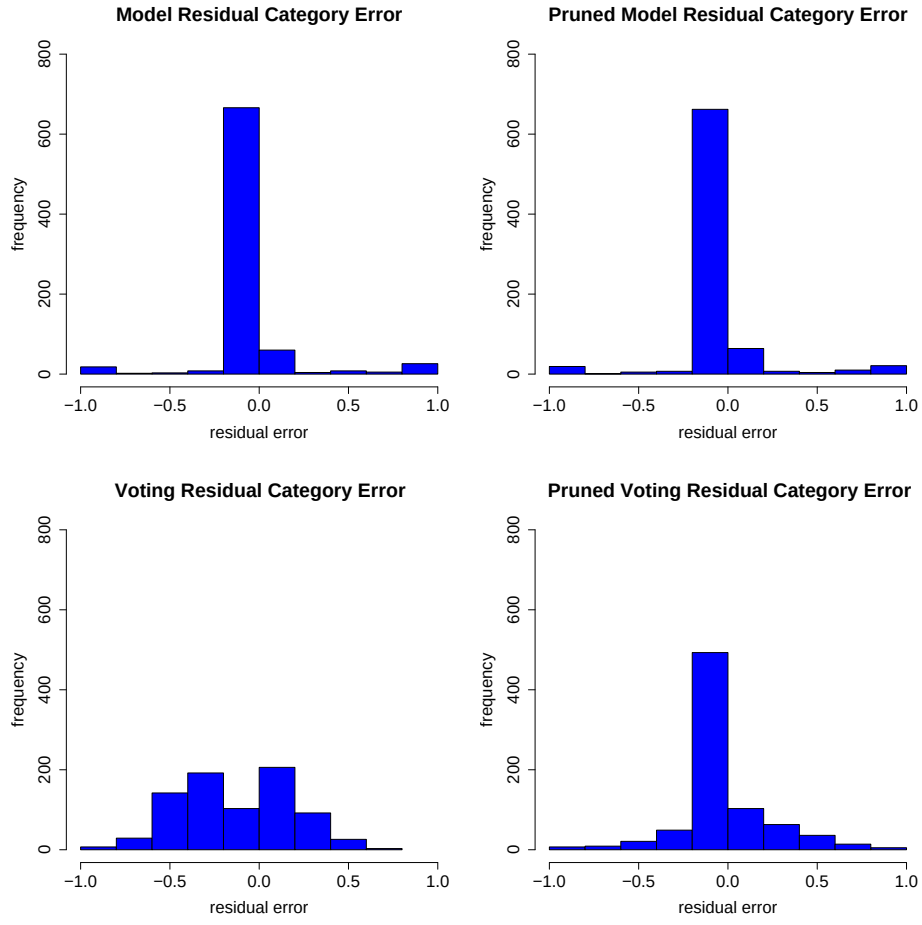


- Diagonal green at chance (below is adversarial)
- Blue lines at estimated prior means
- Circle area is items annotated, center at “gold standard” accuracy, lines to estimated accuracy (note pull to prior)

Annotator Pool Estimates

- Gold-standard balanced (50% prevalence)
- Posterior 95% intervals
  - Prevalence (.45,.52)
  - Specificity (.81,.87)
  - Sensitivity (.82,.87)
  - (Expect balanced sensitivity/specificity due to symmetry of task)
- Dealing with bad Turkers
  - 39% of annotators no better than chance
  - more than 50% of annotations from spammers
  - very little effect on category inference
  - has strong effect on mean and variability of annotators

Residual Cat Errors: $c_i - \Pr(c_i = 1 | \cdots)$



- Most residual errors in gold std (extremes of top graphs), not Turkers
- Pruning bad annotators improves voting more than model estimates

Modeling Item Difficulty

Item Difficulty

- Clear that some items easy and some hard
- Assuming all same leads to suboptimal marginal fit
- Hard to estimate even with 10 annotators/item
  - Posterior intervals too wide for good read on difficulty
  - Fattens posteriors on annotator accuracies
  - Better marginal fits (by χ^2)
- Problem is multiple explanations
  - Good annotator accuracy, but hard items
  - Mediocre annotator accuracy, medium difficulty items
  - Poor annotator accuracy, but easy items

Mixture Item Difficulty

- Assume some items easy in sense all annotators agree
- Simple mixture model over items
- Easy to estimate

- (Espeland and Handelman 1989; Beigman-Klebanov, Beigman, and Diermeier 2008)

Modeling Scalar Item Difficulty

- Assume item difficulties vary on continuous scale
- Logistic Item-Response or Rasch models
- Used in social sciences to model educational testing and voting
- Use logistic scale (maps $(-\infty, \infty)$ to $[0, 1]$)
- α_j: annotator j's bias (ideally 0)
- δ_j: annotator j's discriminativeness (ideally ∞)
- β_i: item i's "location" (true category and difficulty)
- $x_i \sim \text{logit}^{-1}(\delta_j(\alpha_i - \beta_j))$
- Many variants of this model in epidemiology and testing

- (Uebersax and Grove 1993; Qu, Tan and Kutner 1996; Carpenter 2008)

Hierarchical Item Difficulty Model

- Place normal (or other) priors on coefficients,

  e.g. $\beta_i \sim \text{Norm}(0, \sigma^2), \quad \sigma^2 \sim \text{Unif}(0, 100)$
- Priors may be estimated as before; leads to pooling of item difficulties.
- Harder to estimate computationally in BUGS
- Same posterior inferences when converted back to linear scale
- e.g. average annotator accuracies, average item difficulties

Extensions

Extending Coding Types

- Multinomial responses (Dirichlet-multinomial)
- Ordinal responses (ordinal logistic model)
- Scalar responses (continuos responses)

Hierarchical and Multilvel Models

- Assume several coding tasks
  - e.g. multiple part-of-speech corpora
  - e.g. multiple named-entity corpora (see Finkel and Manning 2009)
  - e.g. multiple language newswire categorization
  - e.g. coref corpora in different genres or languages
- Estimate another level of priors
  - e.g. for prevalence
  - e.g. for priors on accuracy priors
- Common approach in social science models
- Even better pooled estimates if corpora are similar
- Multilevel models allow cross-cutting “hierarchies”

Semi-Supervision

- Easy to add in supervised cases with Bayesian models
  - Gibbs sampling skips sampling for supervised cases
- May go half way by mixing in “gold standard” annotators
  - e.g. fixed values from gold standard, or
  - e.g. fixed high, but non-100% accuracies, or
  - e.g. stronger high accuracy prior
- With accurate supervision, improves estimates
  - for prevalence
  - for annotator accuracies
  - for pool of annotators

Multimodal (Mixture) Priors

- Model Mechanical Turk as mixture of spammers and hammers
- This is what the Mechanical Turk data suggests
- May also model covariance of sensitivity/specificity
  - Use multivariate normal or T distribution
  - with covariance matrix
  - Covariance may also be estimated hierarchically (see Lafferty and Blei 2007)

Annotator and Item Random Effects

- May add random effects for annotators
  - amount of annotator training
  - number of items annotated
  - annotator native language
  - annotator field of expertise
  - intern, random undergrad, grad student, task designer
- Also for Items
  - difficulty (already discussed)
  - type of item being annotated
  - frequency of item in a large corpus
  - capitalization in named entity detection
- Use logistic regression with these predictors to model accuracies

Jointly Estimate Model and Annotations

- Can train a model with inferred (probabilistic) gold standard
- May use trained model like another annotator
- May be beneficial to train multiple different annotators

- (Raykar, Yu, Zhao, Jerebko, Florin, Hermosillo-Valadez, Bogoni, and Moy 2009)

Bayesian Estimation and Inference

Bayesian Models

- Observed data: y; Model parameter(s): ϕ
- Likelihood function (or sampling distribution): $p(y|\phi)$
- Prior: $p(\phi)$
- Chain rule: $p(y, \phi) = p(y|\phi)\, p(\phi)$
- Marginal (prior predictive distribution): $p(y) = \int p(y|\phi)\, p(\phi)\, d\phi$
- Posterior: $p(\phi|y)$ calculated via Bayes's rule

$$
\begin{aligned}
p(\phi|y) &= p(y, \phi)/p(y) \\
&= p(y|\phi)\, p(\phi)/p(y) \\
&= p(y|\phi)\, p(\phi)/ \int p(y|\phi')p(\phi')d\phi' \\
&\propto p(y|\phi)\, p(\phi)
\end{aligned}
$$

Point Estimators are "Best" Guesses

- Estimate parameters ϕ given observed data y
- Maximum Likelihood Estimator (ML)

  $\phi^*(y) = \arg\max_\phi p(y|\phi)$

  maximizes probability of observed data given parameters
- Maximum a Posteriori (MAP) Estimate

  $\hat{\phi}(y) = \arg\max_\phi p(\phi|y) = \arg\max_\phi p(y|\phi)\ p(\phi)$

  maximizes probability of parameters given observed data
- If prior is constant, [i.e. $p(\phi) = c$], then $\hat{\phi}(y) = \phi^*(y)$
- Bayesian estimator (given mean square error loss)

  $\bar{\phi}(y) = \mathbb{E}[\phi] = \int \phi\ p(\phi|y)\ d\phi$

  is expected parameter values given observed data
- Bayesian estimates are unbiased by construction
  [i.e. expected estimate is parameter's true value]

Inference

- Observed data y; New data $\tilde{y}$
- Posterior predictive distribution: $p(\tilde{y}|y)$
- Maximum likelihood approximation: $p(\tilde{y}|y) \approx p(\tilde{y}|\phi^*(y))$
- MAP approximation: $p(\tilde{y}|y) \approx p(\tilde{y}|\hat{\phi}(y))$
- Bayesian point approximation: $p(\tilde{y}|y) \approx p(\tilde{y}|\bar{\phi}(y))$
- Bayesian posterior predictive distribution

  $p(\tilde{y}|y) = \int p(\tilde{y}|\phi)\, p(\phi|y)\, d\phi$

  averages over uncertainty in estimate of ϕ [i.e. $p(\phi|y)$]

Bernoulli Distribution (Single Binary Trial)

- Outcome $y \in \{0, 1\}$ [success=1, failure=0]
- Parameter $\theta \in [0, 1]$ is chance of success
- $p(y|\theta) = \text{Bernoulli}(y|\theta) = \theta^y \ (1-\theta)^{1-y} = \begin{cases} \theta & \text{if y = 1} \\ 1-\theta & \text{if y = 0} \end{cases}$
- For N independent trials $y = y_1, \ldots, y_N,$ where $y_n \in \{0, 1\}$,

$$\begin{aligned} p(y|\theta) &= \prod_{n=1}^N p(y_n|\theta) \\ &= \prod_{n=1}^N \theta^{y_n} \ (1-\theta)^{1-y_n} \\ &= \theta^A (1-\theta)^B \end{aligned}$$

  where $A = \sum_{n=1}^N y_n$ and $B = \sum_{n=1}^N (1 - y_n) = N - A$

- $x^0 = 1$ and $x^a x^b = x^{a+b}$

Conjugate Priors

- Given a sampling distribution $p(y|\phi)$
- Given a family of distributions $\mathcal{F}$
- The family $\mathcal{F}$ is conjugate for $p(y|\phi)$ if

  prior $p(\phi) \in \mathcal{F}$ implies posterior $p(\phi|y) \in \mathcal{F}$
- Provides analytic form of posterior (vs. numerical approximation)
- Supports incremental updates
  - Start with prior $p(\phi) \in \mathcal{F}$
  - After data y, have posterior $p(\phi|y) \in \mathcal{F}$
  - Use $p(\phi|y)$ as prior for new data y'
  - New posterior is $p(\phi|y, y') \in \mathcal{F}$
  - i.e. updating with y then y' same as upating for y, y' together
- Not necessary for Bayesian inference

Mean, Mode and Variance for Bernoulli

- Mean and Mode: $\mathbb{E}[\text{Bern}(\theta)] = \text{mode}[\text{Bern}(\theta)] = \theta$
- Variance: $\text{var}[\text{Bern}(\theta)] = \theta\ (1 - \theta)$
- Standard Deviation: $\text{sd}[\text{Bern}(\theta)] = \sqrt{\theta\ (1 - \theta)}$

- For discrete X with N outcomes $x_1, \ldots, x_N$ distributed $p_X(x)$
  - Mode (Max Value): $\text{mode}[X] = \arg\max_{n=1}^{N} p(x_n)$
  - Mean (Average Value): $\mathbb{E}[X] = \sum_{n=1}^{N} p(x_n)\ x_n$
  - Variance: $\text{var}[X] = \mathbb{E}[X - \mathbb{E}[X]] = \sum_{n=1}^{N} p(x_n)\ (x_n - \mathbb{E}[X])^2$
  - Standard Deviation: $\text{sd}[X] = \sqrt{\text{var}[X]}$

Beta Distribution

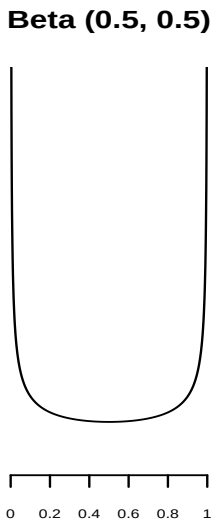
- Outcome $\theta \in [0, 1]$
- Parameters $\alpha, \beta > 0$ [$\alpha - 1$ prior successes; $\beta - 1$ failures]
- Continuous Density Function

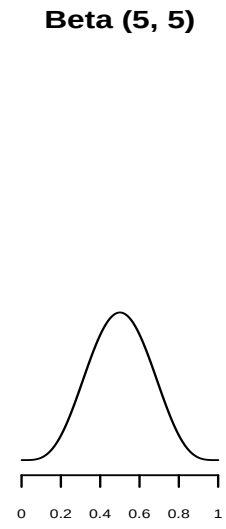
$$\begin{aligned} p(\theta|\alpha, \beta) &= \text{Beta}(\theta|\alpha, \beta) \\ &= \frac{1}{\mathrm{B}(\alpha, \beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1} \\ &\propto \theta^{\alpha-1} (1-\theta)^{\beta-1} \end{aligned}$$

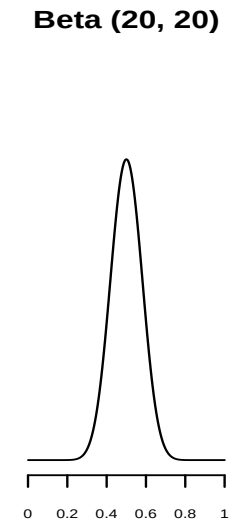
- Continuous densities $p(\theta)$ have $p(\theta) \geq 0$ and $\int p(\theta) d\theta = 1$
- Beta function $\mathrm{B}(\alpha, \beta) = \int_0^1 \theta^{\alpha-1} (1-\theta)^{\beta-1} d\theta = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$
- $\Gamma(x) = \int_0^\infty y^{x-1} \exp(-y) dy$ is continuous generalization of factorial

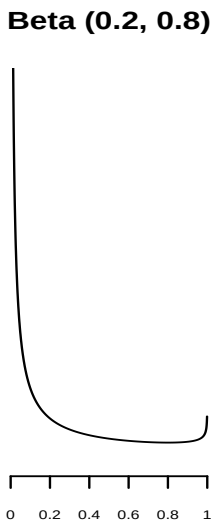
  i.e. $\Gamma(n+1) = n! = n \times (n-1) \times \cdots \times 2 \times 1$ for integer $n \geq 0$

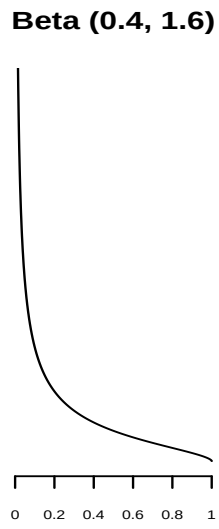
Beta Examples

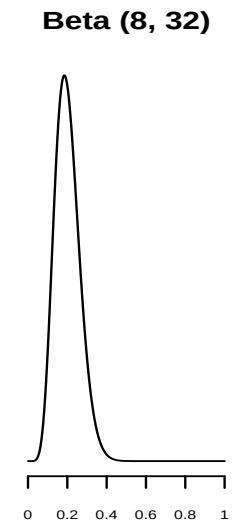












Mean, Mode and Variance for Beta

- Mean: $\mathbb{E}[\text{Beta}(\alpha, \beta)] = \dfrac{\alpha}{\alpha + \beta}$
- Variance: $\text{var}[\text{Beta}(\alpha, \beta)] = \dfrac{\alpha\beta}{(\alpha + \beta)^2 \, (\alpha + \beta + 1)}$
- Mode: $\text{mode}[\text{Beta}(\alpha, \beta)] = \begin{cases} \dfrac{\alpha - 1}{\alpha + \beta - 2} & \text{if } \alpha > 1 \text{ and } \beta > 1 \\ \text{undefined} & \text{otherwise} \end{cases}$

Beta is Conjugate Prior for Bernoulli

- Data is N Bernoulli samples $y = y_1, \ldots, y_N$ for $y_n \in \{0, 1\}$
- Prior $p(\theta) = \text{Beta}(\theta|\alpha, \beta)$
- Likelihood $p(y|\theta) = \prod_{n=1}^{N} \text{Bern}(y_n|\theta) = \theta^A (1-\theta)^B$

  where A is number of successes, B number of failures in y
- Posterior

$$\begin{aligned} p(\theta|y) &\propto p(y|\theta)\, p(\theta) \\ &= \prod_{n=1}^{N} \text{Bern}(y_n|\theta)\, \text{Beta}(\theta|\alpha, \beta) \\ &\propto \theta^A\, (1-\theta)^B\, \theta^{\alpha-1}\, (1-\theta)^{\beta-1} \\ &= \theta^{A+\alpha-1}\, (1-\theta)^{B+\beta-1} \\ &\propto \text{Beta}(A+\alpha, B+\beta) \end{aligned}$$

- i.e. add data counts A and B to prior counts $\alpha - 1$ and $\beta - 1$
- Concrete example of incremental updates – just addition

The End

References

- References
  - `http://lingpipe-blog.com/`
- Contact
  - `carp@alias-i.com`
- R/BUGS (Anon) Subversion Repository

  ```
  svn co https://aliasi.devguard.com/svn/sandbox/hierAnno
  ```