# Topic Modelling

## 1. The Author-Topic Model for Authors and Documents

Authors: Michal Rosen-Zvi; Thomas Griffiths; Mark Steyvers…

Link: <https://arxiv.org/abs/1207.4169>

**Aim:**

The paper introduces the author-topic model, which is an extension of the Latent Dirichlet Allocation (LDA) model. The primary aim is to incorporate authorship information into the generative model of documents. This allows for modeling not only the content of documents but also the interests of the authors.

**Data:**

The model was applied to a collection of 1,740 papers from the NIPS conference and 160,000 abstracts from the CiteSeer database. The data involves documents with multiple authors.

**Assumptions:**

* The exchangeability assumption (bag-of-word).
* The model assumes that each author is associated with a multinomial distribution over topics, and each topic is associated with a multinomial distribution over words.
* It also assumes that a document authored by multiple authors is a mixture of the distributions associated with each of the authors.

**Model Details:A diagram of a book

Description automatically generated**

The author-topic model is a generative model that extends LDA by including a prior distribution over authors. For each word in a document, an author is first selected (uniformly), then a topic is chosen from the distribution specific to that author, and finally, the word is generated from the topic’s distribution over words.

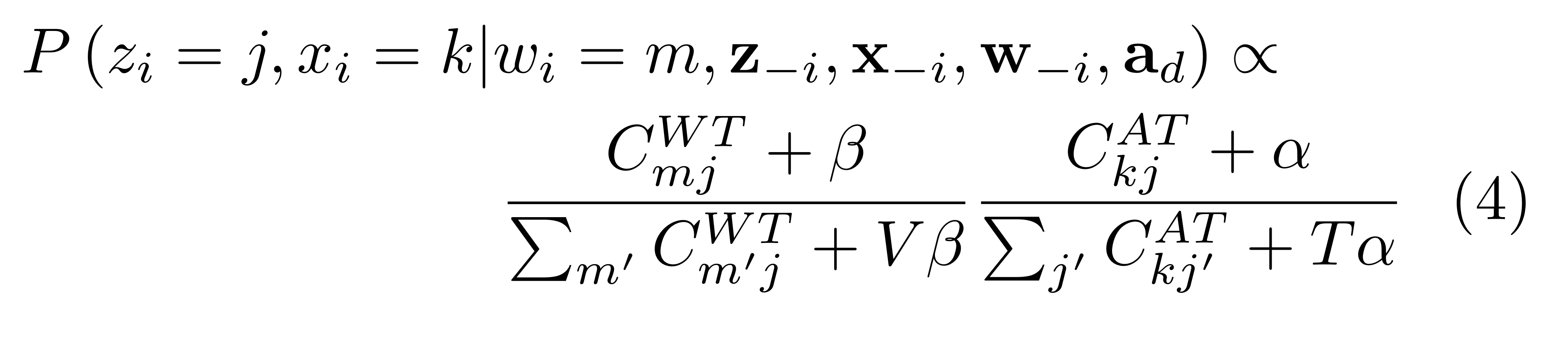
**Parameters:**

α,β are not estimated. They are fixed at 50/T and 0.01 respectively

There are 2 unknow parameters:

1. the distribution of topics for each author, θ
2. the distribution of words for each topic, φ

z and x are latent variables drawn as a pair (z,x) from the following posterior distribution



**Algorithms Used:**

The model uses Gibbs sampling to estimate the distributions of topics and authors. The posterior distribution involved is the equation (4) shown above.

The algorithm is initialized by assigning words to random topics and authors. Each iteration of the algorithm involves applying Equation 4 to every word token in the document collection.

The output of the algorithm is:

1. **Topic-Word Distributions,** φmj**:** the probability of word **m** appearing in topic **j**.
2. **Author-Topic Distributions,** θkj**:** the probability that author **k** writes about topic **j**.
3. **Topic Assignments for Words,** zi**:** the final topic assignments for each word in the corpus, indicating which topic each word belongs to.

A math equations on a white background

Description automatically generated

Where β is the hyperparameter and V is the size of the vocabulary. C(WT,mj) means the number of time the word m is assigned to topic j, taken from the matrix WT.

Two matrices are maintained in the process:

* Word-Topic Matrix (WT): This matrix holds the count of how many times each word m is associated with each topic j. It’s updated in each iteration of Gibbs sampling.
* Author-Topic Matrix (AT): This matrix holds the count of how many times each topic j is associated with each author k. This also gets updated with each Gibbs sampling iteration.

**Similarity to our work:**

We have the information of which user is related to the document. If we can use community detection in the first step to first find out the communities, we can essentially use this model by substituting ad into cd which means the community that the user of the document belongs to.

In the result, we can expect to get a community-topic distribution, θ.

However, in the author model, an author is uniformly drawn from a group of authors ad who have written the paper together. In our case, there may not be a need to choose further within the cd. Meaning that our θ will give information of the topic distribution of each community which is what we want. Therefore, we can run a loop to loop through all communities and find their respective θ.

What about mixed membership community?

## 2. Modelling Topic and Community Structure in Social Tagging: the TTR-LDA-Community Model

Authors: Daifeng Li, Ying Ding, Cassidy Sugimoto, Bing He, Jie Tang, Erjia Yan, Nan Lin, Zheng Qin, Tianxi Dong

Link: <https://yingding.ischool.utexas.edu/Publication/MTC-ST.pdf>

**Aim：**

The paper introduces the TTR-LDA-Community model, which integrates topic modelling and community detection within social tagging systems. The primary goal is to analyse the interplay between topics and community structure, revealing how communities are formed based on shared interests and how these communities evolve over time.

**Data:**

The model is applied to data from Delicious, a social bookmarking service, covering the period from 2005 to 2008. The dataset includes 50,000 taggers, 354,522 unique resources, and the corresponding tags. The data is divided into several time slices to observe the evolution of community structure and topic distribution over time.

**Assumptions:**

1. Topic Distribution: Each tagger (user) is associated with a multinomial distribution over topics.
2. Integration Mechanism: The model assumes that the topic distributions of individual taggers can be aggregated to represent the topic distribution of the entire community.

**Model Description:**

The TTR-LDA model is an adaptation of the Author-Conference-Topic (ACT) model. It is a three-layer Bayesian model that captures the relationships between taggers (users), tags, and resources within a social tagging system.

•**First Layer:** Represents taggers involved in creating posts.

•**Second Layer:** Represents the topics that link taggers to resources.

•**Third Layer:** Represents the resources being tagged.

A diagram of a bookmarking

Description automatically generated

**Parameters:**

Known Parameters:

1. **α:** Hyperparameter for the Dirichlet distribution generating topic distributions for taggers. Fixed at 50/K, where K is the number of topics.
2. **β:** Hyperparameter for the Dirichlet distribution generating word distributions for topics. Fixed at 0.01.
3. **μ:** Hyperparameter for the Dirichlet distribution generating resource distributions for topics. Fixed at 0.1.

Inferred Parameters Using Gibbs Sampling:

1. **θ:** The distribution of topics for each tagger. Represents the probability that a given tagger will use a particular topic.
2. **φ:** The distribution of words (tags) for each topic. Represents the probability of a word being associated with a specific topic.
3. **ψ:** The distribution of resources for each topic. Represents the probability of a resource being associated with a specific topic.

Latent Variables:

1. **z:** Represents the assignment of topics to tags.
2. **x:** Represents the assignment of topics to taggers.

Gibbs sampling is used.

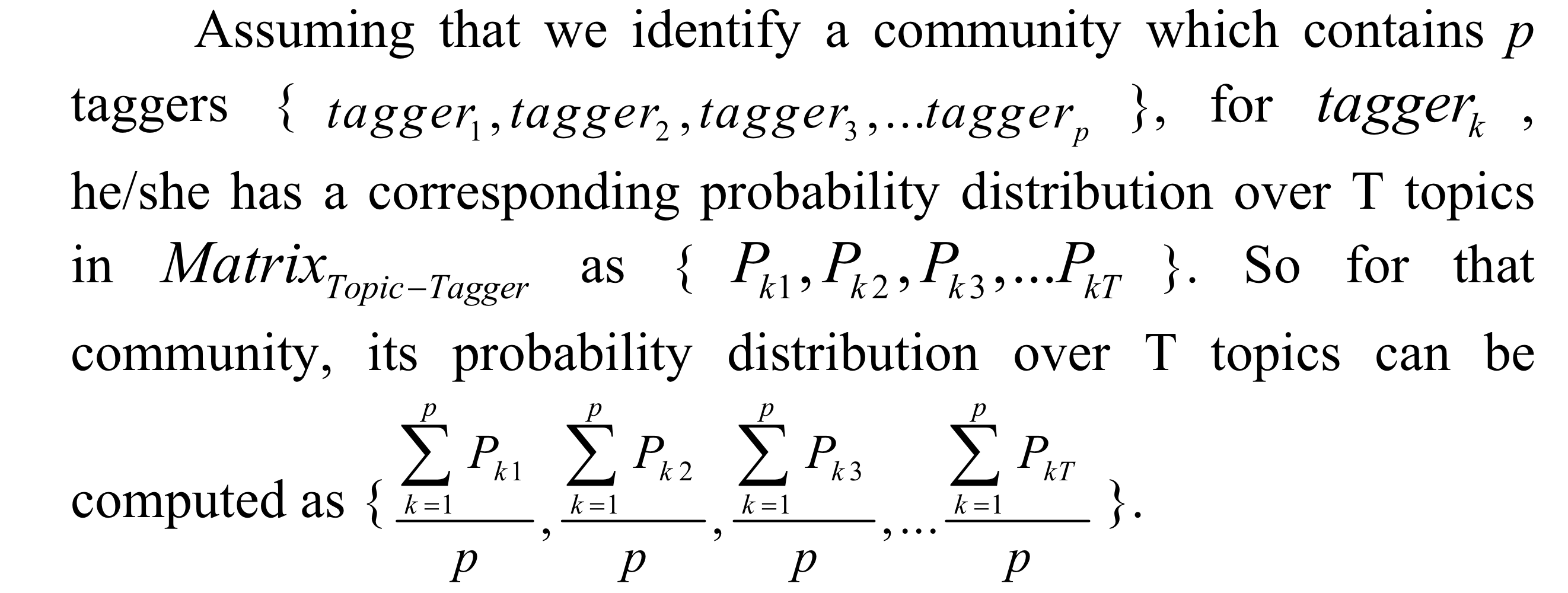
**Output of the Algorithm:**

1. **θkj:** The probability that tagger **k** is associated with topic **j**. This distribution helps profile taggers based on their thematic preferences.
2. **φmj:** The probability that word **m** appears in topic **j**. This distribution defines each topic in terms of the words most characteristic of it.
3. **ψrj:** The probability that resource **r** is associated with topic **j**. This distribution links resources to topics, allowing for content recommendation based on thematic relevance.

**Similarity to our work:**

This paper is included because it talks about how to integrate the results from community detection and topic modelling to get a topic distribution for a community. This is done by:

1. For a community containing ‘p’ users, each user has their corresponding topic distributions over T distribution
2. For all the topics present in the community (union of the topic distributions), the topic distribution T is calculated simply finding the mean:



The generative process is omitted since I think this model is not applicable to our work because this estimates 3 variables (θ, φ,ψ) while our work only needs 2. This goes the same for the Author-Conference-Topic (ACT) model.

Meaning that, one other way to conduct our work is that we can even use the most basic LDA model, finding out the topic-distribution for each node. Then, we can consider using this inference mechanism as well as the result from the community detection to find out the topic distribution for each community. Else, we can also figure out other inference mechanism since this one is rather too simplified.

## 3. Hierarchical Dirichlet Process

Authors: Yee Whye Teh, Michael I. Jordan, Matthew J. Beal and David M. Blei

Link: http://www.jstor.org/stable/27639773

**Aim:**

The paper introduces the **Hierarchical Dirichlet Process (HDP)**, a nonparametric Bayesian approach designed to model grouped data where observations within each group are assumed to be generated from a mixture model. The primary goal is to allow the sharing of mixture components (e.g., topics) across groups, while simultaneously inferring the number of these components.

**Assumptions:**

1. The base measure for the group-specific DPs is itself drawn from a higher-level DP, ensuring that the mixture components (topics) are shared across groups.
2. Observations are organised into groups and assumed to be exchangeable both within each group and across groups.
3. Base distribution H is conjugate to the data distribution.

**Parameters:**

Interpreted in terms of topic modelling

1. Known Parameters:
   1. γ: Concentration parameter of the global DP.
   2. αj: Concentration parameter of the DP specific to each group j.
   3. H: Base measure for the global DP. Drawn from a Dirichlet distribution.
2. Inferred Parameters Using MCMC:
   1. G0: Sampled from DP(γ,H). Countably infinite collection of multinomial probability vectors. This is the set of all topics that can be used for a given corpus. Shared by all documents in the corpus.
   2. Gj: Sampled from DP(αj , G0). The distribution for each document. This subset the topics to be used in document j.
   3. θji: This is a multinomial distribution over topics. It determines the probability of each topic being chosen to generate the next word in the document.
   4. Indicator variables for the assignment of data points to specific mixture components. tij / zji which tells you which φk is selected.
   5. φk is sampled directly from H. It represents the distribution over words for the k topic

**Chinese Restaurant Process (CRP):**

1. The Setup：
   1. Restaurant: Imagine a Chinese restaurant with an infinite number of tables.
   2. Customers: The customers arriving at the restaurant represent the data points that need to be clustered.
   3. Tables: The tables in the restaurant represent the clusters. Each table can seat multiple customers, and each table serves a specific dish (which corresponds to a cluster parameter, such as a topic in topic modelling).
2. The Seating Process

When each customer arrives at the restaurant, they choose where to sit according to the following rules:

First Customer:

* The first customer enters the restaurant and sits at the first table.

Subsequent Customers:

* When a new customer enters, they can either:
  + Sit at an occupied table: The probability that the customer chooses an existing table is proportional to the number of customers already seated at that table. This means that tables with more customers are more likely to attract new customers.
  + Start a new table: The customer can also choose to sit at a new, unoccupied table. The probability of starting a new table is proportional to a concentration α0.

**Chinese Restaurant Franchise (CRF):**

**A group of circles with numbers and circles

Description automatically generated**

CRF extends the CRP concept to multiple restaurants (or groups). Imagine there are multiple restaurants (e.g., multiple documents), and each restaurant follows a similar CRP to decide how its customers are seated at tables. However, all the restaurants share the same menu of dishes (which represent topics or components in the model), meaning that while customers in different restaurants may be seated at different tables, the dishes served at these tables are drawn from the same global set of dishes.

**Global Menu (Base Measure):** There is a global probability distribution over dishes (topics or components) that is shared across all restaurants (groups).

**Restaurants (Groups):** Each restaurant has its own CRP, but when a customer starts a new table, they pick a dish from the global menu.

**Tables (Clusters within a Group):** Different tables within the same restaurant can serve the same dish, and different restaurants can serve the same dish at their tables. This sharing of dishes among restaurants allows the model to tie together the groups, facilitating the sharing of statistical strength across groups.

**Relating to topic modelling:**

**A diagram of a diagram

Description automatically generated**

1. This provides a non-parametric way extension of doing LDA, therefore we do not need to specify the number of topics in advance.
2. The base distribution H should be sampled from a Dirichlet distribution like how we do it in LDA.
3. Convert this H into G0 using Dirichlet Process. This provides a countable infinite collection of multinomial probability vector. These can be viewed as the set of all topics that can be used in each corpus.
4. For each j document, we sample a Gj from G0. This selects the subset of topics to be used in document j.
5. From Gj, we generate the document by repeatedly sampling specific multinomial probability vectors θji from Gj.
6. Sample a topic zji from the document’s topic distribution Gj this gives us which topic distribution to use (φzji)
7. Sample a word from φzji

**Algorithm used and inference:**

1. Gibbs sampling is used.
2. The posterior distribution is obtained…

**Outputs:**

Assuming this is a one level hierarchy structure.

1. G0, global topic distribution (For the entire corpus)
2. Gj, individual topic distribution (document specific)

**Similarity to our work:**

This hierarchical structure can be extended. If we use 2 level of hierarchy, first level can be used to find the topic distribution for each node. The second level of hierarchy can be used to find topic distribution among multiple corpora.

This is non-parametric! The number of resulting groups is equal to the number of Gj.

Focus on potential layers. Is it a parametric or non-parametric one?.

Should there be upper limits.

How to decide whether to add another layer.

1. Read the current report more in depth and generate a more detailed introduction of the paper.
2. Understand the process of adding layers 🡺 find more relevant papers.

Focus on how to add potential layers.

Try find some dataset by yourself. Requirement: network, covariate dataset: this should be about text data. Sample size should be around 1000

Meaning that a big community group can have multiple subgroups. Now you want to sample Gjk which subsets Gj with 3 variables, alpha, and the adjacency matrix A. yes it should depend on the entire network’s A(preferably).

**Authors:**

**Link:**

**Aim:**

**Assumptions:**

**Model details:**

**Data used:**

**Output:**