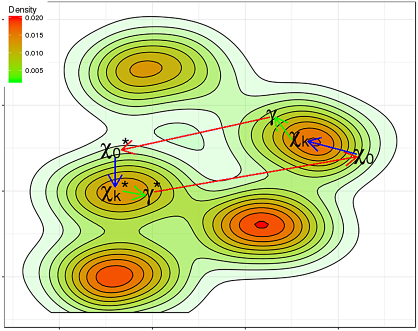
Additional topics on Monte Carlo

Geir Storvik

Geilo Winter school 2023

Outline

- MCMC: Work reasonable well for unimodal distributions
    - Struggle more with multimodal distributions



- Several possible approaches:
    - Simulated tempering: Use $\pi_k(\boldsymbol{x}) = \pi_k(\boldsymbol{x})^{T_k}$, $T_k \leq 1$, move between differet "models"
    - SMC: Similar sequence
- Here: Mode jumping MCMC Tjelmeland and Hegstad (2001)

Mode jumping MCMC

- Aim: Allow for large changes
- Main problem: Large move in space will typically result in low density value
  - M-H: Very low acceptance rate
- Main idea
  1. Make a large change $\boldsymbol{x} \rightarrow \boldsymbol{x}_0^*$
  2. Perform a local optimization $\boldsymbol{x}_0^* \rightarrow \boldsymbol{x}_k^*$
     - Possibly through k steps of some optimization routine
  3. Small perturbation: $\boldsymbol{x}_k^* \rightarrow \boldsymbol{x}^*$
  4. Accept $\boldsymbol{x}^*$ through an M-H step
- M-H: Detailed balance require possibility for moving backwards as well
  - The small perturbation in step 3 makes this possible

Algorithm

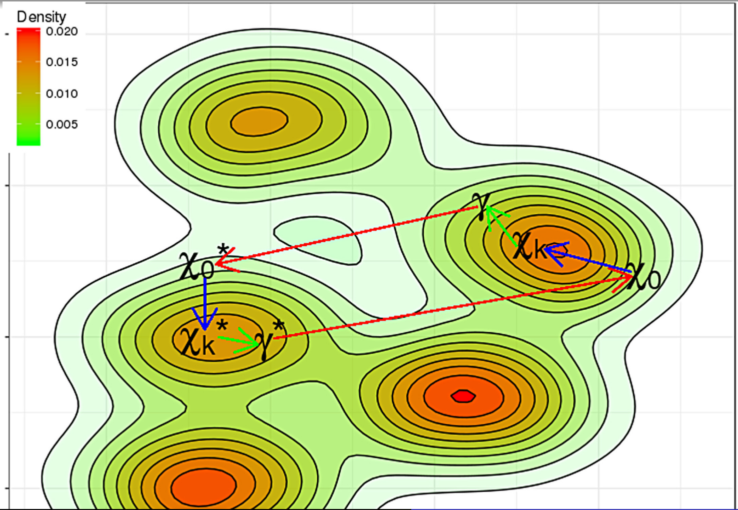
Algorithm 1 MJMCMC step from current state $\boldsymbol{x}$

1: Generate $\boldsymbol{x}_0^* = \boldsymbol{x}^* + \varepsilon^*$, $\varepsilon^* \sim N(\mathbf{0}, \sigma_L^2 \boldsymbol{R})$, σ_L large
2: Optimize $\boldsymbol{x}_0^* \rightarrow \boldsymbol{x}_k^*$
3: Small perturbation: $\boldsymbol{x}^* \sim g_S(\boldsymbol{x}^* | \boldsymbol{x}_k^*)$
4: Generate $\boldsymbol{x}_0 = \boldsymbol{x}^* - \varepsilon^*$
5: Optimize $\boldsymbol{x}_0 \rightarrow \boldsymbol{x}_k$
6: Calculate

$$r = \frac{\pi(\boldsymbol{x}^*) q_r(\boldsymbol{x} | \boldsymbol{x}_k)}{\pi(\boldsymbol{x}) q_r(\boldsymbol{x}^* | \boldsymbol{x}_k^*)}$$

7: Accept $\boldsymbol{x}^*$ with probability $\min\{1, r\}$

MJMCMC - graphical illustration



MCMCMC for model selection

- Consider a model

$$\begin{aligned} y_i \sim & f(y_i; \eta_i, \phi) \\ \eta_i = & \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j z_{i,j} \\ \gamma_j \sim & \text{Bern}(q) \\ \beta_j | \gamma_j = 1 \sim & N(0, \sigma_\beta^2) \end{aligned}$$

- Aim: $p(\boldsymbol{\gamma}|\boldsymbol{y})$.
- 2^p possible models, in addtion unknown β_j's
- Possible: $p(\boldsymbol{\gamma}, \boldsymbol{\beta}|\boldsymbol{y})$ through Reversible jump MCMC
- Pseudo-Marginal MCMC:
  1. Generate proposal $\boldsymbol{\gamma}^*$ from $\boldsymbol{\gamma}$
  2. Accept $\boldsymbol{\gamma}^*$ with probability $\min\{1, r\}$ where

$$r = \frac{p(\boldsymbol{\gamma}^*|\boldsymbol{y})g(\boldsymbol{\gamma}|\boldsymbol{\gamma}^*)}{p(\boldsymbol{\gamma}|\boldsymbol{y})g(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})}$$

- Hubin and Storvik (2018): Linear models
- Hubin et al. (2021): Neural network type modes

Particle MCMC

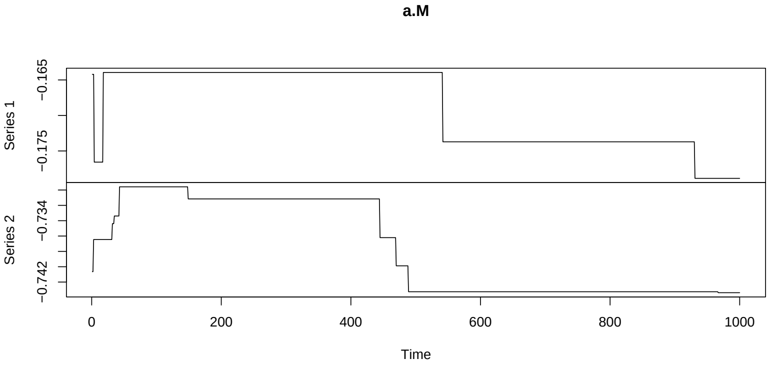
- Andrieu et al. (2010)
- Ideal MCMC ($p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\boldsymbol{\theta})L(\boldsymbol{\theta})$):
  1. Sample $\boldsymbol{\theta}^* \sim g(\boldsymbol{\theta}^*|\boldsymbol{\theta})$
  2. Calculate M-H ratio $r = \frac{p(\boldsymbol{\theta}^*)L(\boldsymbol{\theta}^*)g(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{p(\boldsymbol{\theta})L(\boldsymbol{\theta})g(\boldsymbol{\theta}^*|\boldsymbol{\theta})}$
  3. Accept $\boldsymbol{\theta}^*$ with prob $\min\{1, r\}$

Particle MCMC

- Andrieu et al. (2010)
- Ideal MCMC ($p(\theta|\mathbf{y}) \propto p(\theta)L(\theta)$):
  1. Sample $\theta^* \sim g(\theta^*|\theta)$
  2. Calculate M-H ratio $r = \frac{p(\theta^*)L(\theta^*)g(\theta|\theta^*)}{p(\theta)L(\theta)g(\theta^*|\theta)}$
  3. Accept θ^* with prob $\min\{1, r\}$
- Pseudo-Marginal algorithm:
  1. Sample $\theta^* \sim g(\theta^*|\theta)$
  2. Calculate $\hat{L}(\theta^*)$
  3. Calculate M-H ratio $\hat{r} = \frac{\pi(\theta^*)p(\theta|\theta^*)}{\pi(\theta)p(\theta^*|\theta)}$
  4. Accept θ^* with prob $\min\{1, \hat{r}\}$
- Particle MCMC: Use SMC to calculate $\hat{L}(\theta^*)$

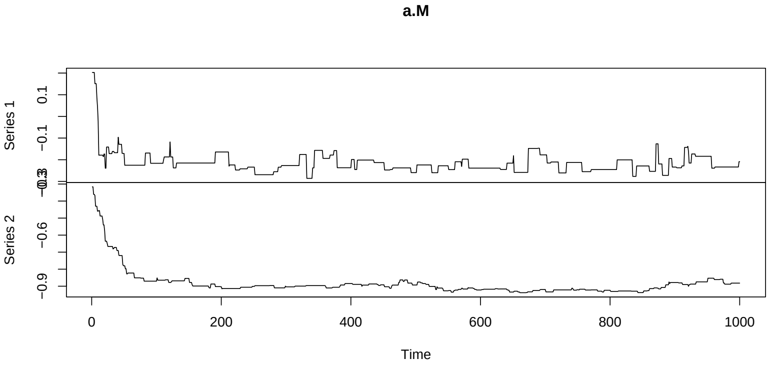
Lemmings - AR(2) process

Now: $x_t = a_1 x_{t-1} + a_2 x_{t-2} + \varepsilon_t$



Lemmings - AR(2) process

Now: $x_t = a_1 x_{t-1} + a_2 x_{t-2} + \varepsilon_t$



Reversible jump MCMC

- Examples of changing dimensions
  - $Y_i = \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j x_{ij} + \varepsilon_i$
  - Neural networks with some weights put to zero.
- Reversible Jump MCMC
  - Assume several models $\mathcal{M}_1, ..., \mathcal{M}_K$
  - Corresponding parameters $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$ of different dimensions!
  - Aim: Simulate $\boldsymbol{x} = (\mathcal{M}, \boldsymbol{\theta}_{\mathcal{M}})$
  - RJMCMC: Green (1995)
  - RJMCMC: M-H method for moving between spaces of different dimensions
  - Main challenge: When changing $\mathcal{M} \rightarrow \mathcal{M}^*$, how to propose $\boldsymbol{\theta}_{\mathcal{M}^*}$?

Reversible jump MCMC

- Examples of changing dimensions
  - $Y_i = \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j x_{ij} + \varepsilon_i$
  - Neural networks with some weights put to zero.
- Reversible Jump MCMC
  - Assume several models $\mathcal{M}_1, ..., \mathcal{M}_K$
  - Corresponding parameters $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$ of different dimensions!
  - Aim: Simulate $\boldsymbol{x} = (\mathcal{M}, \boldsymbol{\theta}_{\mathcal{M}})$
  - RJMCMC: Green (1995)
  - RJMCMC: M-H method for moving between spaces of different dimensions
  - Main challenge: When changing $\mathcal{M} \to \mathcal{M}^*$, how to propose $\boldsymbol{\theta}_{\mathcal{M}^*}$?

Changing dimensions

- Assume several models $\mathcal{M}_1, ..., \mathcal{M}_K$
- Corresponding parameters $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$ of different dimensions!
- Aim: Simulate $\boldsymbol{x} = (\mathcal{M}, \boldsymbol{\theta}_{\mathcal{M}})$
- RJMCMC: M-H method for moving between spaces of different dimensions
- Main challenges:
  - When changing dimensions, how to compare densities on different spaces?
  - When changing $\mathcal{M} \rightarrow \mathcal{M}^*$, how to propose $\boldsymbol{\theta}_{\mathcal{M}^*}$?

Reversible jump MCMC

- Green (1995): Include auxiliary variables to match dimensions.
- Consider change $(\mathcal{M}_1, \boldsymbol{\theta}_1)$ to $(\mathcal{M}_2, \boldsymbol{\theta}_2)$ with $|\boldsymbol{\theta}_1| < |\boldsymbol{\theta}_2|$
  - $j(1 \to 2)$ probability for moving from $\mathcal{M}_1$ to $\mathcal{M}_2$
- Algorithm
  1. Generate $\boldsymbol{u}_1$ such that $|\boldsymbol{\theta}_1| + |\boldsymbol{u}_1| = |\boldsymbol{\theta}_2|$
  2. Propose $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_2(\boldsymbol{\theta}_1, \boldsymbol{u}_1)$ (bijective)
  3. Calculate acceptance ratio

$$r = \frac{\pi(\mathcal{M}_2, \boldsymbol{\theta}_2) q(2 \to 1)}{\pi(\mathcal{M}_1, \boldsymbol{\theta}_1) q(1 \to 2) q(\boldsymbol{u}_1)} \left| \frac{\partial(\boldsymbol{\theta}_2)}{\partial(\boldsymbol{\theta}_1, \boldsymbol{u}_1)} \right|$$

  4. Accept with probability $\min\{1, r\}$.
- Use $1/r$ for opposite move
- More general settings possible

Logistic regression

- Assume model

$$Y_i \sim \text{Binom}(p_i) \qquad \text{logit}(p_i) = \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j x_{ij}$$

$$\Pr(\gamma_j = 1) = q \qquad \beta_j | \gamma_j = 1 \sim N(0, \sigma_\beta^2)$$

- Assume $\gamma_j = 0$, want to change to $\gamma_j^* = 1$
- Generate $u_1 \sim g_j()$
- Put

$$\beta_k^* = \begin{cases} \beta_k & k \neq j; \\ u_1 & k = j. \end{cases}$$

- Accept with probabilty $\min\{1, r\}$ where

$$r = \frac{\pi(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)}{\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}) g_j(\beta_j^*)} \times 1$$

- Script `Log_reg_RJ.R`

Non-reversible MCMC

- Main criterion (π-invariance)

$$\pi(\boldsymbol{x}^*) = \int_{\boldsymbol{x}} \pi(\boldsymbol{x}) P(\boldsymbol{x}^* | \boldsymbol{x}) d\boldsymbol{x}$$

- Sufficient criterion for stationarity

$$\pi(\boldsymbol{x}) P(\boldsymbol{x}^* | \boldsymbol{x}) = \pi(\boldsymbol{x}^*) P(\boldsymbol{x} | \boldsymbol{x}^*) \quad \text{Detailed balance}$$

- Results in a reversible MCMC (moving backwards is similar to moving forwards)

Non-reversible MCMC

- Main criterion (π-invariance)

$$\pi(\boldsymbol{x}^*) = \int_{\boldsymbol{x}} \pi(\boldsymbol{x}) P(\boldsymbol{x}^*|\boldsymbol{x}) d\boldsymbol{x}$$

- Sufficient criterion for stationarity

$$\pi(\boldsymbol{x}) P(\boldsymbol{x}^*|\boldsymbol{x}) = \pi(\boldsymbol{x}^*) P(\boldsymbol{x}|\boldsymbol{x}^*) \quad \text{Detailed balance}$$

- Results in a reversible MCMC (moving backwards is similar to moving forwards)
- Assume now $x \in \mathcal{Z}$
- Introduce $v \in \{-1, 1\}$ and consider extended distribution $\bar{\pi}(x, v) = 0.5\pi(x) I(v \in \{-1, 1\})$.
- Define Markov chain

$$P(x^*, v|x, w) = \alpha(x, v) I(x^* = x+v, w = v) + (1-\alpha(x, v)) I(x^* = x, w = -v)$$

with $\alpha(x, v) = \min\{1, \pi(x + v)/\pi(x)\}$

The zig-zag process

- Continuous-time Markov process
- Can use sub-sampling with an exact approximate scheme
- Can be super-efficient when combined with control-covariate ideas
- References: Bierkens et al. (2019) (and references therein)
- Main idea:
  - Move all components lineary in a given direction: $x_i(t) = x_i^k + z_i^k t$
  - Change direction of z_i^k at random (continuous) time points

Algorithm

Let $(T^0, \boldsymbol{x}^0, \theta^0) = (0, \xi, \theta)$
for $k = 1, 2, \cdots$ **do**
 Let $\boldsymbol{\xi}^k(t) \equiv \boldsymbol{x}^{k-1} + \boldsymbol{\theta}^{k-1} t, t \geq 0$
 For $i = 1, ..., p$, let τ_i^k be distributed according to

$$\Pr(\tau_i^k \geq t) = \exp\left(-\int_0^t \lambda_i(\boldsymbol{\xi}^k(s), \boldsymbol{z}^{k-1}) ds\right)$$

 Let $i_0 \equiv \arg\min_{i \in \{1,...,p\}} \tau_i^k$
 Let $T^k \equiv T^{k-1} + \tau_{i_0}^k$
 Let $\boldsymbol{x}^k \equiv \boldsymbol{\xi}^k(T^k)$
 Let

$$z_i^k = \begin{cases} z_i^{k-1} & \text{if } i \neq i_0 \\ -z_i^{k-1} & \text{if } i = i_0 \end{cases}$$

end for

Trajectories

- Piecewise deterministic trajectories $(\boldsymbol{x}(t), \theta(t))$:

$$(\boldsymbol{x}(t), \theta(t)) = (\boldsymbol{x}^k + \boldsymbol{z}^k(t - T^k), \boldsymbol{z}^k) \quad \text{for } t \in [T^k, T^{k+1}), k = 0, 1, 2 \cdots$$

- Monte Carlo estimate:

$$\begin{aligned}
\hat{\mu}_i =& \frac{1}{T}\int_0^T x_i(t)dt \\
=& \frac{1}{T}\sum_{k=0}^{K}\int_{T^k}^{T^{k+1}} [x_i^k + z_i^k(t - T^k)]dt \\
=& x_i^k(T^{k+1} - T^k) + 0.5 z_i^k (T^{k+1} - T^k)^2
\end{aligned}$$

What does it converge to?

- Distribution depending om the functions $\lambda_i(\xi, \boldsymbol{z})$.
- Assume $\theta_i \in \{-1, 1\}$
- Assume a Bayesian setting:

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})$$

- Define

$$\begin{aligned}\Psi(\boldsymbol{x}) &= -\log p(\boldsymbol{x}) - \log p(\boldsymbol{y}|\boldsymbol{x})\\ \lambda_i(\boldsymbol{x}, \boldsymbol{\theta}) &= (\theta_i \partial_i \Psi(\boldsymbol{x}))^+ + \gamma_i(\boldsymbol{x}, \boldsymbol{\theta})\end{aligned}$$

  where γ_i is non-negative and $\gamma_i(\boldsymbol{x}, \boldsymbol{\theta}) = \gamma_i(\boldsymbol{x}, \boldsymbol{\theta}_{-i})$ with $\boldsymbol{\theta}_{-i}$ is equal to $\boldsymbol{\theta}$ except for the ith component which is flipped.
- Then (under some regularity conditions)
  - The Zig-Zag process has $p(\boldsymbol{x}|\boldsymbol{y})$ as invariant distribution
  - The process is ergodic:

$$\lim_{t\to\infty} \int_0^t f(\boldsymbol{x}(s))ds = \int f(\boldsymbol{x})\pi(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x}$$

Additional topics in MCMC

- Adaptive MCMC: Automatic tuning of proposal distributions
  - Main challenge: Specifying proposal based on history of chain breaks down the Markov property
  - Solution: Reduce the amount of tuning as the number of iterations increases

Additional topics in MCMC

- Adaptive MCMC: Automatic tuning of proposal distributions
  - Main challenge: Specifying proposal based on history of chain breaks down the Markov property
  - Solution: Reduce the amount of tuning as the number of iterations increases
- Simulated tempering
  - Define $f^i(\boldsymbol{x}) \propto \pi(\boldsymbol{x})^{1/\tau_i}$, $1 = \tau_1 < \tau_2 < \cdots < \tau_m$
  - Simulate $(\boldsymbol{x}, I)$, where I changes distribution
  - Easier to move around when $\tau_i > 1$
  - Keep samples for which $I = 1$

Additional topics in MCMC

- Adaptive MCMC: Automatic tuning of proposal distributions
  - Main challenge: Specifying proposal based on history of chain breaks down the Markov property
  - Solution: Reduce the amount of tuning as the number of iterations increases
- Simulated tempering
  - Define $f^i(\boldsymbol{x}) \propto \pi(\boldsymbol{x})^{1/\tau_i}$, $1 = \tau_1 < \tau_2 < \cdots < \tau_m$
  - Simulate $(\boldsymbol{x}, I)$, where I changes distribution
  - Easier to move around when $\tau_i > 1$
  - Keep samples for which $I = 1$
- Multiple-Try M-H
  - Generate k proposals $\boldsymbol{x}_1^*, ..., \boldsymbol{x}_k^*$ from $g(\cdot|\boldsymbol{x}^{(t)})$
  - Select $\boldsymbol{x}_j^*$ with probability $w(\boldsymbol{x}^{(t)}, \boldsymbol{x}_j^*) = \pi(\boldsymbol{x}^{(t)})g(\boldsymbol{x}_j^*|\boldsymbol{x}^{(t)})\lambda(\boldsymbol{x}^{(t)}, \boldsymbol{x}_j^*)$, λ symmetric
  - Sample $\boldsymbol{x}_1^{**}, ..., \boldsymbol{x}_{k-1}^{**}$ from $g(\cdot|\boldsymbol{x}_j^*)$, put $\boldsymbol{x}_k^{**} = \boldsymbol{x}^{(t)}$
  - Use Generalized M-H ratio

$$R_g = \frac{\sum_{i=1}^k w(\boldsymbol{x}^{(t)}, \boldsymbol{x}_i^*)}{\sum_{i=1}^k w(\boldsymbol{x}_j^*, \boldsymbol{x}_i^{**})}$$

Implement your SMC algorithm

- Consider the model for lemmings data
- Model

$$\begin{aligned} \boldsymbol{y}_t \sim& \text{Binom}\left(1, \tfrac{\exp(\boldsymbol{x}_t)}{1+\exp(\boldsymbol{x}_t)}\right) \\ \boldsymbol{x}_t =& a\boldsymbol{x}_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2) \\ a \sim& \text{Uniform}[0, 1] \end{aligned}$$

  Use $a = 0.5, \sigma = 1$

  - Some data are missing: How to handle this?

Implement your MCMC algorithm

- Consider the model for lemmings data
- Model

$$\begin{aligned}\boldsymbol{y}_t \sim& \text{Binom}\left(1, \frac{\exp(\boldsymbol{x}_t)}{1+\exp(\boldsymbol{x}_t)}\right)\\ \boldsymbol{x}_t =& a\boldsymbol{x}_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0,\sigma^2)\\ a \sim& \text{Uniform}[0,1]\end{aligned}$$

  Use $a = 0.5, \sigma = 1$
- Note: if only changing $x_t \to x_t^*$:

$$\frac{\pi(\boldsymbol{x}^*)}{\pi(\boldsymbol{x})} = \frac{p(x_t^*|x_{t-1})p(x_{t+1}|x_t^*)p(y_t|x_t^*)}{p(x_t|x_{t-1})p(x_{t+1}|x_t)p(y_t|x_t)}$$

References

C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

J. Bierkens, P. Fearnhead, and G. Roberts. The zig-zag process and super-efficient sampling for bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320, 2019.

P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

A. Hubin and G. Storvik. Mode jumping mcmc for bayesian variable selection in glmm. *Computational Statistics & Data Analysis*, 127:281–297, 2018.

A. Hubin, G. Storvik, and F. Frommlet. Flexible bayesian nonlinear model configuration. *Journal of Artificial Intelligence Research*, 72:901–942, 2021.

H. Tjelmeland and B. K. Hegstad. Mode jumping proposals in mcmc. *Scandinavian journal of statistics*, 28(1):205–223, 2001.