UPPSALA
UNIVERSITET

Guest lecture:
MCMC with Discrete Parameters

Jakob Torgander

UPPSALA UNIVERSITET

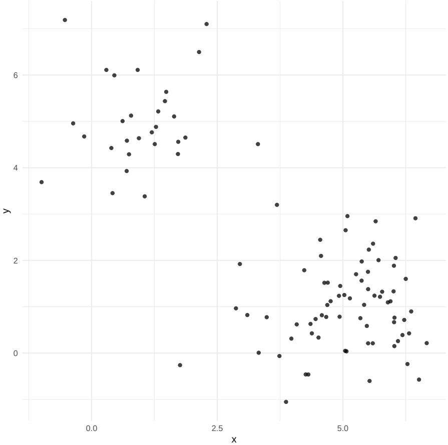
Outline

1. Discrete parameters - Introduction & discussion
2. Describe three methods for computing posteriors with discrete latent parameters
   - Marginalization
   - Gibbs sampling
   - Continuous approximation using Gumbel-Softmax-distribution
3. (Short) demonstration of methods.

Motivation

- Discrete variables are everywhere
  - Count data: e.g. number of car accidents
  - Categorical data
  - Decision/classification problems: (eg. yes/no)
  - Factor analysis
- In many problems latent (hidden) discrete variables exists: conclusions changes if data is segmented into groups
- While current state-of-the-art method Hamiltonian Monte Carlo (HMC) works for discrete *data* HMC does not directly work for discrete *parameters*.

Case study - Gaussian mixture model



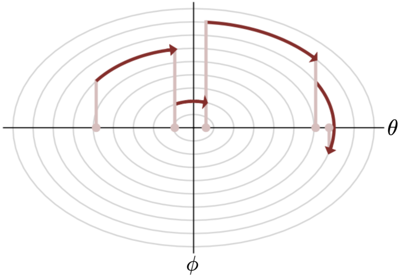
- Latent class variable C
- $\text{p(y)} = \sum_{k=1}^{K} \mathbb{1}(C = k)\mathcal{N}(y|\mu_k, \sigma_k)$,
- Task: identify cluster assignments, probabilities and centers

UPPSALA UNIVERSITET

Section 2

Computation

Hamiltonian Monte Carlo (HMC) - Recap



(?, ?): Given a current parameter-momentum pair (θ_i, ϕ_i), Hamiltonian H and mass matrix M:

1. Sample a new momentum variable $\phi_{i+1} \sim \mathcal{N}(0, M)$
2. Lift θ_i onto the joint phase space (θ, ϕ)
3. Integrate the flow defined by $H(\theta_i, \phi_{i+1}) = \text{constant}$ using Hamilton's equations
4. Project back to original parameter space to receive new parameter sample θ_{i+1}

Recap: the leapfrog integrator

Step 3 of HMC is based on the leapfrog algorithm

$$
\begin{aligned}
\psi &\leftarrow \psi + \frac{1}{2}\epsilon \frac{d \log q(\theta|y)}{d\theta} && \textit{1st momentum update} \\
\theta &\leftarrow \theta + \epsilon M^{-1}\psi && \textit{Parameter update} \\
\psi &\leftarrow \psi + \frac{1}{2}\epsilon \frac{d \log q(\theta|y)}{d\theta} && \textit{2nd momentum update},
\end{aligned}
$$

where q denotes the target posterior density.

UPPSALA
UNIVERSITET

Recap: the leapfrog integrator

Step 3 of HMC is based on the leapfrog algorithm

$$
\begin{aligned}
\psi &\leftarrow \psi + \frac{1}{2}\epsilon \frac{d \log q(\theta|y)}{d\theta} && \textit{1st momentum update} \\
\theta &\leftarrow \theta + \epsilon M^{-1}\psi && \textit{Parameter update} \\
\psi &\leftarrow \psi + \frac{1}{2}\epsilon \frac{d \log q(\theta|y)}{d\theta} && \textit{2nd momentum update},
\end{aligned}
$$

where q denotes the target posterior density. Q: Why does this fail when q is discrete?

HMC does not work for discrete posteriors

Main problem: Computation of the gradient $\frac{d \log q(\theta|y)}{d\theta}$ requires the limits (partial derivatives)

$$\frac{\partial q}{\partial \theta_i} = \lim_{h \to 0} \frac{q(\theta + h\boldsymbol{e}_i) - q(\theta)}{h}$$

to exist. This only happens when q is continuous!

Method 1: Marginalization

UPPSALA UNIVERSITET

Idea: Sum (marginalize) out the latent discrete parameters (?, ?).
By the law of total probability:

$$p(y) = \sum_{k=1}^{K} p(y|c_k)p(c_k).$$

Then, $p(y)$ is continuous if $p(y|c_k), p(c_k)$ are.

Marginalization

Example: for the GM-model:

$$p_Y(y, |\pi, \mu, \sigma) = \sum_{k=1}^{K} \underbrace{\pi_k}_{p(c_k)} \underbrace{\mathcal{N}(y|\mu_k, \sigma_k)}_{p(y|c_k)},$$

where π_k are (continuous) parameters.

Marginalization

UPPSALA UNIVERSITET

Example: for the GM-model:

$$p_Y(y, |\pi, \mu, \sigma) = \sum_{k=1}^{K} \underbrace{\pi_k}_{p(c_k)} \underbrace{\mathcal{N}(y|\mu_k, \sigma_k)}_{p(y|c_k)},$$

where π_k are (continuous) parameters. Remark: Compare with the original model formulation

$$p(y|\pi, \mu, \sigma) = \sum_{k=1}^{K} \mathbb{1}(C = k)\mathcal{N}(y|\mu_k, \sigma_k)$$

Method 2: Gibbs sampling

Recall: Gibbs sampling: conditional (or block) sampling of θ

$$\theta_j \sim p(\theta_j | \theta_{-j}, y)$$

Method 2: Gibbs sampling

Recall: Gibbs sampling: conditional (or block) sampling of θ

$$\theta_j \sim p(\theta_j|\theta_{-j}, y)$$

For the GM-model, with σ known:

1. For each observation y, sample classes c_i with probability

$$p(c_i|\mu, \sigma, y) = \frac{p(y|c_i)p(c_i)}{\sum_{j=1}^{K} p(y|c_j)p(c_j)} = \frac{p(c_i)\mathcal{N}(y|\mu_i, \sigma)}{\sum_{j=1}^{K} p(c_j)\mathcal{N}(y|\mu_j, \sigma)}$$

UPPSALA UNIVERSITET

Method 2: Gibbs sampling

Recall: Gibbs sampling: conditional (or block) sampling of θ

$$\theta_j \sim p(\theta_j | \theta_{-j}, y)$$

For the GM-model, with σ known:

1. For each observation y, sample classes c_i with probability

$$p(c_i | \mu, \sigma, y) = \frac{p(y|c_i)p(c_i)}{\sum_{j=1}^{K} p(y|c_j)p(c_j)} = \frac{p(c_i)\mathcal{N}(y|\mu_i, \sigma)}{\sum_{j=1}^{K} p(c_j)\mathcal{N}(y|\mu_j, \sigma)}$$

2. Sample means μ_i using the conditional distributions $p(\mu_i | y, c_i)$ (normal if likelihood and prior for μ is)

UPPSALA UNIVERSITET

Method 3: Continuous approximation Gumbel-Softmax

Ideas:

- Approximate a discrete (categorical) distribution with a continuous distribution.
- The approximated distribution can then be used with HMC
- Use the "Gumbel trick" (?, ?) from the field of deep learning

"The Gumbel trick"

Proposition: Let Z be a categorical r.w with probability distribution $\pi = (\pi_1, \ldots, \pi_K)$ and let G_i be Gumbel(0,1)-distributed with density

$$f_{G_i} = e^{-x-e^{-x}}.$$

Then the random variable

$$U = \arg\max_i G_i + \log \pi_i,$$

follows the same distribution as Z.

"The Gumbel trick"

Proposition: Let Z be a categorical r.w with probability distribution $\pi = (\pi_1, \ldots, \pi_K)$ and let G_i be Gumbel(0,1)-distributed with density

$$f_{G_i} = e^{-x-e^{-x}}.$$

Then the random variable

$$U = \arg\max_i G_i + \log \pi_i,$$

follows the same distribution as Z.Q: How to use the argmax-function in a density?

UPPSALA UNIVERSITET

Gumbel-Softmax distribution

Idea by ? (?): Approximate argmax with the softmax function.

$$Y_i = \frac{\exp((\log(\pi_i) + G_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + G_j)/\tau)},$$

where G_i are Gumbel(0,1)-distributed and τ is a "temperature" parameter.

UPPSALA UNIVERSITET

Gumbel-Softmax distribution

Idea by ? (?): Approximate argmax with the softmax function.

$$Y_i = \frac{\exp((\log(\pi_i) + G_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + G_j)/\tau)},$$

where G_i are Gumbel(0,1)-distributed and τ is a "temperature" parameter.

As τ approaches 0, $Y = (Y_1, \ldots, Y_K)$ then tends to a "one-hot" vector on the form

$$[0, \ldots, 0, 1, 0, \ldots, 0],$$

where a "1" in position m indicates the m-th class.

Gumbel-Softmax distribution

This yields the Gumbel-Softmax (GS) density function:

$$p_{\pi,\tau}(y_1,\ldots,y_K) = (K-1)! \cdot \tau^{K-1} \left(\sum_{i=1}^{K} \pi_i / y_i^{\tau} \right)^{-k} \prod (\pi_i / y_i^{\tau+1}).$$

Continuous! Can hence be used with HMC and Stan.

UPPSALA UNIVERSITET

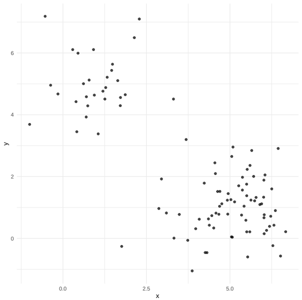
Methods - Summary

Method	Pros	Cons
Marginalization	Works efficiently with HMC	Does not return the discrete parameter
Gibbs	Returns classes, reliable for "simple" distributions	Difficult (& sometimes less efficient) for non-conjugate distributions
Gumbel-Softmax.	Returns classes, works with HMC	High dependency on temperature τ, leapfrog (very) unstable for low temperatures

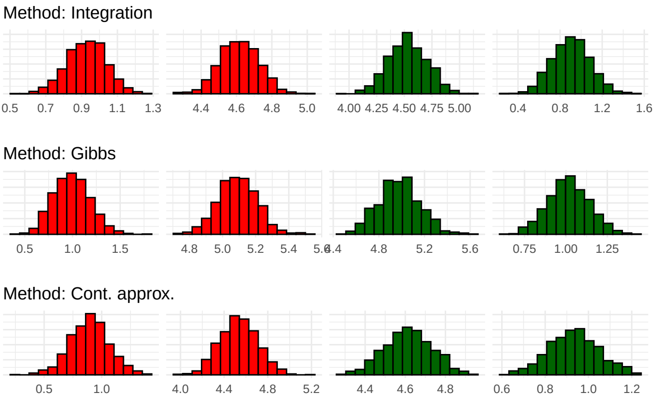
UPPSALA
UNIVERSITET

Section 3

Demonstration

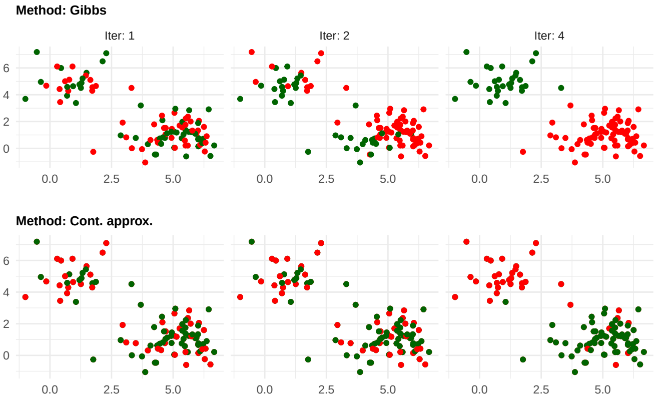


- Data: simulated gaussian mixtures with means $\mu_1 = (1, 5), \mu_2 = (5, 1)$ and $\sigma_1 = \sigma_2 = \mathbf{I}$
- Weakly informative $\mathcal{N}(0, 10)$-prior used for all μ
- Dirichlet(1,1)-prior (see e.g. (?, ?, p. 69)) used for HMC methods (1 and 3)
- 2000 samples generated for each method



- All three methods correctly identify the centers (red $= \mu_1$, green $= \mu_2$)
- Gibbs sampler closer to "ground truth" in this case
- Difference possible due to weakly informative Dirichlet prior of Method 1,3. Needs further investigation..

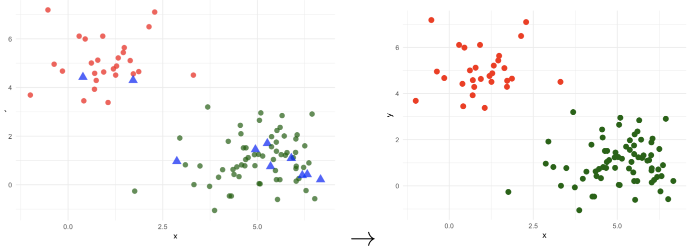
Class assignments - convergence



- Both Gibbs and Continuous approx. converges quickly to the correct classes
- Gibbs sampler one iteration quicker

UPPSALA
UNIVERSITET

Use case: imputing missing values



- Gibbs, and Gumbel-Softmax method can be used to impute missing values (classes)
- Idea: Generate class parameter if non-present in the data and use the actual class otherwise
- For general tips about handling missing values, see (?, ?)

UPPSALA UNIVERSITET

Future research

- How do the methods scale with data size and dimension?
- How can τ in the GS-approximation be selected and tuned?
- Performance and convergence of methods on more complicated, high-dimensional posteriors?

UPPSALA
UNIVERSITET

Thank you!

References