Probabilistic Models for Graphs, and Intro to Nonparametric Bayesian Modeling

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

March 25, 2019

Modeling Graphs

- Often we wish to understand the underlying structure (e.g., communities/groups/topics) in a graph, predict links, classify nodes, visualize, etc.
- An example graph† (a 575,000 node citation network of papers; each node is a paper):

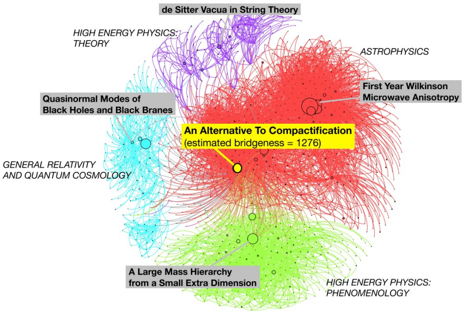


Fig. 1. The discovered community structure in a subgraph of the arXiv citation network (21). The figure shows the top four link communities that include citations to "An alternative to compactification" (22), an article that bridges several communities. We visualize the links between the articles and show some highly cited titles. Each community is labeled with its dominant subject area; nodes are sized by their bridgeness (39), an inferred measure of their impact on multiple communities. This is taken from an analysis of the full 575,000 node network.

† Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Modeling Graphs

- Often we wish to understand the underlying structure (e.g., communities/groups/topics) in a graph, predict links, classify nodes, visualize, etc.
- An example graph[†] (a 575,000 node citation network of papers; each node is a paper):

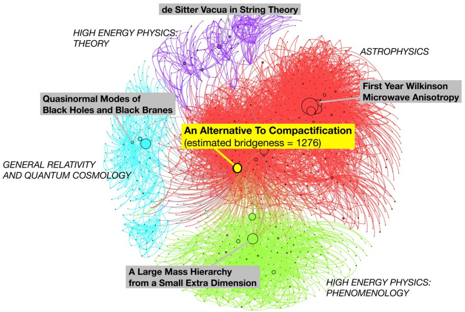


Fig. 1. The discovered community structure in a subgraph of the arXiv citation network (21). The figure shows the top four link communities that include citations to "An alternative to compactification" (22), an article that bridges several communities. We visualize the links between the articles and show some highly cited titles. Each community is labeled with its dominant subject area; nodes are sized by their bridgeness (39), an inferred measure of their impact on multiple communities. This is taken from an analysis of the full 575,000 node network.

- Statistical models of graphs can help us solve these problems

[†] Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K
- We can model each link/non-link $A_{nm} \in \{0,1\}$ via a probability model, e.g.

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta) = f(\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

  where f is some function of $\boldsymbol{z}_n, \boldsymbol{z}_m$ and params θ, and returns a probability

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K
- We can model each link/non-link $A_{nm} \in \{0, 1\}$ via a probability model, e.g.

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta) = f(\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

  where f is some function of $\boldsymbol{z}_n, \boldsymbol{z}_m$ and params θ, and returns a probability
- The overall probability of the observed graph

$$p(\mathbf{A} | \mathbf{Z}, \theta) = \prod_{n,m} p(A_{nm} | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K
- We can model each link/non-link $A_{nm} \in \{0,1\}$ via a probability model, e.g.

$$p(A_{nm} = 1|\boldsymbol{z}_n, \boldsymbol{z}_m, \theta) = f(\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

  where f is some function of $\boldsymbol{z}_n, \boldsymbol{z}_m$ and params θ, and returns a probability
- The overall probability of the observed graph

$$p(\mathbf{A}|\mathbf{Z}, \theta) = \prod_{n,m} p(A_{nm}|\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

- Various models differ in terms of what the embeddings $\boldsymbol{z}_n$ look like, and what f is defined as

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K
- We can model each link/non-link $A_{nm} \in \{0, 1\}$ via a probability model, e.g.

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta) = f(\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

  where f is some function of $\boldsymbol{z}_n, \boldsymbol{z}_m$ and params θ, and returns a probability
- The overall probability of the observed graph

$$p(\mathbf{A}|\mathbf{Z}, \theta) = \prod_{n,m} p(A_{nm}|\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

- Various models differ in terms of what the embeddings $\boldsymbol{z}_n$ look like, and what f is defined as
- Some representative models

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K
- We can model each link/non-link $A_{nm} \in \{0,1\}$ via a probability model, e.g.

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta) = f(\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

  where f is some function of $\boldsymbol{z}_n, \boldsymbol{z}_m$ and params θ, and returns a probability

- The overall probability of the observed graph

$$p(\mathbf{A}|\mathbf{Z}, \theta) = \prod_{n,m} p(A_{nm}|\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

- Various models differ in terms of what the embeddings $\boldsymbol{z}_n$ look like, and what f is defined as
- Some representative models
  - Latent Space Model
  - Stochastic Blockmodel
  - Mixed-Membership Blockmodel (MMSB)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space
- One such possible model would be

$$p(A_{nm} = 1|\boldsymbol{z}_n, \boldsymbol{z}_m) = \sigma(-||\boldsymbol{z}_n - \boldsymbol{z}_m||)$$

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space
- One such possible model would be

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m) = \sigma(-||\boldsymbol{z}_n - \boldsymbol{z}_m||)$$

- A reasonable model for link prediction

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space
- One such possible model would be

$$p(A_{nm} = 1|\boldsymbol{z}_n, \boldsymbol{z}_m) = \sigma(-||\boldsymbol{z}_n - \boldsymbol{z}_m||)$$

- A reasonable model for link prediction
- However, the real-valued embeddings in LSM don't provide a good interpretability

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space
- One such possible model would be

$$p(A_{nm} = 1|\boldsymbol{z}_n, \boldsymbol{z}_m) = \sigma(-||\boldsymbol{z}_n - \boldsymbol{z}_m||)$$

- A reasonable model for link prediction
- However, the real-valued embeddings in LSM don't provide a good interpretability
  - Therefore not very ideal for discovering clusters etc.

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Latent Space Model for Graphs

- LSM[†] assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space
- One such possible model would be

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m) = \sigma(-||\boldsymbol{z}_n - \boldsymbol{z}_m||)$$

- A reasonable model for link prediction
- However, the real-valued embeddings in LSM don't provide a good interpretability
  - Therefore not very ideal for discovering clusters etc.
- Blockmodels and its variants has such properties as we will see next

[†] Latent space approaches to social network analysis (Hoff et al, 2002)

Stochastic Blockmodel

- SBM† assumes each node belongs to a cluster/community or "block" (total K clusters)

† Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM† assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K

† Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1|\boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM† assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like
    - For $n = 1, \ldots, N$

$$\boldsymbol{z}_n \sim \text{multinoulli}(\pi)$$

† Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like
  - For $n = 1, \ldots, N$

$$\boldsymbol{z}_n \sim \text{multinoulli}(\pi)$$

  - For $n = 1, \ldots, N$

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like
  - For $n = 1, \ldots, N$

$$\boldsymbol{z}_n \sim \text{multinoulli}(\pi)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like
  - For $n = 1, \ldots, N$

$$\boldsymbol{z}_n \sim \text{multinoulli}(\pi)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

$$A_{nm} \sim \text{Bernoulli}(\eta_{\boldsymbol{z}_n, \boldsymbol{z}_m})$$

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Stochastic Blockmodel

- SBM[†] assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like
  - For $n = 1, \ldots, N$
  $$\boldsymbol{z}_n \sim \text{multinoulli}(\pi)$$
  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$
    $$A_{nm} \sim \text{Bernoulli}(\eta_{\boldsymbol{z}_n, \boldsymbol{z}_m})$$
- Note: In the fully Bayesian version, π and η can also be given priors

[†] Estimation and prediction for stochastic blockmodels for graphs with latent block structure (Snijders and Nowicki, 1997), Discovering latent classes in relational data (Kemp et al, 2004)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB[†] assumes each node n to have a $K \times 1$ probability vector π_n

[†] Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB† assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities

† Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB[†] assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

[†] Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB† assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

† Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB† assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

$$\boldsymbol{z}_{n \to m} \sim \text{multinoulli}(\pi_n)$$

† Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB[†] assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

$$\begin{aligned} \boldsymbol{z}_{n \to m} &\sim \text{multinoulli}(\pi_n) \\ \boldsymbol{z}_{m \to n} &\sim \text{multinoulli}(\pi_m) \end{aligned}$$

[†] Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB† assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

$$\begin{aligned} \boldsymbol{z}_{n\to m} &\sim \text{multinoulli}(\pi_n) \\ \boldsymbol{z}_{m\to n} &\sim \text{multinoulli}(\pi_m) \\ A_{nm} &\sim \text{Bernoulli}(\eta_{\boldsymbol{z}_{n\to m}, \boldsymbol{z}_{m\to n}}) \end{aligned}$$

† Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB[†] assumes each node n to have a $K \times 1$ probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

$$\begin{aligned} \boldsymbol{z}_{n \rightarrow m} &\sim \text{multinoulli}(\pi_n) \\ \boldsymbol{z}_{m \rightarrow n} &\sim \text{multinoulli}(\pi_m) \\ A_{nm} &\sim \text{Bernoulli}(\eta_{\boldsymbol{z}_{n \rightarrow m}, \boldsymbol{z}_{m \rightarrow n}}) \end{aligned}$$

- Unlike SBM in which node n has a unique one-hot $\boldsymbol{z}_n$ vector, in MMSB, each node n has an interaction-specific cluster assignment

[†] Mixed-Membership Stochastic Blockmodel (Airoldi et al, 2008), Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013)

Modeling Graphs: Some Other Comments

- A lot of work on various extensions* of LSM, SBM, MMSB, etc.
- A lot of work on scalable Bayesian inference# in these models (e.g., online MCMC/VI)

---

*Nonparametric Bayesian modeling of complex networks: an introduction (Schmidt et al, 2013), #Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013), †Graph Convolutional Networks (Kipf and Welling, 2016), ‡GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models (You et al, 2018)

Modeling Graphs: Some Other Comments

- A lot of work on various extensions* of LSM, SBM, MMSB, etc.
- A lot of work on scalable Bayesian inference# in these models (e.g., online MCMC/VI)
- Some of the recent trends in this area

*Nonparametric Bayesian modeling of complex networks: an introduction (Schmidt et al, 2013), #Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013), †Graph Convolutional Networks (Kipf and Welling, 2016), ‡GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models (You et al, 2018)

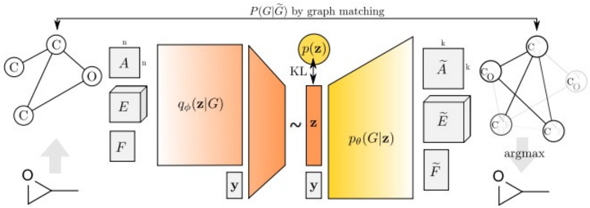
Modeling Graphs: Some Other Comments

- A lot of work on various extensions* of LSM, SBM, MMSB, etc.
- A lot of work on scalable Bayesian inference# in these models (e.g., online MCMC/VI)
- Some of the recent trends in this area
  - Combining classical prob. models of graphs with graph neural nets (e.g., Graph Convolutional Net†)

*Nonparametric Bayesian modeling of complex networks: an introduction (Schmidt et al, 2013), #Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013), †Graph Convolutional Networks (Kipf and Welling, 2016), ‡GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models (You et al, 2018)

Modeling Graphs: Some Other Comments

- A lot of work on various extensions* of LSM, SBM, MMSB, etc.
- A lot of work on scalable Bayesian inference# in these models (e.g., online MCMC/VI)
- Some of the recent trends in this area
  - Combining classical prob. models of graphs with graph neural nets (e.g., Graph Convolutional Net†)
  - Learning to generate graphs‡ (just like image or text generation in deep learning)



*Nonparametric Bayesian modeling of complex networks: an introduction (Schmidt et al, 2013), #Efficient discovery of overlapping communities in massive networks (Gopalan and Blei, 2013), †Graph Convolutional Networks (Kipf and Welling, 2016), ‡GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models (You et al, 2018)

Nonparametric Bayesian Modeling

(A way of learning the “right” model size/complexity)

Motivating Problem: Mixture Models

- A mixture model can be used to cluster/partition the data into multiple groups
- Defined by K component distributions (e.g., K Gaussians for a Gaussian Mixture Model)
- Every component distribution $p(\boldsymbol{x}|\phi_k)$ has a mixing weight $\pi_k \in (0,1)$, and $\sum_{k=1}^{K} \pi_k = 1$

Motivating Problem: Mixture Models

- A mixture model can be used to cluster/partition the data into multiple groups
- Defined by K component distributions (e.g., K Gaussians for a Gaussian Mixture Model)
- Every component distribution $p(\boldsymbol{x}|\phi_k)$ has a mixing weight $\pi_k \in (0,1)$, and $\sum_{k=1}^{K} \pi_k = 1$
- The distribution of any observation $\boldsymbol{x}$

$$p(\boldsymbol{x}|\pi, \phi) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\phi_k)$$

where $\pi = \{\pi_k\}_{k=1}^{K}$ and $\phi = \{\phi_k\}_{k=1}^{K}$

Motivating Problem: Mixture Models

- A mixture model can be used to cluster/partition the data into multiple groups
- Defined by K component distributions (e.g., K Gaussians for a Gaussian Mixture Model)
- Every component distribution $p(\boldsymbol{x}|\phi_k)$ has a mixing weight $\pi_k \in (0,1)$, and $\sum_{k=1}^K \pi_k = 1$
- The distribution of any observation $\boldsymbol{x}$

$$p(\boldsymbol{x}|\pi,\phi) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\phi_k)$$

where $\pi = \{\pi_k\}_{k=1}^K$ and $\phi = \{\phi_k\}_{k=1}^K$

- Question: What's the "right" number of clusters?

Motivating Problem: Mixture Models

- A mixture model can be used to cluster/partition the data into multiple groups
- Defined by K component distributions (e.g., K Gaussians for a Gaussian Mixture Model)
- Every component distribution $p(\boldsymbol{x}|\phi_k)$ has a mixing weight $\pi_k \in (0,1)$, and $\sum_{k=1}^{K} \pi_k = 1$
- The distribution of any observation $\boldsymbol{x}$

$$p(\boldsymbol{x}|\pi,\phi) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\phi_k)$$

  where $\pi = \{\pi_k\}_{k=1}^{K}$ and $\phi = \{\phi_k\}_{k=1}^{K}$

- Question: What's the "right" number of clusters? Can do Bayesian model comparison (try different K's and compute marginal likelihood for each choice of K).

Motivating Problem: Mixture Models

- A mixture model can be used to cluster/partition the data into multiple groups
- Defined by K component distributions (e.g., K Gaussians for a Gaussian Mixture Model)
- Every component distribution $p(\boldsymbol{x}|\phi_k)$ has a mixing weight $\pi_k \in (0,1)$, and $\sum_{k=1}^{K} \pi_k = 1$
- The distribution of any observation $\boldsymbol{x}$

$$p(\boldsymbol{x}|\pi, \phi) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\phi_k)$$

  where $\pi = \{\pi_k\}_{k=1}^{K}$ and $\phi = \{\phi_k\}_{k=1}^{K}$
- Question: What's the "right" number of clusters? Can do Bayesian model comparison (try different K's and compute marginal likelihood for each choice of K). But that can be expensive.

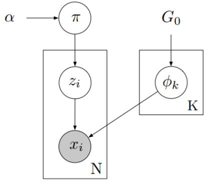
Motivating Problem: Mixture Models

- A mixture model can be used to cluster/partition the data into multiple groups
- Defined by K component distributions (e.g., K Gaussians for a Gaussian Mixture Model)
- Every component distribution $p(\boldsymbol{x}|\phi_k)$ has a mixing weight $\pi_k \in (0,1)$, and $\sum_{k=1}^{K} \pi_k = 1$
- The distribution of any observation $\boldsymbol{x}$

$$p(\boldsymbol{x}|\pi,\phi) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\phi_k)$$

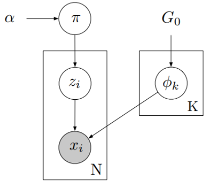
  where $\pi = \{\pi_k\}_{k=1}^{K}$ and $\phi = \{\phi_k\}_{k=1}^{K}$
- Question: What's the "right" number of clusters? Can do Bayesian model comparison (try different K's and compute marginal likelihood for each choice of K). But that can be expensive.
- How about having a single model but allowing the number of clusters to "grow" with data?

Prelude: A Bayesian Mixture Model (with fixed K)



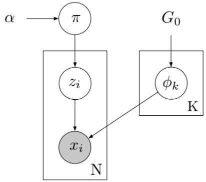
- Assuming some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the generative model can be defined as follows

Prelude: A Bayesian Mixture Model (with fixed K)



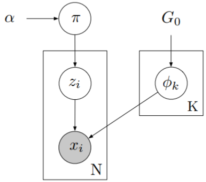
- Assuming some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the generative model can be defined as follows
  - Draw mixture proportion vector $\pi = [\pi_1, \ldots, \pi_K]$ from the prior Dirichlet$(\alpha/K, \ldots, \alpha/K)$

Prelude: A Bayesian Mixture Model (with fixed K)



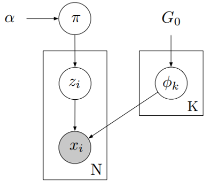
- Assuming some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the generative model can be defined as follows
  - Draw mixture proportion vector $\pi = [\pi_1, \ldots, \pi_K]$ from the prior Dirichlet$(\alpha/K, \ldots, \alpha/K)$
  - Draw parameters $\{\phi_k\}_{k=1}^K$ of each mixture component i.i.d. from a prior "base distribution" G_0 (note: choice of G_0 depends on what the component distributions are; e.g., for Gaussians, G_0 can be NIW)

Prelude: A Bayesian Mixture Model (with fixed K)



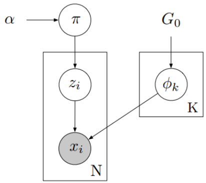
- Assuming some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the generative model can be defined as follows
  - Draw mixture proportion vector $\pi = [\pi_1, \ldots, \pi_K]$ from the prior Dirichlet$(\alpha/K, \ldots, \alpha/K)$
  - Draw parameters $\{\phi_k\}_{k=1}^K$ of each mixture component i.i.d. from a prior "base distribution" G_0 (note: choice of G_0 depends on what the component distributions are; e.g., for Gaussians, G_0 can be NIW)
  - Draw the data: For each observation $i = 1, \ldots, N$

Prelude: A Bayesian Mixture Model (with fixed K)



- Assuming some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the generative model can be defined as follows
  - Draw mixture proportion vector $\pi = [\pi_1, \ldots, \pi_K]$ from the prior Dirichlet$(\alpha/K, \ldots, \alpha/K)$
  - Draw parameters $\{\phi_k\}_{k=1}^K$ of each mixture component i.i.d. from a prior "base distribution" G_0 (note: choice of G_0 depends on what the component distributions are; e.g., for Gaussians, G_0 can be NIW)
  - Draw the data: For each observation $i = 1, \ldots, N$
    - Draw a cluster id $\boldsymbol{z}_i \in \{1, \ldots, K\}$ from multinoulli(π)

Prelude: A Bayesian Mixture Model (with fixed K)



- Assuming some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the generative model can be defined as follows
  - Draw mixture proportion vector $\pi = [\pi_1, \ldots, \pi_K]$ from the prior Dirichlet$(\alpha/K, \ldots, \alpha/K)$
  - Draw parameters $\{\phi_k\}_{k=1}^K$ of each mixture component i.i.d. from a prior "base distribution" G_0 (note: choice of G_0 depends on what the component distributions are; e.g., for Gaussians, G_0 can be NIW)
  - Draw the data: For each observation $i = 1, \ldots, N$
    - Draw a cluster id $\boldsymbol{z}_i \in \{1, \ldots, K\}$ from multinoulli(π)
    - Suppose $\boldsymbol{z}_i = k$. Draw $\boldsymbol{x}_i$ from $p(\boldsymbol{x}|\phi_k)$

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW
- Due to conjugacy, we can easily derive a Gibbs sampler for this model

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW
- Due to conjugacy, we can easily derive a Gibbs sampler for this model
- The basic Gibbs sampler is sketched below

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW
- Due to conjugacy, we can easily derive a Gibbs sampler for this model
- The basic Gibbs sampler is sketched below
  - Randomly initialize Z, π, ϕ. Repeat until we have enough samples

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW
- Due to conjugacy, we can easily derive a Gibbs sampler for this model
- The basic Gibbs sampler is sketched below
  - Randomly initialize Z, π, ϕ. Repeat until we have enough samples
    - Sample π as (due to Dirichlet-multinomial conjugacy)

$$\pi|\mathbf{Z} \sim \text{Dirichlet}(n_1 + \alpha/K, \ldots, n_K + \alpha/K)$$

.. where n_k is the number of observations currently assigned to cluster k

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW
- Due to conjugacy, we can easily derive a Gibbs sampler for this model
- The basic Gibbs sampler is sketched below
  - Randomly initialize Z, π, ϕ. Repeat until we have enough samples
    - Sample π as (due to Dirichlet-multinomial conjugacy)

$$\pi|\mathbf{Z} \sim \text{Dirichlet}(n_1 + \alpha/K, \ldots, n_K + \alpha/K)$$

      .. where n_k is the number of observations currently assigned to cluster k

    - Sample each $\boldsymbol{z}_i$ using

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \pi, \phi, \mathbf{X}) \propto p(\boldsymbol{z}_i|\pi) \times p(\boldsymbol{x}_i|\phi_k) = \pi_k \times p(\boldsymbol{x}_i|\phi_k)$$

      .. (note: the above is equivalent to computing the above posterior probabilities for $k = 1, \ldots, K$ and drawing $\boldsymbol{z}_i$ from a multinoulli with probability parameter vector given by these posterior probabilities)

Bayesian Inference for Mixture Models: A Gibbs Sampler

- Assume a Gaussian Mixture Model (GMM) with $\phi_k = (\mu_k, \Sigma_k)$. Thus G_0 can be NIW
- Due to conjugacy, we can easily derive a Gibbs sampler for this model
- The basic Gibbs sampler is sketched below
  - Randomly initialize Z, π, ϕ. Repeat until we have enough samples
    - Sample π as (due to Dirichlet-multinomial conjugacy)

      $$\pi|\mathbf{Z} \sim \text{Dirichlet}(n_1 + \alpha/K, \ldots, n_K + \alpha/K)$$

      .. where n_k is the number of observations currently assigned to cluster k
    - Sample each $\boldsymbol{z}_i$ using

      $$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \pi, \phi, \mathbf{X}) \propto p(\boldsymbol{z}_i|\pi) \times p(\boldsymbol{x}_i|\phi_k) = \pi_k \times p(\boldsymbol{x}_i|\phi_k)$$

      .. (note: the above is equivalent to computing the above posterior probabilities for $k = 1, \ldots, K$ and drawing $\boldsymbol{z}_i$ from a multinoulli with probability parameter vector given by these posterior probabilities)
    - Sample $\phi_k = (\mu_k, \Sigma_k)$ from NIW posterior given $\mathbf{Z}$ and $\mathbf{X}$ (NIW posterior also has a closed form).

A "Collapsed" Gibbs Sampler

- Let's integrate out π (due to Dirichlet-multinomial conjugacy) from prior $p(\boldsymbol{z}_i|\pi)$ and eliminate π

A "Collapsed" Gibbs Sampler

- Let's integrate out π (due to Dirichlet-multinomial conjugacy) from prior $p(\boldsymbol{z}_i|\pi)$ and eliminate π

$$p(\boldsymbol{z}_i = k|\alpha, \mathbf{Z}_{-i}) = \int p(\boldsymbol{z}_i = k|\pi)p(\pi|\mathbf{Z}_{-i})d\pi = \int \pi_k p(\pi|\mathbf{Z}_{-i})d\pi = \mathbb{E}_{p(\pi|\mathbf{Z}_{-i})}[\pi_k] = \frac{n_k + \alpha/K}{\alpha + N - 1}$$

.. where n_k is the number of examples (other than $\boldsymbol{x}_i$) assigned to cluster k

A "Collapsed" Gibbs Sampler

- Let's integrate out π (due to Dirichlet-multinomial conjugacy) from prior $p(\boldsymbol{z}_i|\pi)$ and eliminate π

$$p(\boldsymbol{z}_i = k|\alpha, \mathbf{Z}_{-i}) = \int p(\boldsymbol{z}_i = k|\pi)p(\pi|\mathbf{Z}_{-i})d\pi = \int \pi_k p(\pi|\mathbf{Z}_{-i})d\pi = \mathbb{E}_{p(\pi|\mathbf{Z}_{-i})}[\pi_k] = \frac{n_k + \alpha/K}{\alpha + N - 1}$$

.. where n_k is the number of examples (other than $\boldsymbol{x}_i$) assigned to cluster k

- The collapsed Gibbs sampler is sketched below

A "Collapsed" Gibbs Sampler

- Let's integrate out π (due to Dirichlet-multinomial conjugacy) from prior $p(\boldsymbol{z}_i|\pi)$ and eliminate π

$$p(\boldsymbol{z}_i = k|\alpha, \mathbf{Z}_{-i}) = \int p(\boldsymbol{z}_i = k|\pi)p(\pi|\mathbf{Z}_{-i})d\pi = \int \pi_k p(\pi|\mathbf{Z}_{-i})d\pi = \mathbb{E}_{p(\pi|\mathbf{Z}_{-i})}[\pi_k] = \frac{n_k + \alpha/K}{\alpha + N - 1}$$

.. where n_k is the number of examples (other than $\boldsymbol{x}_i$) assigned to cluster k

- The collapsed Gibbs sampler is sketched below
  - Randomly initialize Z, ϕ. Repeat until we have enough samples

A "Collapsed" Gibbs Sampler

- Let's integrate out π (due to Dirichlet-multinomial conjugacy) from prior $p(\boldsymbol{z}_i|\pi)$ and eliminate π

$$p(\boldsymbol{z}_i = k|\alpha, \mathbf{Z}_{-i}) = \int p(\boldsymbol{z}_i = k|\pi)p(\pi|\mathbf{Z}_{-i})d\pi = \int \pi_k p(\pi|\mathbf{Z}_{-i})d\pi = \mathbb{E}_{p(\pi|\mathbf{Z}_{-i})}[\pi_k] = \frac{n_k + \alpha/K}{\alpha + N - 1}$$

.. where n_k is the number of examples (other than $\boldsymbol{x}_i$) assigned to cluster k

- The collapsed Gibbs sampler is sketched below
  - Randomly initialize Z, ϕ. Repeat until we have enough samples
    - Sample each $\boldsymbol{z}_i$ as

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$$

.. (note: just like the uncollapsed case, this is equivalent to drawing $\boldsymbol{z}_i$ from a multinoulli with probability parameter vector given by the above posterior probabilities evaluated for $k = 1, \ldots, K$)

A "Collapsed" Gibbs Sampler

- Let's integrate out π (due to Dirichlet-multinomial conjugacy) from prior $p(\boldsymbol{z}_i|\pi)$ and eliminate π

$$p(\boldsymbol{z}_i = k|\alpha, \mathbf{Z}_{-i}) = \int p(\boldsymbol{z}_i = k|\pi)p(\pi|\mathbf{Z}_{-i})d\pi = \int \pi_k p(\pi|\mathbf{Z}_{-i})d\pi = \mathbb{E}_{p(\pi|\mathbf{Z}_{-i})}[\pi_k] = \frac{n_k + \alpha/K}{\alpha + N - 1}$$

  .. where n_k is the number of examples (other than $\boldsymbol{x}_i$) assigned to cluster k

- The collapsed Gibbs sampler is sketched below
  - Randomly initialize Z, ϕ. Repeat until we have enough samples
    - Sample each $\boldsymbol{z}_i$ as

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$$

      .. (note: just like the uncollapsed case, this is equivalent to drawing $\boldsymbol{z}_i$ from a multinoulli with probability parameter vector given by the above posterior probabilities evaluated for $k = 1, \ldots, K$)

    - Sample $\phi_k = (\mu_k, \Sigma_k)$ from NIW posterior given $\mathbf{Z}$ and $\mathbf{X}$

Nonparametric Bayesian Mixture Model

(.. which you get when you allow unbounded K)

(.. i.e., you allow $K \to \infty$)

Let's Have $K \to \infty$ (i.e., Unbounded)

- For the collapsed Gibbs sampler, we saw that $p(\boldsymbol{z}_i = k | \mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K) p(\boldsymbol{x}_i | \phi_k)$

Let's Have $K \to \infty$ (i.e., Unbounded)

- For the collapsed Gibbs sampler, we saw that $p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$
- As $K \to \infty$, the probability that $\boldsymbol{x}_i$ will be assigned to an existing cluster (say k) is given by

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto n_k \times p(\boldsymbol{x}_i|\phi_k)$$

Let's Have $K \to \infty$ (i.e., Unbounded)

- For the collapsed Gibbs sampler, we saw that $p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$
- As $K \to \infty$, the probability that $\boldsymbol{x}_i$ will be assigned to an existing cluster (say k) is given by

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto n_k \times p(\boldsymbol{x}_i|\phi_k)$$

- It's proportional to the no. of obs. already in cluster k (it's like a "rich gets richer" tendency)

Let's Have $K \to \infty$ (i.e., Unbounded)

- For the collapsed Gibbs sampler, we saw that $p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$
- As $K \to \infty$, the probability that $\boldsymbol{x}_i$ will be assigned to an existing cluster (say k) is given by

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto n_k \times p(\boldsymbol{x}_i|\phi_k)$$

- It's proportional to the no. of obs. already in cluster k (it's like a "rich gets richer" tendency)
- Now suppose there are a total of K_+ occupied clusters (with number of data points ≥ 1)

Let's Have $K \to \infty$ (i.e., Unbounded)

- For the collapsed Gibbs sampler, we saw that $p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$
- As $K \to \infty$, the probability that $\boldsymbol{x}_i$ will be assigned to an existing cluster (say k) is given by

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto n_k \times p(\boldsymbol{x}_i|\phi_k)$$

- It's proportional to the no. of obs. already in cluster k (it's like a "rich gets richer" tendency)
- Now suppose there are a total of K_+ occupied clusters (with number of data points ≥ 1)
- Number of empty clusters $= K - K_+$. Can think of all of these as a single unoccupied cluster

Let's Have $K \to \infty$ (i.e., Unbounded)

- For the collapsed Gibbs sampler, we saw that $p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto (n_k + \alpha/K)p(\boldsymbol{x}_i|\phi_k)$
- As $K \to \infty$, the probability that $\boldsymbol{x}_i$ will be assigned to an existing cluster (say k) is given by

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto n_k \times p(\boldsymbol{x}_i|\phi_k)$$

- It's proportional to the no. of obs. already in cluster k (it's like a "rich gets richer" tendency)
- Now suppose there are a total of K_+ occupied clusters (with number of data points ≥ 1)
- Number of empty clusters $= K - K_+$. Can think of all of these as a single unoccupied cluster
- The probability of $\boldsymbol{x}_i$ being assigned to this new (so far empty) cluster

$$p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto \alpha \times p(\boldsymbol{x}_i|G_0) \qquad \text{(WHY? Reason on the next slide!)}$$

where $p(\boldsymbol{x}_i|G_0) = \int p(\boldsymbol{x}_i|\phi_{new})p(\phi_{new}|G_0)d\phi_{new}$

Let's Have $K \to \infty$ (i.e., Unbounded)

- The probability of $\boldsymbol{x}_i$ being assigned to the new (so far empty) cluster

$$p(\boldsymbol{z}_i = k_{new} | \mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto \alpha \times p(\boldsymbol{x}_i | G_0)$$

Let's Have $K \to \infty$ (i.e., Unbounded)

- The probability of $\boldsymbol{x}_i$ being assigned to the new (so far empty) cluster

$$p(\boldsymbol{z}_i = k_{new} | \mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto \alpha \times p(\boldsymbol{x}_i | G_0)$$

- The α above is due to the fact that, for the conditional prior on $\boldsymbol{z}_i$, as $K \to \infty$ we have

$$p(\boldsymbol{z}_i = k_{new} | \alpha, \mathbf{Z}_{-i}) = \frac{0 + (\alpha/K) \times (K - K_+)}{\alpha + N - 1} \to \frac{\alpha}{\alpha + N - 1}$$

Let's Have $K \to \infty$ (i.e., Unbounded)

- The probability of $\boldsymbol{x}_i$ being assigned to the new (so far empty) cluster

$$p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto \alpha \times p(\boldsymbol{x}_i|G_0)$$

- The α above is due to the fact that, for the conditional prior on $\boldsymbol{z}_i$, as $K \to \infty$ we have

$$p(\boldsymbol{z}_i = k_{new}|\alpha, \mathbf{Z}_{-i}) = \frac{0 + (\alpha/K) \times (K - K_+)}{\alpha + N - 1} \to \frac{\alpha}{\alpha + N - 1}$$

- Note: Another way - $p(\boldsymbol{z}_i = k_{new}|\alpha, \mathbf{Z}_{-i}) = 1 - \sum_{k=1}^{K_+} \frac{n_k}{\alpha+N-1} = \frac{\alpha}{\alpha+N-1}$ (since $\sum_{k=1}^{K_+} n_k = N - 1$)

Let's Have $K \to \infty$ (i.e., Unbounded)

- The probability of $\boldsymbol{x}_i$ being assigned to the new (so far empty) cluster

$$p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto \alpha \times p(\boldsymbol{x}_i|G_0)$$

- The α above is due to the fact that, for the conditional prior on $\boldsymbol{z}_i$, as $K \to \infty$ we have

$$p(\boldsymbol{z}_i = k_{new}|\alpha, \mathbf{Z}_{-i}) = \frac{0 + (\alpha/K) \times (K - K_+)}{\alpha + N - 1} \to \frac{\alpha}{\alpha + N - 1}$$

- Note: Another way - $p(\boldsymbol{z}_i = k_{new}|\alpha, \mathbf{Z}_{-i}) = 1 - \sum_{k=1}^{K_+} \frac{n_k}{\alpha+N-1} = \frac{\alpha}{\alpha+N-1}$ (since $\sum_{k=1}^{K_+} n_k = N - 1$)
- Also, instead of the likelihood, we use the marginal likelihood

$$p(\boldsymbol{x}_i|G_0) = \int p(\boldsymbol{x}_i|\phi_{new})p(\phi_{new}|G_0)d\phi_{new}$$

.. because the new cluster hasn't yet been created and thus we don't have its $\phi_{k_{new}}$

Let's Have $K \to \infty$ (i.e., Unbounded)

- The probability of $\boldsymbol{x}_i$ being assigned to the new (so far empty) cluster

$$p(\boldsymbol{z}_i = k_{new} | \mathbf{Z}_{-i}, \phi, \mathbf{X}) \propto \alpha \times p(\boldsymbol{x}_i | G_0)$$

- The α above is due to the fact that, for the conditional prior on $\boldsymbol{z}_i$, as $K \to \infty$ we have

$$p(\boldsymbol{z}_i = k_{new} | \alpha, \mathbf{Z}_{-i}) = \frac{0 + (\alpha/K) \times (K - K_+)}{\alpha + N - 1} \to \frac{\alpha}{\alpha + N - 1}$$

- Note: Another way - $p(\boldsymbol{z}_i = k_{new} | \alpha, \mathbf{Z}_{-i}) = 1 - \sum_{k=1}^{K_+} \frac{n_k}{\alpha+N-1} = \frac{\alpha}{\alpha+N-1}$ (since $\sum_{k=1}^{K_+} n_k = N - 1$)
- Also, instead of the likelihood, we use the marginal likelihood

$$p(\boldsymbol{x}_i | G_0) = \int p(\boldsymbol{x}_i | \phi_{new}) p(\phi_{new} | G_0) d\phi_{new}$$

.. because the new cluster hasn't yet been created and thus we don't have its $\phi_{k_{new}}$

- Note: Once the new cluster has been created (after a data point has been assigned to it), we also have to sample for ϕ_{new} from its posterior.

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

$$p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t)}, \mathbf{X}) \quad \propto \quad n_k^{(t-1)} \times p(\boldsymbol{x}_i|\phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K)$$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

$$\begin{aligned} p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t)}, \mathbf{X}) &\propto n_k^{(t-1)} \times p(\boldsymbol{x}_i|\phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K) \\ p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) &\propto \alpha^{(t-1)} \times p(\boldsymbol{x}_i|G_0) = \hat{\pi}_{ik_{new}} \end{aligned}$$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

$$
\begin{aligned}
p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t)}, \mathbf{X}) &\propto n_k^{(t-1)} \times p(\boldsymbol{x}_i|\phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K) \\
p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) &\propto \alpha^{(t-1)} \times p(\boldsymbol{x}_i|G_0) = \hat{\pi}_{ik_{new}} \\
\boldsymbol{z}_i^{(t)} &\sim \text{multinoulli}(\hat{\pi}_{i1}, \hat{\pi}_{i2}, \ldots, \hat{\pi}_{ik_{new}})
\end{aligned}
$$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

$$
\begin{aligned}
p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t)}, \mathbf{X}) &\propto n_k^{(t-1)} \times p(\boldsymbol{x}_i|\phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K) \\
p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) &\propto \alpha^{(t-1)} \times p(\boldsymbol{x}_i|G_0) = \hat{\pi}_{ik_{new}} \\
\boldsymbol{z}_i^{(t)} &\sim \text{multinoulli}(\hat{\pi}_{i1}, \hat{\pi}_{i2}, \ldots, \hat{\pi}_{ik_{new}}) \\
\text{set} \quad K &= K + 1 \quad (\text{if } \boldsymbol{x}_i \text{ assigned to a new cluster})
\end{aligned}
$$

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

$$
\begin{aligned}
p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t)}, \mathbf{X}) &\propto n_k^{(t-1)} \times p(\boldsymbol{x}_i|\phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K) \\
p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) &\propto \alpha^{(t-1)} \times p(\boldsymbol{x}_i|G_0) = \hat{\pi}_{ik_{new}} \\
\boldsymbol{z}_i^{(t)} &\sim \text{multinoulli}(\hat{\pi}_{i1}, \hat{\pi}_{i2}, \ldots, \hat{\pi}_{ik_{new}}) \\
\text{set} \quad K &= K + 1 \quad (\text{if } \boldsymbol{x}_i \text{ assigned to a new cluster})
\end{aligned}
$$

- Sample the mixture component parameters $\{\phi_k^{(t)}\}_{k=1}^K$ and $\alpha^{(t)}$ from the respective CPs

Gibbs Sampler for Nonparametric Bayesian Mixture Model

A brief sketch of a basic Gibbs sampler (samples $\mathbf{Z}$ and $\{\phi_k\}_{k=1}^K$) for this model with unbounded K (note: The mixing proportions π_k's were collapsed from the prior $p(\boldsymbol{z}_i|\pi)$)

Gibbs Sampler for NPBayes Mixture Model

- Set an initial K. Initialize $\mathbf{Z}^{(0)}$ and $\{\phi_k^{(0)}\}_{k=1}^K$
- For $t = 1, \ldots, T$
  - For each observation $i = 1, \ldots, N$, sample the cluster id $\boldsymbol{z}_i$

$$
\begin{aligned}
p(\boldsymbol{z}_i = k|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t)}, \mathbf{X}) &\propto n_k^{(t-1)} \times p(\boldsymbol{x}_i|\phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K) \\
p(\boldsymbol{z}_i = k_{new}|\mathbf{Z}_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) &\propto \alpha^{(t-1)} \times p(\boldsymbol{x}_i|G_0) = \hat{\pi}_{ik_{new}} \\
\boldsymbol{z}_i^{(t)} &\sim \text{multinoulli}(\hat{\pi}_{i1}, \hat{\pi}_{i2}, \ldots, \hat{\pi}_{ik_{new}}) \\
\text{set} \quad K &= K + 1 \quad (\text{if } \boldsymbol{x}_i \text{ assigned to a new cluster})
\end{aligned}
$$

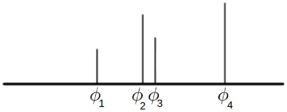
- Sample the mixture component parameters $\{\phi_k^{(t)}\}_{k=1}^K$ and $\alpha^{(t)}$ from the respective CPs

Note: "Markov Chain Sampling Methods for Dirichlet Process Mixture Models" (Neal, 2000) is an excellent reference for various MCMC sampling algorithms for nonparametric Bayesian mixture models (including collapsed versions that don't require sampling for $\{\phi_k\}_{k=1}^K$

Nonparametric Bayesian Mixture Models (A More Formal Perspective..)

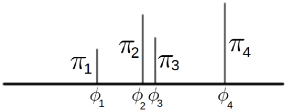
A Gentle Start: A Discrete Distribution

- Assume some space Ω (e.g., the real line) and K "locations" $\phi_1, \ldots, \phi_K$ in that space



A Gentle Start: A Discrete Distribution

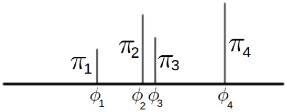
- Assume some space Ω (e.g., the real line) and K "locations" $\phi_1, \ldots, \phi_K$ in that space



- Assume these locations have "weights" $\pi_1, \ldots, \pi_K$ where $\pi_k \in (0,1), \forall k$ and $\sum_{k=1}^K \pi_k = 1$

A Gentle Start: A Discrete Distribution

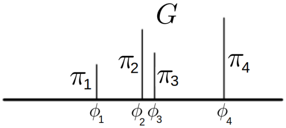
- Assume some space Ω (e.g., the real line) and K "locations" $\phi_1, \ldots, \phi_K$ in that space



- Assume these locations have "weights" $\pi_1, \ldots, \pi_K$ where $\pi_k \in (0,1), \forall k$ and $\sum_{k=1}^{K} \pi_k = 1$
  - Can think of π_k as how "popular" location ϕ_k is

A Gentle Start: A Discrete Distribution

- Assume some space Ω (e.g., the real line) and K "locations" $\phi_1, \ldots, \phi_K$ in that space



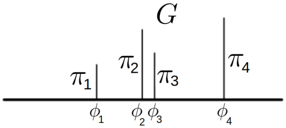
- Assume these locations have "weights" $\pi_1, \ldots, \pi_K$ where $\pi_k \in (0,1), \forall k$ and $\sum_{k=1}^K \pi_k = 1$
  - Can think of π_k as how "popular" location ϕ_k is
- Then we can define a discrete distribution G as

$$G = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$$

where δ_{ϕ_k} is an "atom" or point-mass at location ϕ_k ($\delta_{\phi_k}(\phi) = 1$ iff $\phi = \phi_k$, and 0 otherwise)

A Gentle Start: A Discrete Distribution

- Assume some space Ω (e.g., the real line) and K "locations" $\phi_1, \ldots, \phi_K$ in that space



- Assume these locations have "weights" $\pi_1, \ldots, \pi_K$ where $\pi_k \in (0,1), \forall k$ and $\sum_{k=1}^{K} \pi_k = 1$
  - Can think of π_k as how "popular" location ϕ_k is
- Then we can define a discrete distribution G as

$$G = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$$

where δ_{ϕ_k} is an "atom" or point-mass at location ϕ_k ($\delta_{\phi_k}(\phi) = 1$ iff $\phi = \phi_k$, and 0 otherwise)

- Important: The support of this discrete distribution G is $\{\phi_k\}_{k=1}^{K}$

A Bayesian Construction of G

- Let's define appropriate priors on $\{\phi_k\}_{k=1}^K$ and $\{\pi_k\}_{k=1}^K$

$$(\pi_1, \ldots, \pi_K) \quad \sim \quad \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$$

A Bayesian Construction of G

- Let's define appropriate priors on $\{\phi_k\}_{k=1}^K$ and $\{\pi_k\}_{k=1}^K$

$$
\begin{aligned}
(\pi_1, \ldots, \pi_K) \quad &\sim \quad \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k \sim G_0 \quad & \quad\quad k = 1, \ldots, K
\end{aligned}
$$

A Bayesian Construction of G

- Let's define appropriate priors on $\{\phi_k\}_{k=1}^K$ and $\{\pi_k\}_{k=1}^K$

$$\begin{aligned}(\pi_1, \ldots, \pi_K) &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\ \phi_k \sim G_0 & \qquad k = 1, \ldots, K\end{aligned}$$

- G_0 (its support being Ω) is a "base distribution" (the choice depends on what ϕ_k's are)

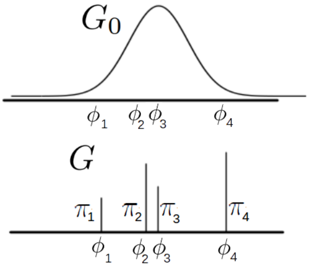
A Bayesian Construction of G

- Let's define appropriate priors on $\{\phi_k\}_{k=1}^K$ and $\{\pi_k\}_{k=1}^K$

$$
\begin{aligned}
(\pi_1, \ldots, \pi_K) \quad &\sim \quad \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k \sim G_0 \quad & \quad\quad k = 1, \ldots, K
\end{aligned}
$$

- G_0 (its support being Ω) is a "base distribution" (the choice depends on what ϕ_k's are)

$$
\begin{aligned}
(\pi_1, \ldots, \pi_K) \quad &\sim \quad \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k \sim G_0 \quad & \quad\quad k = 1, \ldots, K \\
G = \textstyle\sum_{k=1}^K \pi_k \delta_{\phi_k} &
\end{aligned}
$$



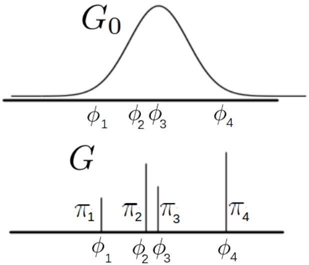
A Bayesian Construction of G

- Let's define appropriate priors on $\{\phi_k\}_{k=1}^K$ and $\{\pi_k\}_{k=1}^K$

$$\begin{aligned}(\pi_1, \ldots, \pi_K) \quad &\sim \quad \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\ \phi_k \sim G_0 \quad & \quad k = 1, \ldots, K\end{aligned}$$

- G_0 (its support being Ω) is a "base distribution" (the choice depends on what ϕ_k's are)

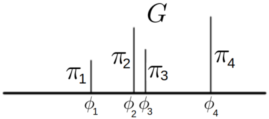
$$\begin{aligned}(\pi_1, \ldots, \pi_K) \quad &\sim \quad \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\ \phi_k \sim G_0 \quad & \quad k = 1, \ldots, K \\ G = \textstyle\sum_{k=1}^K \pi_k \delta_{\phi_k} &\end{aligned}$$



- Note that G is now a random distribution (or a random measure)

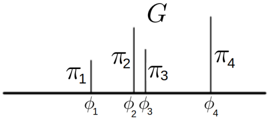
Discrete Distributions Induce Clustering!

- Drawing values repeatedly from a discrete distribution leads to repetitions



Discrete Distributions Induce Clustering!

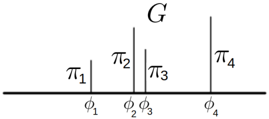
- Drawing values repeatedly from a discrete distribution leads to repetitions



- E.g., Drawing 5 times from the above G is guaranteed to have at least one repetition

Discrete Distributions Induce Clustering!

- Drawing values repeatedly from a discrete distribution leads to repetitions

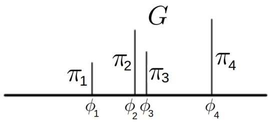


- E.g., Drawing 5 times from the above G is guaranteed to have at least one repetition
- Suppose we draw $N > K$ "parameters" $\theta_1, \ldots, \theta_N$ i.i.d. from $G = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$

$$\theta_i \sim G \qquad i = 1, \ldots, N$$

Discrete Distributions Induce Clustering!

- Drawing values repeatedly from a discrete distribution leads to repetitions



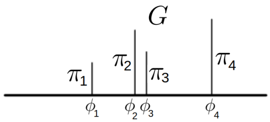
- E.g., Drawing 5 times from the above G is guaranteed to have at least one repetition
- Suppose we draw $N > K$ "parameters" $\theta_1, \ldots, \theta_N$ i.i.d. from $G = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$

$$\theta_i \sim G \qquad i = 1, \ldots, N$$

.. then the collection $(\theta_1, \ldots, \theta_N)$ will have at most K unique parameters $(\phi_1, \ldots, \phi_K)$

Discrete Distributions Induce Clustering!

- Drawing values repeatedly from a discrete distribution leads to repetitions



- E.g., Drawing 5 times from the above G is guaranteed to have at least one repetition
- Suppose we draw $N > K$ "parameters" $\theta_1, \ldots, \theta_N$ i.i.d. from $G = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$

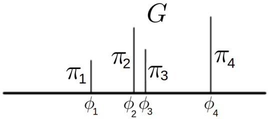
$$\theta_i \sim G \qquad i = 1, \ldots, N$$

.. then the collection $(\theta_1, \ldots, \theta_N)$ will have at most K unique parameters $(\phi_1, \ldots, \phi_K)$

- Thus G induces a clustering of parameters θ_i's, s.t. θ_i's within any group are all **identical**

Discrete Distributions Induce Clustering!

- Drawing values repeatedly from a discrete distribution leads to repetitions



- E.g., Drawing 5 times from the above G is guaranteed to have at least one repetition
- Suppose we draw $N > K$ "parameters" $\theta_1, \ldots, \theta_N$ i.i.d. from $G = \sum_{k=1}^K \pi_k \delta_{\phi_k}$

$$\theta_i \sim G \qquad i = 1, \ldots, N$$

.. then the collection $(\theta_1, \ldots, \theta_N)$ will have at most K unique parameters $(\phi_1, \ldots, \phi_K)$

- Thus G induces a clustering of parameters θ_i's, s.t. θ_i's within any group are all **identical**
- Therefore G can be used in a mixture model where θ_i's define the params of mixture distributions

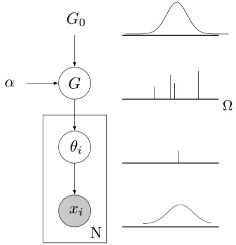
Constructing a Mixture Model for Data

- Let us use G to construct a mixture model for some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$

Constructing a Mixture Model for Data

- Let us use G to construct a mixture model for some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
- How: Generate each observation $\boldsymbol{x}_i, i = 1, \ldots, N$, assuming the following generative model

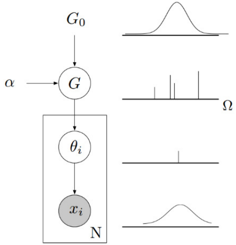
$$\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
G &= \sum_{i=1}^{K} \pi_k \delta_{\phi_k} \\
\theta_i &\sim G \\
x_i &\sim p(x|\theta_i)
\end{aligned}$$



Constructing a Mixture Model for Data

- Let us use G to construct a mixture model for some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
- How: Generate each observation $\boldsymbol{x}_i, i = 1, \ldots, N$, assuming the following generative model

$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
G &= \sum_{i=1}^{K} \pi_k \delta_{\phi_k} \\
\theta_i &\sim G \\
x_i &\sim p(x|\theta_i)
\end{aligned}
$$

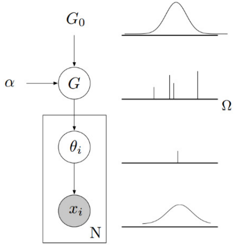


- Why drawing the parameters $\{\theta_i\}_{i=1}^N$ from G clusters the observations $\{\boldsymbol{x}_i\}_{i=1}^N$?

Constructing a Mixture Model for Data

- Let us use G to construct a mixture model for some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
- How: Generate each observation $\boldsymbol{x}_i, i = 1, \ldots, N$, assuming the following generative model

$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
G &= \sum_{i=1}^{K} \pi_k \delta_{\phi_k} \\
\theta_i &\sim G \\
x_i &\sim p(x|\theta_i)
\end{aligned}
$$

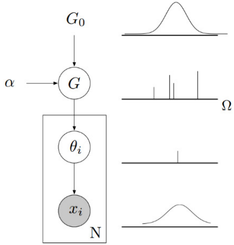


- Why drawing the parameters $\{\theta_i\}_{i=1}^N$ from G clusters the observations $\{\boldsymbol{x}_i\}_{i=1}^N$?
  - Reason: Since G is discrete, θ_i's generated by G won't be unique (only K unique values $\{\phi_k\}_{k=1}^K$)

Constructing a Mixture Model for Data

- Let us use G to construct a mixture model for some observed data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
- How: Generate each observation $\boldsymbol{x}_i, i = 1, \ldots, N$, assuming the following generative model

$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
G &= \sum_{i=1}^{K} \pi_k \delta_{\phi_k} \\
\theta_i &\sim G \\
x_i &\sim p(x|\theta_i)
\end{aligned}
$$

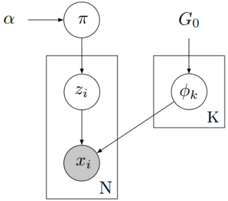


- Why drawing the parameters $\{\theta_i\}_{i=1}^N$ from G clusters the observations $\{\boldsymbol{x}_i\}_{i=1}^N$?
  - Reason: Since G is discrete, θ_i's generated by G won't be unique (only K unique values $\{\phi_k\}_{k=1}^K$)
  - Thus effectively the data is generated from not N separate distributions but only $K < N$ unique distributions $\{p(\boldsymbol{x}|\phi_k)\}_{k=1}^K$, which naturally results in a clustering of data

Two Equivalent Views

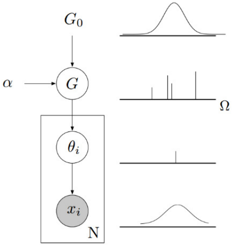
View-1 (the familiar one!): Clustering is explicitly described by the indicator $\boldsymbol{z}_i$

$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
z_i &\sim \text{multinoulli}(\pi) \\
\boldsymbol{x}_i &\sim p(x|\phi_k) \text{ if } z_i = k
\end{aligned}
$$



(Equivalent) View-2: Clustering is implicit (non-uniqueness of θ_i's denotes clustering of $\boldsymbol{x}_i$'s)

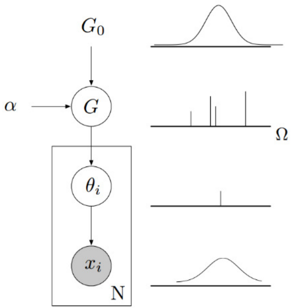
$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
G &= \sum_{i=1}^{K} \pi_k \delta_{\phi_k} \\
\theta_i &\sim G \\
x_i &\sim p(x|\theta_i)
\end{aligned}
$$



An Infinite Mixture Model

- Recall that our Bayesian mixture model construction for the finite K was

$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\
\phi_k &\sim G_0 \\
G &= \sum_{i=1}^{K} \pi_k \delta_{\phi_k} \\
\theta_i &\sim G \\
x_i &\sim p(x|\theta_i)
\end{aligned}
$$



- To get a mixture model with unbounded number of cluster, we need G of the form

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

- How can we formally construct such a G that has potentially infinite mixture components?

The Stick-Breaking Construction

- Sethuraman (1994) gave an "explicit" construction of G having infinite mixture components

The Stick-Breaking Construction

- Sethuraman (1994) gave an "explicit" construction of G having infinite mixture components

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

The Stick-Breaking Construction

- Sethuraman (1994) gave an "explicit" construction of G having infinite mixture components

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

- We basically need to generate $\{\pi_k\}_{k=1}^{\infty}$ s.t. $\pi_k \in (0,1)$ and $\sum_{k=1}^{\infty} \pi_k = 1$

The Stick-Breaking Construction

- Sethuraman (1994) gave an "explicit" construction of G having infinite mixture components

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

- We basically need to generate $\{\pi_k\}_{k=1}^{\infty}$ s.t. $\pi_k \in (0,1)$ and $\sum_{k=1}^{\infty} \pi_k = 1$
- Can be done using a stick-breaking construction for $\{\pi_k\}_{k=1}^{\infty}$ as follows

The Stick-Breaking Construction

- Sethuraman (1994) gave an "explicit" construction of G having infinite mixture components

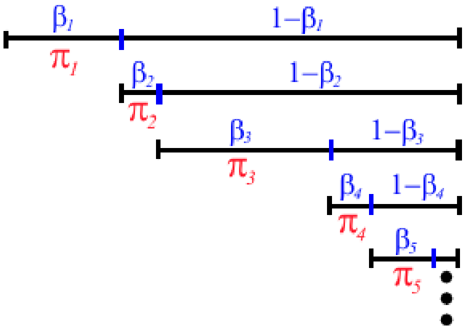
$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

- We basically need to generate $\{\pi_k\}_{k=1}^{\infty}$ s.t. $\pi_k \in (0,1)$ and $\sum_{k=1}^{\infty} \pi_k = 1$
- Can be done using a stick-breaking construction for $\{\pi_k\}_{k=1}^{\infty}$ as follows

$$\begin{aligned}
\beta_k &\sim \text{Beta}(1,\alpha) \qquad k = 1,\ldots,\infty \\
\pi_1 &= \beta_1 \\
\pi_k &= \beta_k \prod_{\ell=1}^{k-1} (1-\beta_{k-1}) \qquad k = 2,\ldots,\infty
\end{aligned}$$

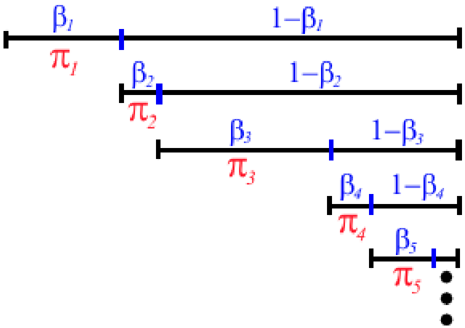
The Stick-Breaking Construction

- Assume a stick of length 1 to begin with. Now recursively break it as follows:
  - Choose a random location $\beta_k \in (0,1)$ drawn from $\text{Beta}(1,\alpha)$ at which to break the stick
  - Record π_k as "β_k times the length of the remaining stick"



The Stick-Breaking Construction

- Assume a stick of length 1 to begin with. Now recursively break it as follows:
  - Choose a random location $\beta_k \in (0, 1)$ drawn from $\text{Beta}(1, \alpha)$ at which to break the stick
  - Record π_k as "β_k times the length of the remaining stick"



- Can show that $\sum_{k=1}^{K} \pi_k = 1$ as $K \to \infty$. One easy way to verify this is by showing that $1 - \sum_{k=1}^{K} \pi_k \to 0$ as $K \to \infty$

Next Class

- Some other (equivalent) ways of looking at nonparametric Bayesian mixture models
  - Dirichlet Process
  - Chinese Restaurant Process
  - Pólya-Urn Scheme
  - Hierarchical Dirichlet Process
- Some other examples of nonparametric Bayesian models