The need for Bayesian inference

Clustering algorithms so far

k-means

Assign observations to closest cluster center

$$z_i \leftarrow \arg\min_j \|\mu_j - x_i\|_2^2$$

Revise cluster centers

$$\mu_j \leftarrow \arg\min_\mu \sum_{i:z_i=j} \|\mu - x_i\|_2^2$$

Iterative hard assignment to max objective

EM for MoG

E-step: estimate cluster responsibilities

$$r_{ik} = \frac{\hat{\pi}_k N(x_i \mid \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^K \hat{\pi}_j N(x_i \mid \hat{\mu}_j, \hat{\Sigma}_j)}$$

M-step: maximize likelihood over parameters

$$\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k \mid \{\hat{r}_{ik}, x_i\}$$

Iterative soft assignment to max objective

What can we do for our bag-of-words models?

Part 1: Clustering model

One topic indicator z_i per document i

All words come from (get scored under) same topic z_i

Distribution on prevalence of topics in corpus $\pi = [\pi_1\ \pi_2 \dots \pi_K]$

Can derive EM algorithm

- Gaussian likelihood of tf-idf vector

$\downarrow$

multinomial likelihood of word counts (m_w successes of word w)

- Result: mixture of multinomial model

Part 2: LDA model

Can derive EM algorithm, but not common (performs poorly)

Typical LDA implementations

Normally LDA is specified as a **Bayesian model**
(otherwise, "probabilistic latent semantic analysis/indexing)

- Account for uncertainty in parameters when making predictions
- Naturally regularizes parameter estimates in contrast to MLE

EM-like algorithm (eg. "variational EM"), or...

Gibbs sampling from 10,000 feet

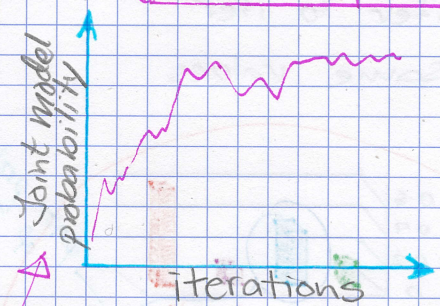
Gibbs sampling

$\boxed{\text{Iterative random hard assignment!}}$

Benefits
- Typically intuitive updates
- Very straightforward to implement

What do we know about this process?

$\boxed{\text{Not an optimization algorithm}}$



Eventually provides "correct" Bayesian estimates...

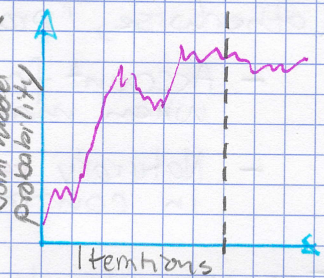
probability of observation given variables/parameters and probability of variables/parameters themselves

What to do with sampling output?

Predictions:
1) Make prediction for each snapshot of randomly assigned variables/parameters (full iteration)
2) Average predictions for final result

What to do with sampling output? (cont'd)

Parameter or assignment estimate
- Look at snapshot of randomly assigned variables/parameters that maximizes "joint model probability"



A standard Gibbs sampler for LDA

Standard Gibbs sampling step

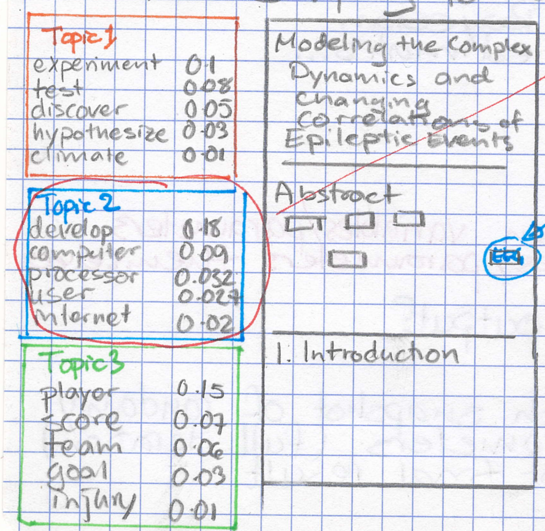
Gibbs sampling algorithm outline

$\boxed{\text{Iterative random hard assignment!}}$

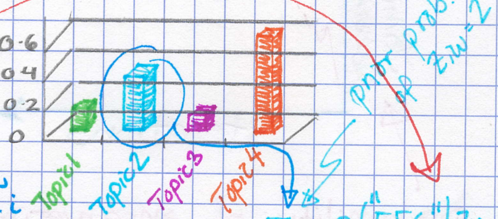
Assignment variables and model parameters treated similarly

Iteratively draw variable/parameter from conditional distribution having fixed:
- all other variables/parameters
  - values randomly selected in previous rounds
  - changes from iter to iter
- observations
  - always the same values

Gibbs sampling for LDA



word w in doc i



prior prob. of $z_{iw}=2$

$$r_{iw2} = \frac{\pi_{i2} \cdot p(\text{"EEG"} \mid z_{iw}=2)}{\sum_{j=1}^{K} \pi_{ij}\, p(\text{"EEG"} \mid z_{iw}=j)}$$

prob. of assigning $z_{iw}=2$

Step 1: Randomly reassign all z_{iw} based on
- doc topic proportions
- topic vocab distributions

Draw randomly from responsibility vector $[r_{iw1}\ r_{iw2}\ \dots\ r_{iwK}]$

Gibbs sampling for LDA (cont'd)

Step 2: Randomly reassign doc topic proportions based on assignments z_{iw} in current doc

Step 3: Repeat for all docs

Step 4: Randomly reassign topic vocab distributions based on assignments z_{iw} in entire corpus

Repeat Steps 1-4 until max iter reached