Modeling process

- Number of words (V), number of topics (K)

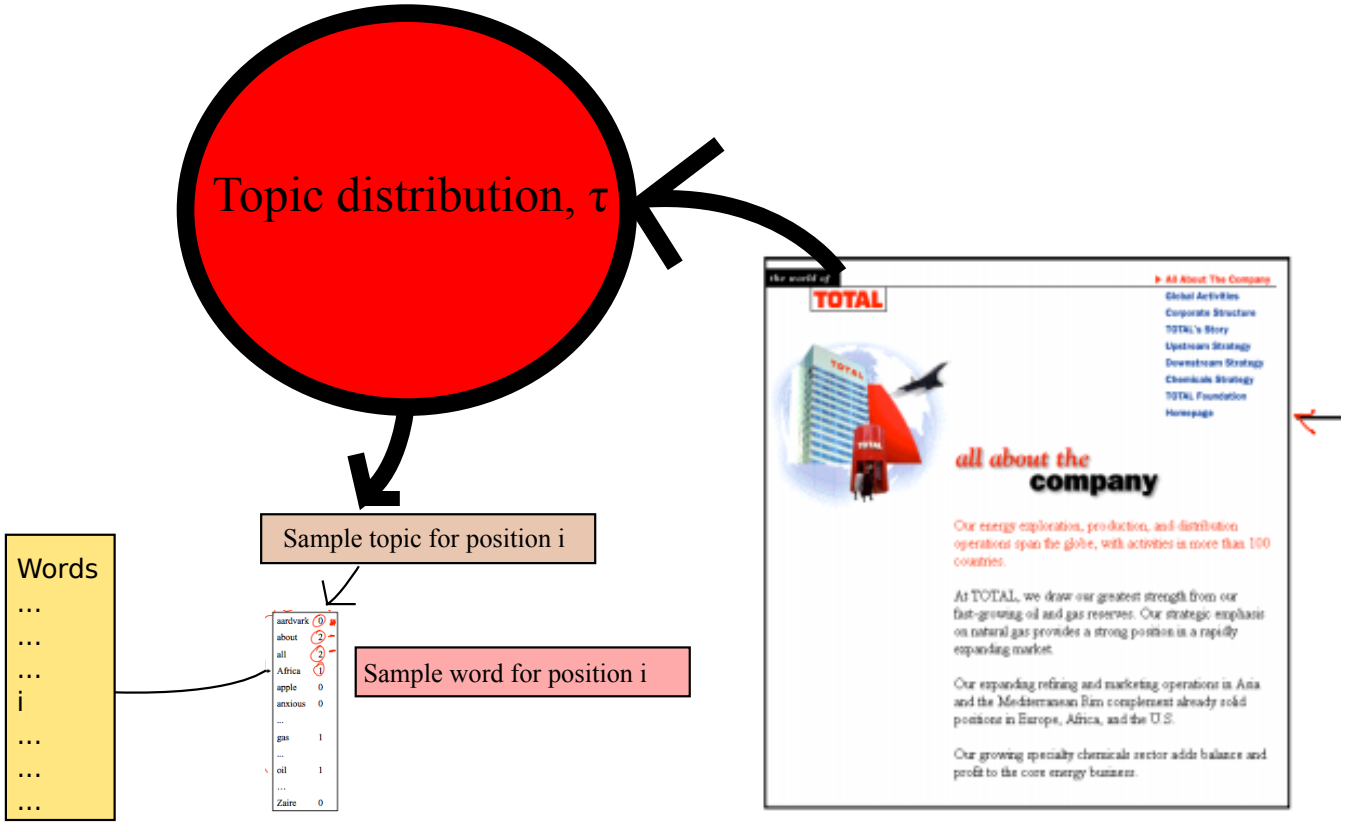

Figure: Admixture model

Task

The learning task is to find the word-topic matrix A with the statistical model above. This is the essence of statistical recovery. They also show how to learn the hyperparameters of the topic distribution when it is a Dirichlet distribution

Word-topic matrix

Kristy's slide goes here

Topic recovery using Bayes' Rule

Mathematical prerequisites

- For any two words w_1 and w_2 with respective topic assignments z_1 and z_2 the elements of the word-topic matrix can be interpreted as

$A_{i,k} = p(w_1 = i \,|z_1 = k)$ (1)
The A matrix gives probability of word in the ith row given a topic k (column)

- Word co-occurences
  - Q $\rightarrow$ measures joint probability of words occuring together
  - $Q_{i,j} = p(w_1 = i, w_2 = j)$ (2)
  - $\bar{Q}_{i,j} = p(w_2 = j | w_1 = i)$ (3)

Topic recovery using Bayes' Rule

Convex Hull I

For an anchor word (row) in the co-ocurrence matrix

$$Q_{s_k,j} = \sum_{k'} p(z_1 = k' | w_1 = s_k) p(w_2 = j | z_1 = k')$$

= 1 because of the anchor word property

$$= p(w_2 = j | z_1 = k) = C_{i,k}$$

For any other row

$$\bar{Q}_{i,j} = \sum_k p(z_1 = k | w_1 = i) p(w_2 = j | z_1 = k)$$

But this is clearly a convex combination of anchor words

$$\bar{Q}_{i,j} = \sum_k C_{i,k} \bar{Q}_{s,k}$$

Topic recovery using Bayes' Rule

Convex Hull II

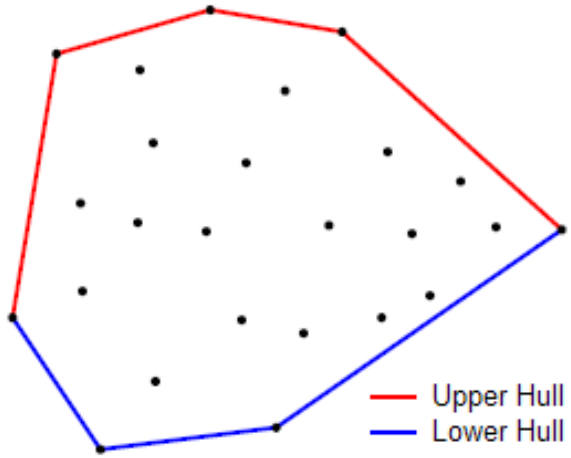

Figure: Convex Hull

- Using this geometric simplification, we can determine the relevant probabilities to allow us to use Bayes' Rule in the end.

$$p(w_1 = i|z_1 = k) = \frac{p(z_1 = k|w_1 = i)p(w_1 = i)}{\sum_i p(z_1 = k|w_1 = i')p(w_1 = i')} \quad (4)$$

Anchor words

Finding anchor words

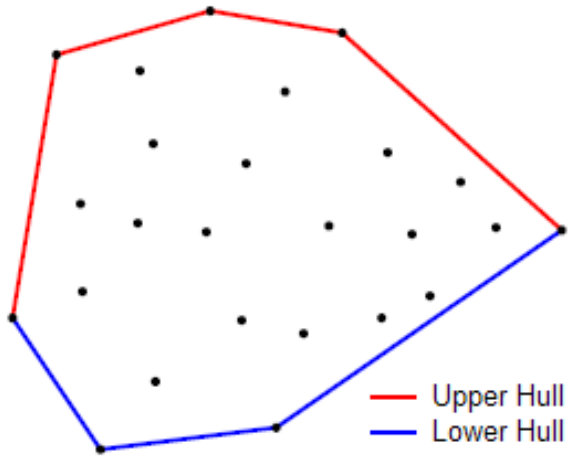

Figure: Convex Hull

- Previous algorithm: tests whether each of the V points is a vertex of the convex hull (and thus an anchor word) using the linear programming technique
  - Inefficient

Anchor words

Efficient algorithm

- Iterative algorithm
  - Finds farthest point from subspace spanned by anchor words so far
  - Farthest point will be the new anchor word
- Finds anchor word most different from the ones found so far
- Terminates when it has found K anchor (K is input to algorithm, # topics)