

Probabilistic Machine Learning

Lecture 1 | Fundamentals of Probabilistic Modeling

Amirabbas Asadi

amir.asadi78@sharif.edu

2024

Outline

- Probabilistic (generative) models
- Bayesian Learning
- Parametric Probabilistic models
- Nonparametric Probabilistic models
- Bayesian Inference
 - MCMC Inference
 - Variational Inference
 - Particle-Based Inference
- Scaling up probabilistic inference for real-world scenarios
- Discussed examples:
 - Probabilistic models for Time Series
 - Probabilistic models for Reinforcement Learning

Probabilistic Models

What is a probabilistic models?

Probabilistic Models

What is a probabilistic models?

$$p(X_1, X_2, \dots, X_n)$$

Any representation of a joint distribution is a probabilistic model

Probabilistic Models

Why are we interested in designing probabilistic models?

Applications of Probabilistic Models

Sampling

$$x_1, x_2, \dots, x_n \sim p(X_1, X_2, \dots, X_n)$$

Applications of Probabilistic Models

Sampling

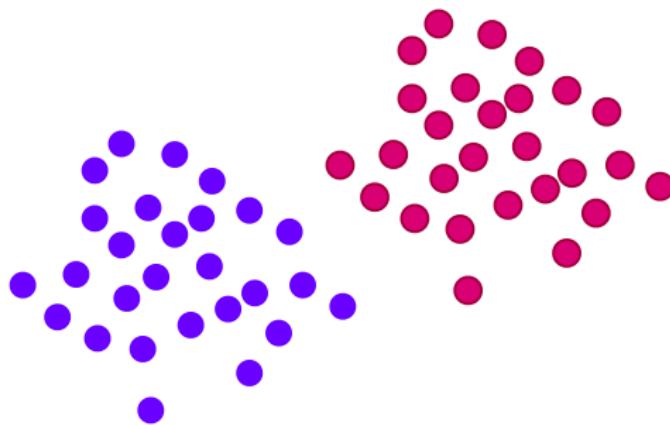
$$x_1, x_2, \dots, x_n \sim p(X_1, X_2, \dots, X_n)$$

Probabilistic Inference

$$p(Z, X_1, X_2, \dots, X_n) \rightarrow p(Z|X_1, X_2, \dots, X_n)$$

Applications of Probabilistic Models

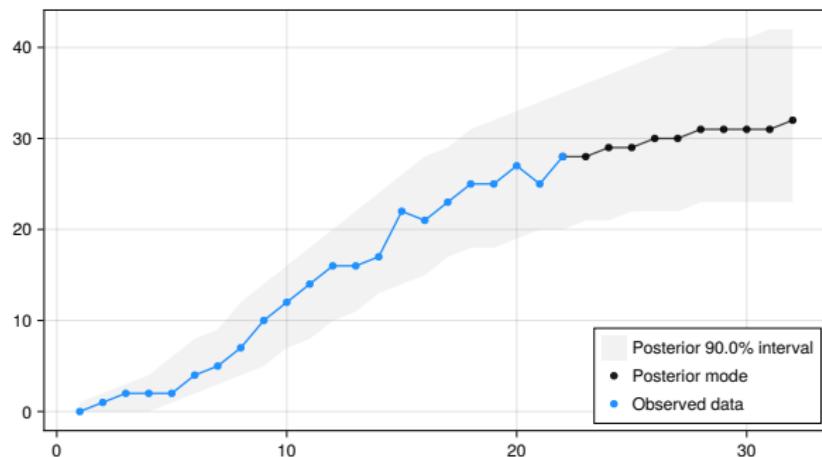
Classification : Discriminant Analysis



$$p_{C_1}(x) \quad p_{C_2}(x)$$

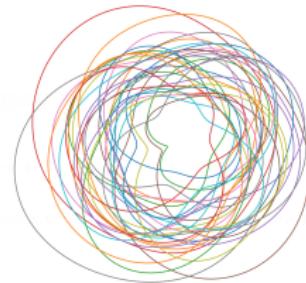
Applications of Probabilistic Models

Forecasting and Regression



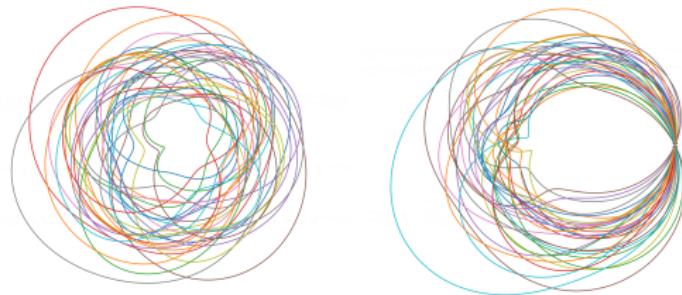
Applications of Probabilistic Models

Uncertainty Quantification



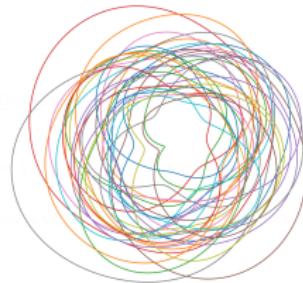
Applications of Probabilistic Models

Uncertainty Quantification



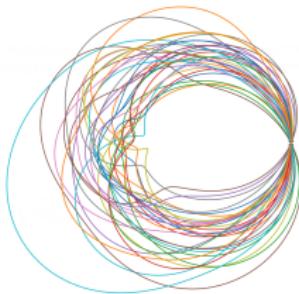
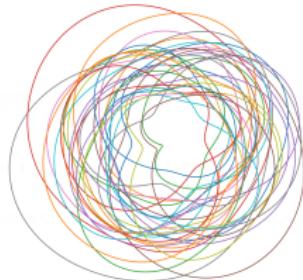
Applications of Probabilistic Models

Uncertainty Quantification



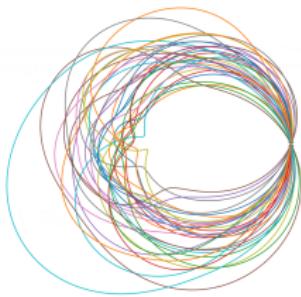
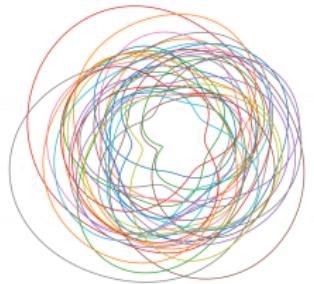
Applications of Probabilistic Models

Uncertainty Quantification



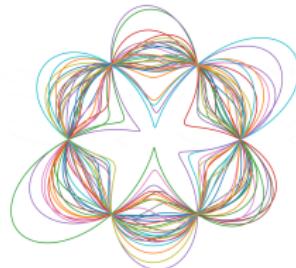
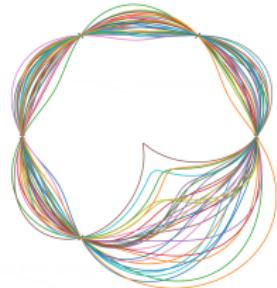
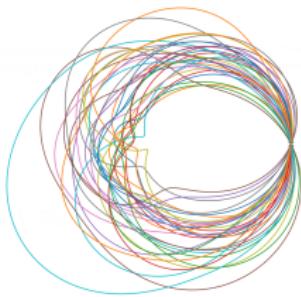
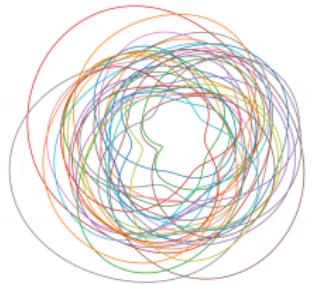
Applications of Probabilistic Models

Uncertainty Quantification



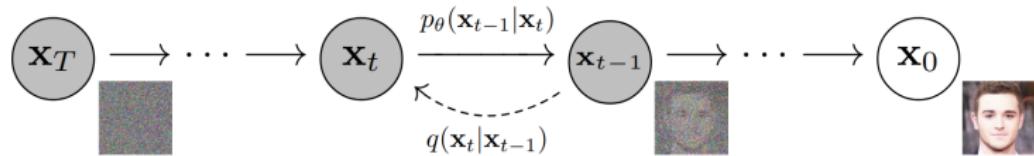
Applications of Probabilistic Models

Uncertainty Quantification



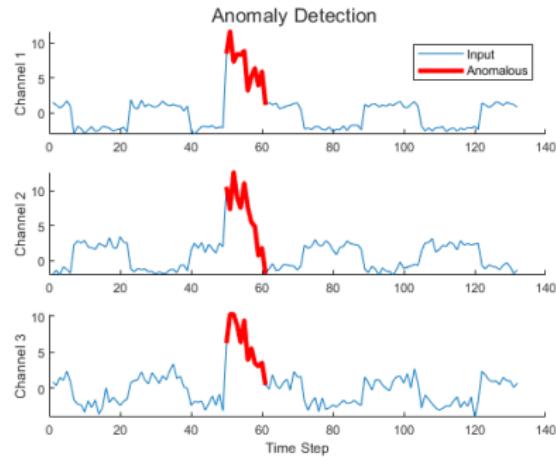
Applications of Probabilistic Models

Data Generation



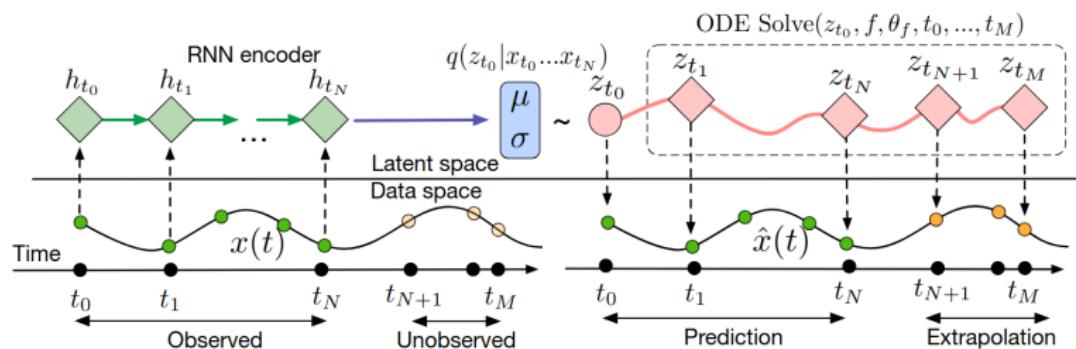
Applications of Probabilistic Models

Anomaly Detection



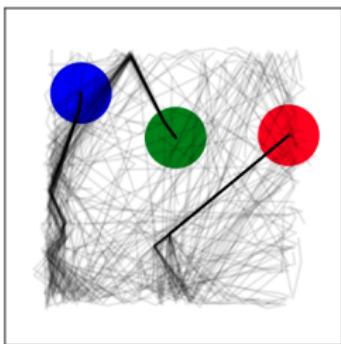
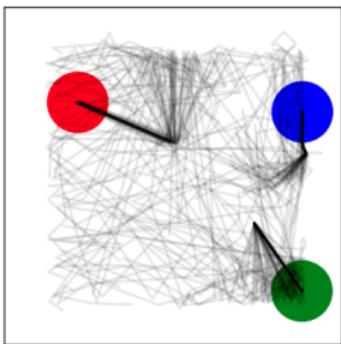
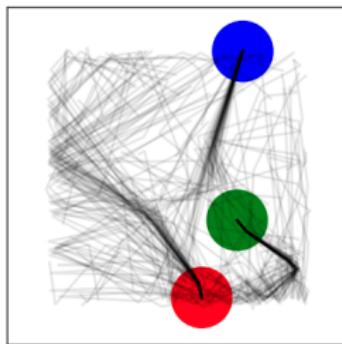
Applications of Probabilistic Models

Scientific Discovery



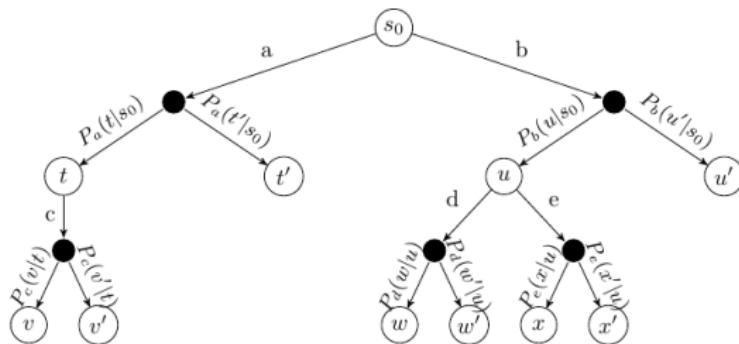
Applications of Probabilistic Models

Data-driven Reasoning



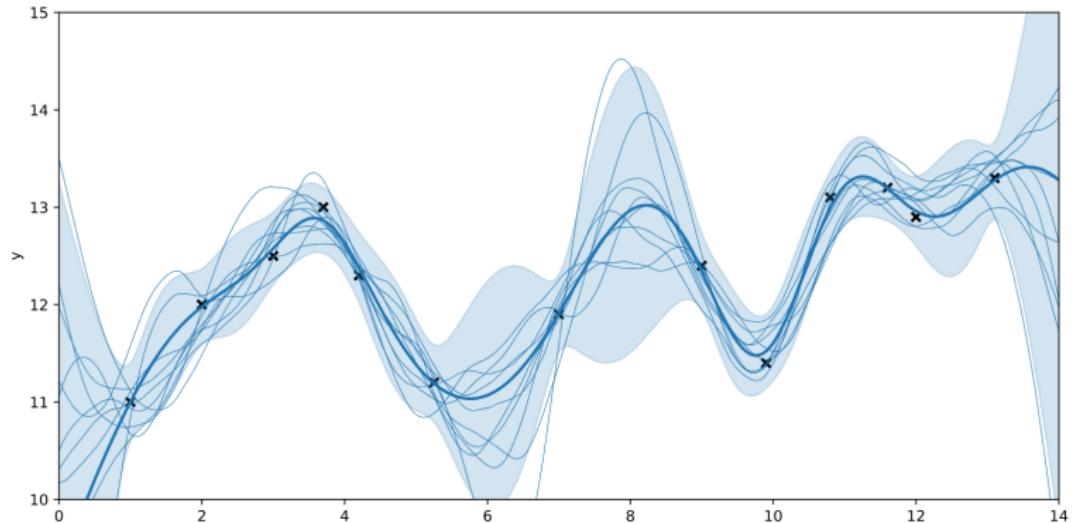
Applications of Probabilistic Models

Reinforcement Learning : Planning



Applications of Probabilistic Models

Guiding Exploration



Challenges : Fantasy vs Reality

$$p(x_1, x_2, \dots, x_n)$$

Having this object, we can solve a wide range of problems

p has encoded anything we know about x_1, \dots, x_n and their interactions

Finding it is not going to be easy!

Challenges : Representation

What does a probabilistic model look like?

Challenges : Inference

Suppose we eventually found a model

$$p(x, z)$$

How can we perform inference?

$$p(z|x)$$

Challenges : Inference

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Challenges : Inference

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

$$p(x) = \int p(x, z) dz$$

Challenges : Inference

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

$$p(x) = \int p(x, z) dz$$

Now imagine what does $p(x)$ look like if we have used something like neural networks inside the model!

Challenges : Inference

If we choose likelihood and prior carefully, Exact Inference is possible.

Likelihood	Conjugate Prior
Bernoulli	Beta
Binomial	Beta
Poisson	Gamma
Categorical	Dirichlet
Uniform	Pareto

Challenges

For a long time it was believed that Bayesian methods are

- limited to small data

Challenges

For a long time it was believed that Bayesian methods are

- limited to small data
- practically not applicable to complex models

Challenges

For a long time it was believed that Bayesian methods are

- limited to small data
- practically not applicable to complex models
- time-consuming and consequently not suitable for online applications

Milestones

So what has changed now!?

- Powerful frameworks for representation

Milestones

So what has changed now!?

- Powerful frameworks for representation
- Advances in inference methods

Milestones

So what has changed now!?

- Powerful frameworks for representation
- Advances in inference methods
- Deep Learning and Differentiable Programming

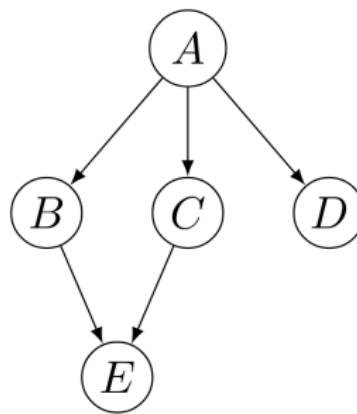
Milestones

So what has changed now!?

- Powerful frameworks for representation
- Advances in inference methods
- Deep Learning and Differentiable Programming
- Advances in hardware (CPU, GPU, TPU, ...)

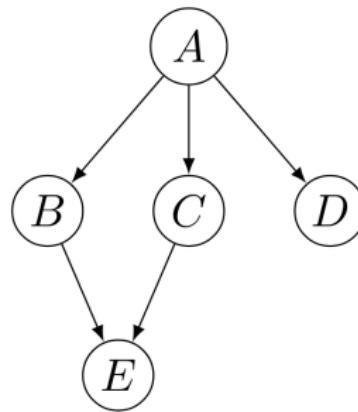
Representing Probabilistic Models

Directed Graphical Models



Representing Probabilistic Models

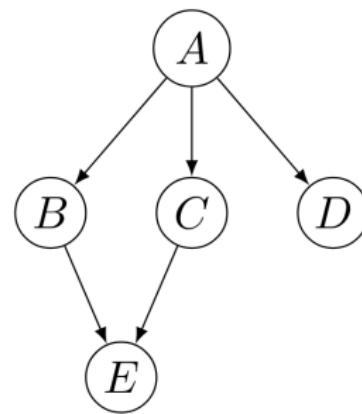
Directed Graphical Models



$$p(A, B, C, D, E) =$$

Representing Probabilistic Models

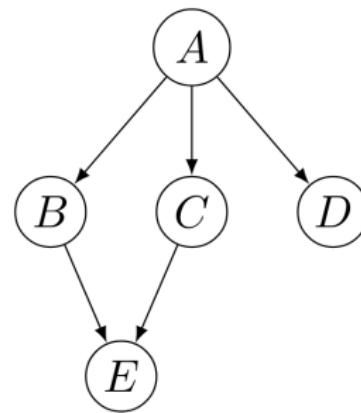
Directed Graphical Models



$$p(A, B, C, D, E) = p(E|B, C)$$

Representing Probabilistic Models

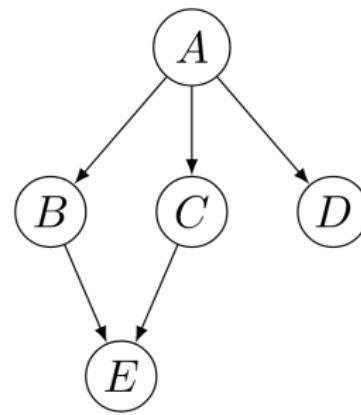
Directed Graphical Models



$$p(A, B, C, D, E) = p(E|B, C)p(B|A)$$

Representing Probabilistic Models

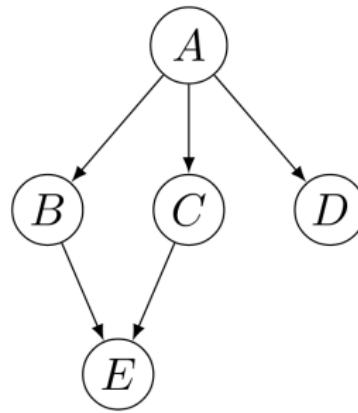
Directed Graphical Models



$$p(A, B, C, D, E) = p(E|B, C)p(B|A)p(C|A)$$

Representing Probabilistic Models

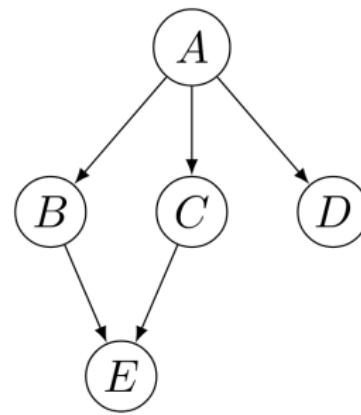
Directed Graphical Models



$$p(A, B, C, D, E) = p(E|B, C)p(B|A)p(C|A)p(D|A)$$

Representing Probabilistic Models

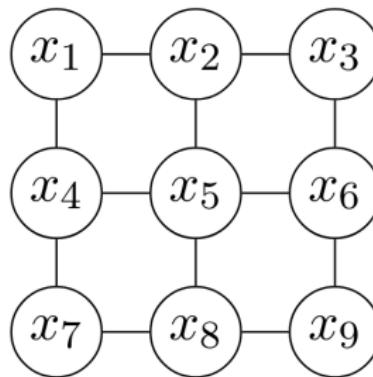
Directed Graphical Models



$$p(A, B, C, D, E) = p(E|B, C)p(B|A)p(C|A)p(D|A)p(A)$$

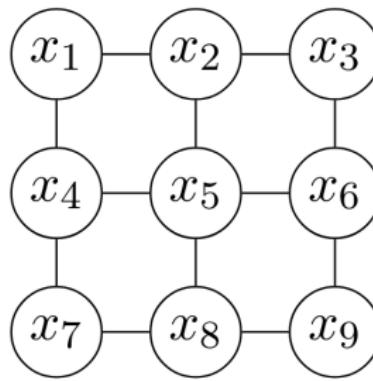
Representing Probabilistic Models

Undirected Graphical Models



Representing Probabilistic Models

Undirected Graphical Models



$$p() = \frac{1}{Z} \prod_{c \in C} \psi_c(x_c)$$

Representing Probabilistic Models

Can we represent a probabilistic model using an algorithm?

Representing Probabilistic Models

$$p(\theta, X)$$

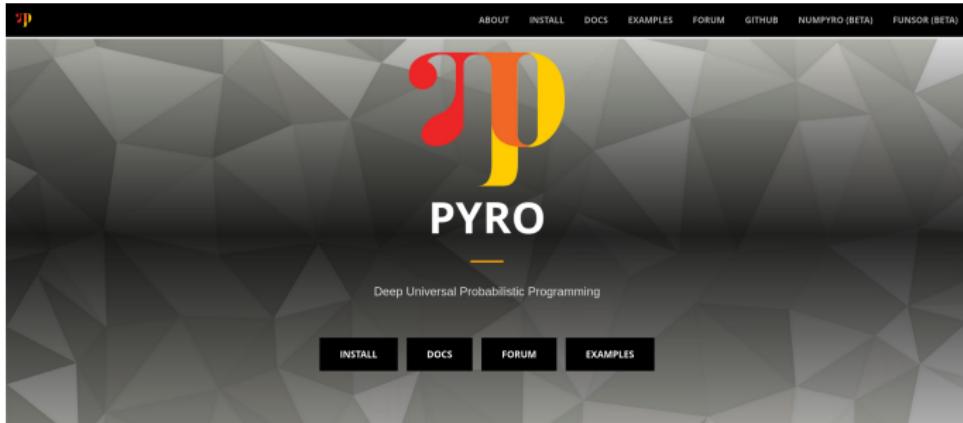
Representing Probabilistic Models

$$p(\theta, X)$$

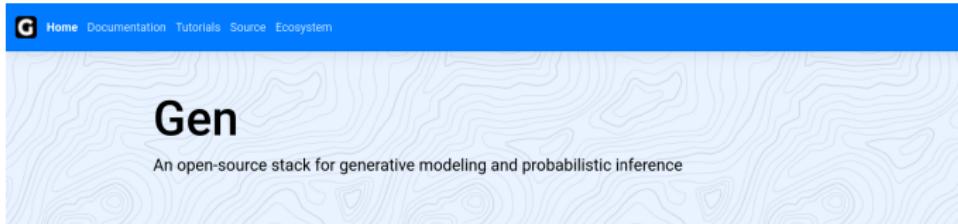
```
 $\theta \sim \text{Beta}(1.0, 1.0)$ 
```

```
for i in 1:N  
    X[i] ~ Bernoulli(theta)  
end
```

Probabilistic Programming Languages



Probabilistic Programming Languages



Why Gen

Gen automates the implementation details of probabilistic inference algorithms

Gen's inference library gives users building blocks for writing efficient probabilistic inference algorithms that are tailored to their models, while automating the tricky math and the low-level implementation details. Gen helps users write hybrid algorithms that combine neural networks, variational inference, sequential Monte Carlo samplers, and Markov chain Monte Carlo.

Probabilistic Programming Languages



Turing.jl: Bayesian inference with probabilistic programming.

Intuitive

Turing models are easy to write and communicate — syntax is close to mathematical notations.

General-purpose

Turing supports models with discrete parameters and stochastic control flow.

Modular and composable

Turing is modular, written entirely in Julia, and is interoperable with the powerful Julia ecosystem.

[Get Started](#)

Probabilistic Programming Languages

PyMC

Home Examples Learn API Community Contributing

Search ctrl + k

On this page

Features Get started Announcements Sponsors

Edit on GitHub Show Source Support PyMC

PyMC

PyMC is a probabilistic programming library for Python that allows users to build Bayesian models with a simple Python API and fit them using Markov chain Monte Carlo (MCMC) methods.

Section Navigation

- PyMC ecosystem
- History
- Testimonials

External links

- Discourse
- Twitter
- YouTube
- LinkedIn
- Meetup
- Github

Features

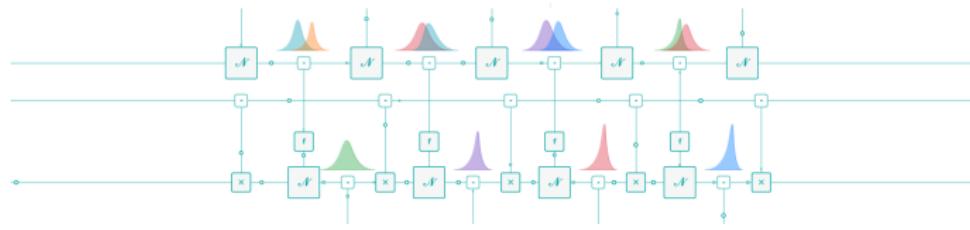
PyMC strives to make Bayesian modeling as simple and painless as possible, allowing users to focus on their problem rather than the methods.

Here is what sets it apart:

Probabilistic Programming Languages

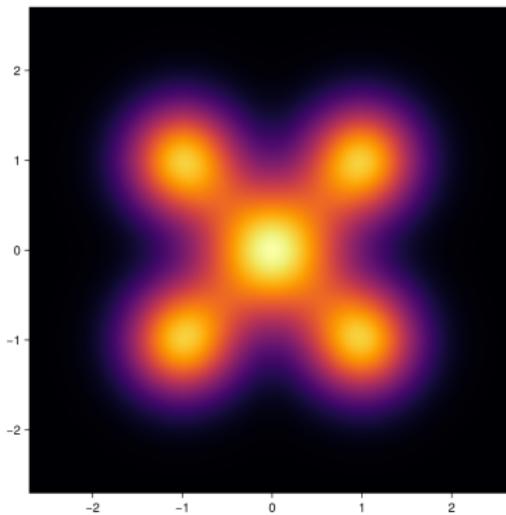


Get Started Documentation Examples Discussions Team Contact GitHub



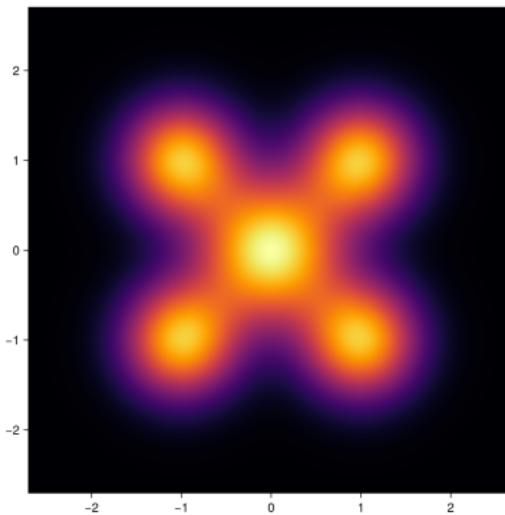
Approximate Inference

Exact Inference is not possible for most of the models



Approximate Inference

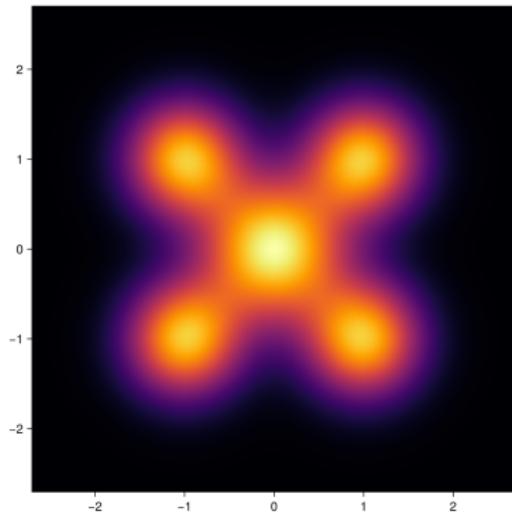
Exact Inference is not possible for most of the models



Fortunately there are some methods for approximate inference

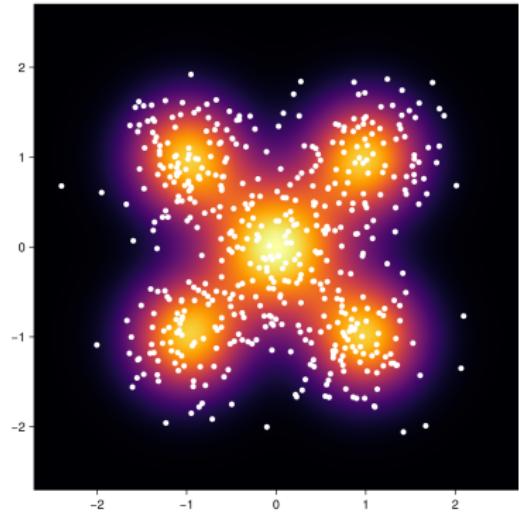
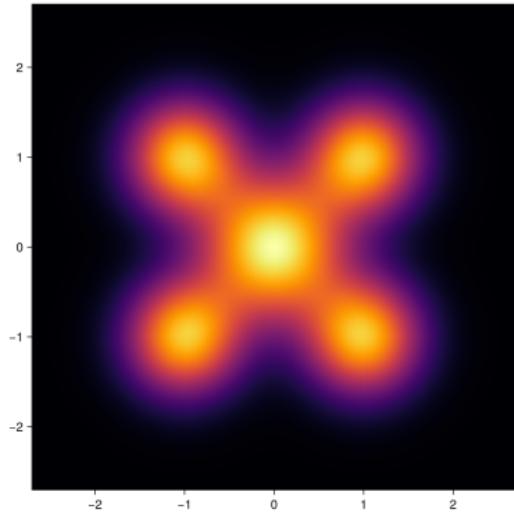
Approximate Inference : MCMC

posterior $p(z|x)$ is intractable



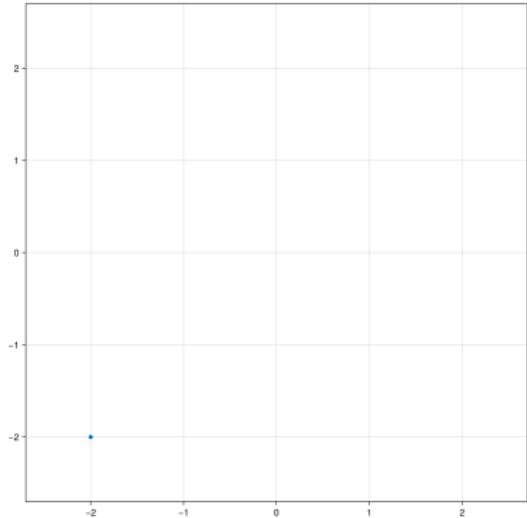
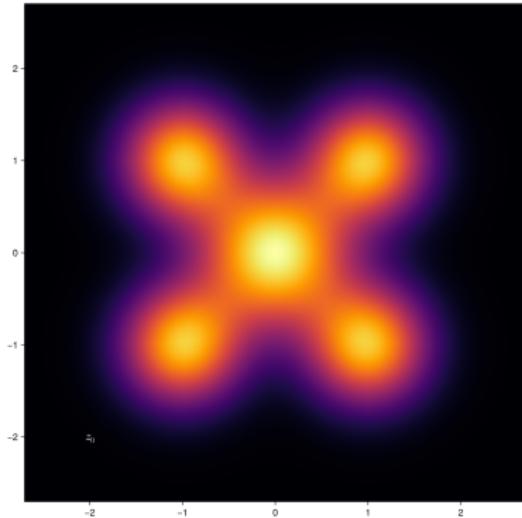
Approximate Inference : MCMC

posterior $p(z|x)$ is intractable

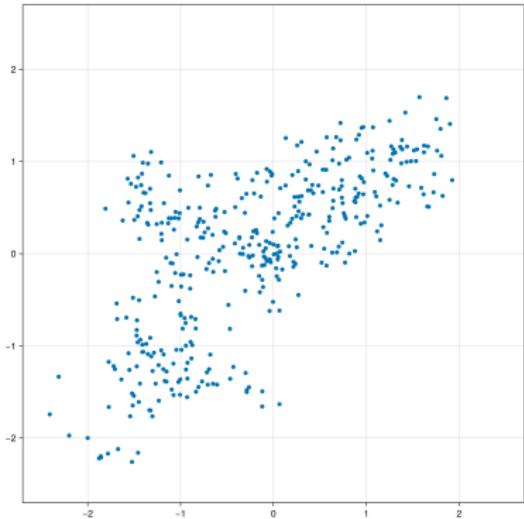
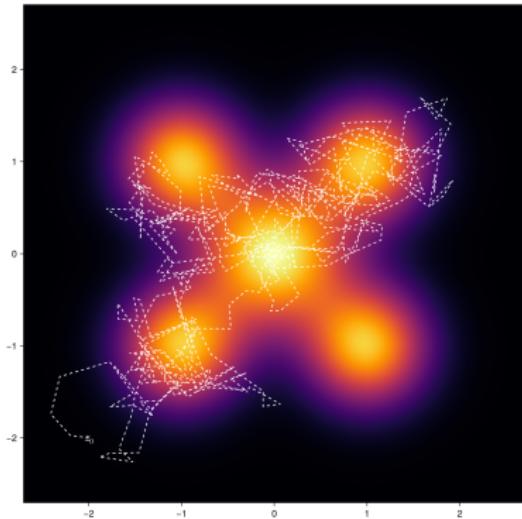


If we somehow generate enough number of samples from $p(z|x)$ then we can estimate our desired quantities.

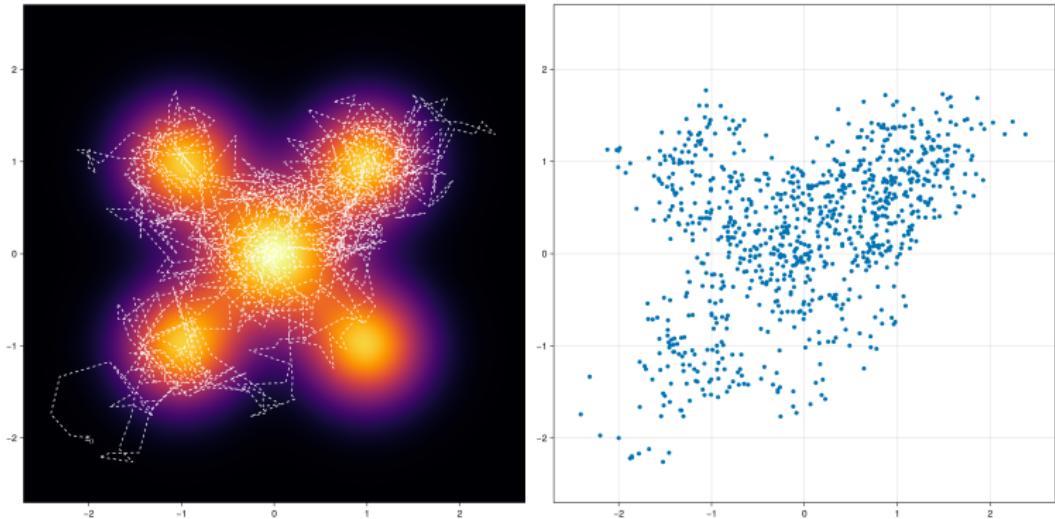
Approximate Inference : MCMC



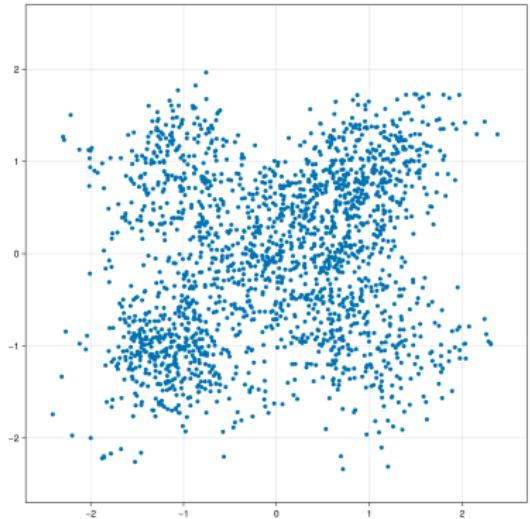
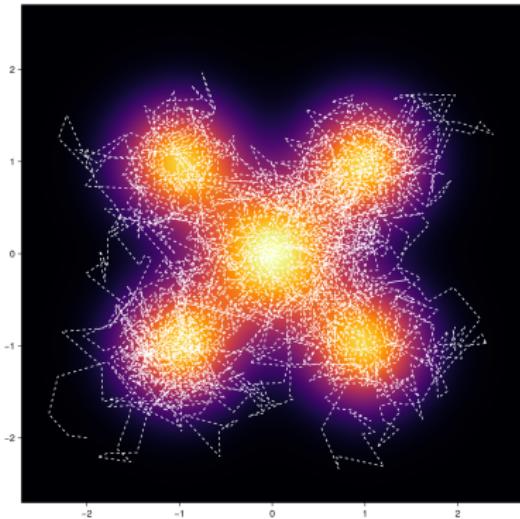
Approximate Inference : MCMC



Approximate Inference : MCMC

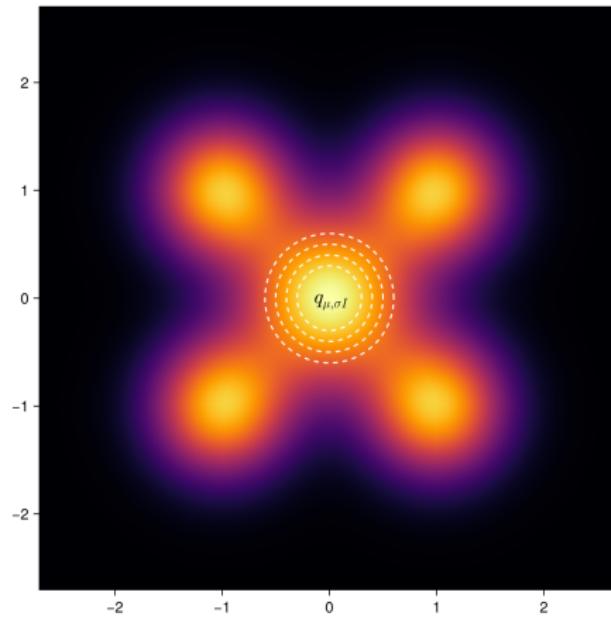


Approximate Inference : MCMC

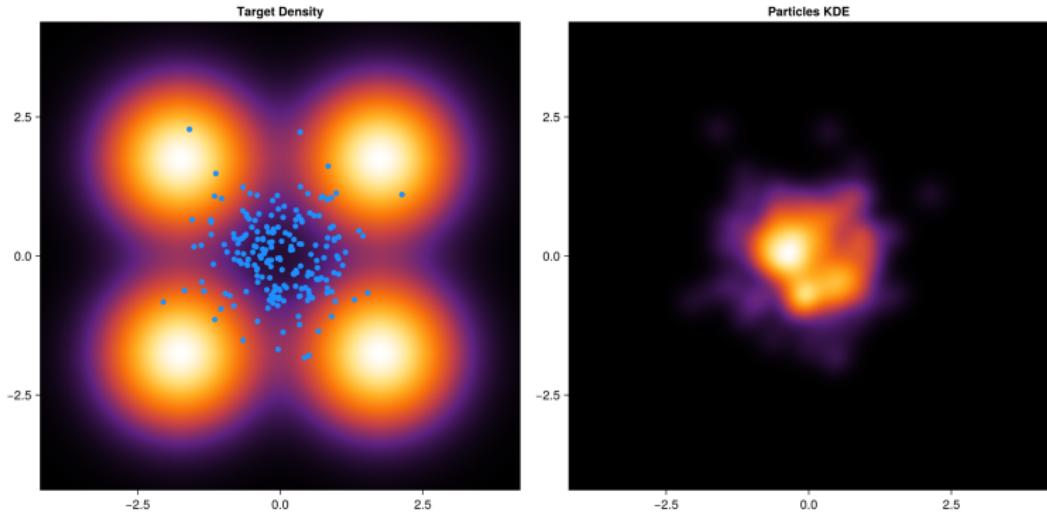


Approximate Inference : Variational Inference

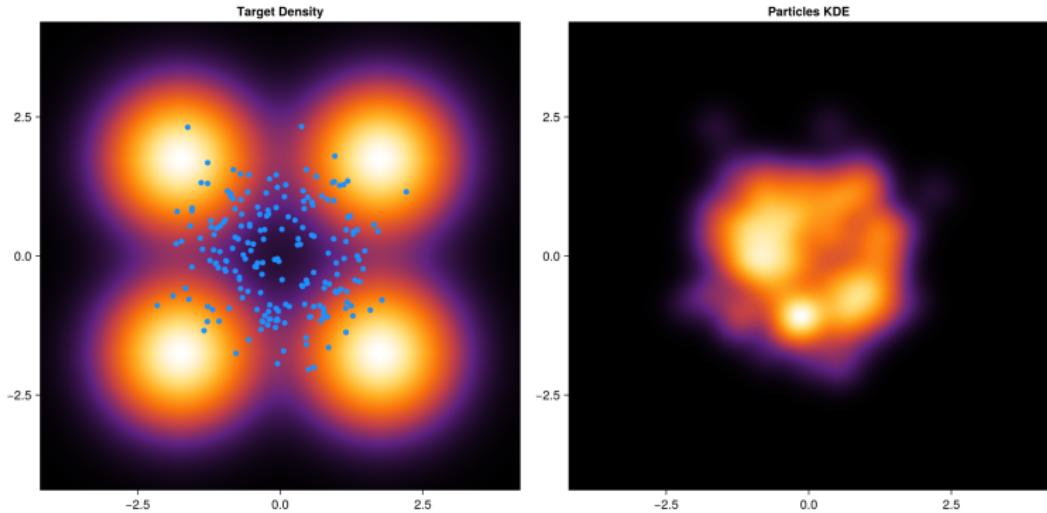
Another idea is trying to find a tractable density $q(z; \lambda)$ as close as possible to the posterior.



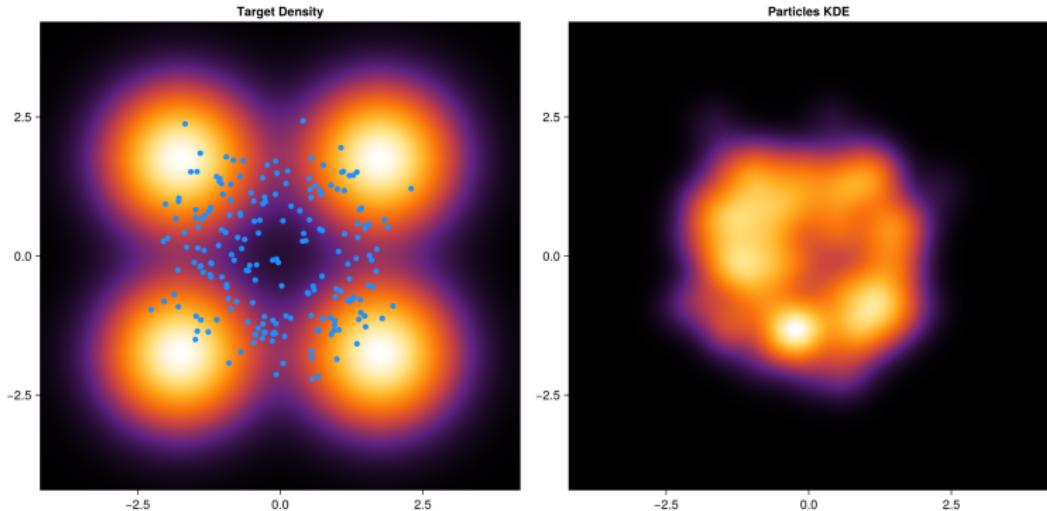
Approximate Inference : Particle-Based Inference



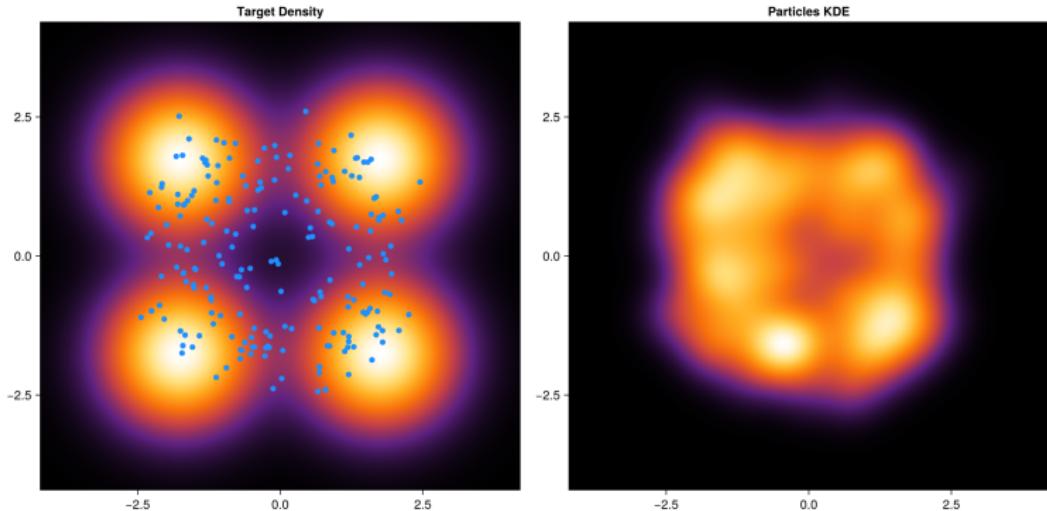
Approximate Inference : Particle-Based Inference



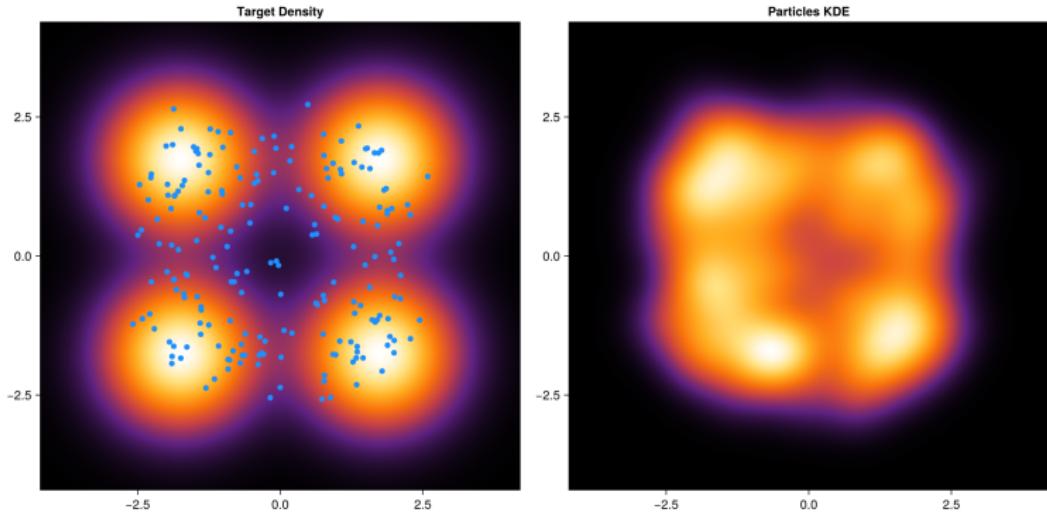
Approximate Inference : Particle-Based Inference



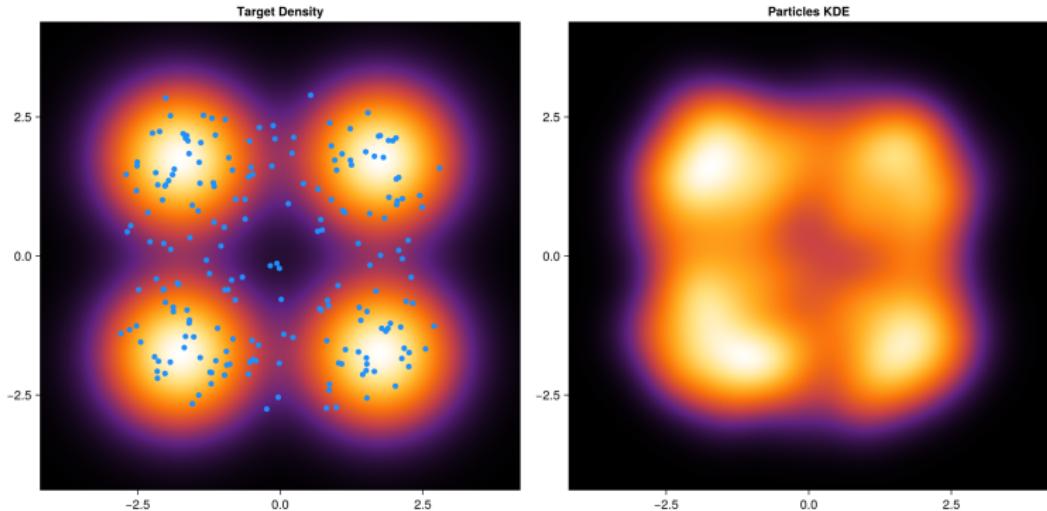
Approximate Inference : Particle-Based Inference



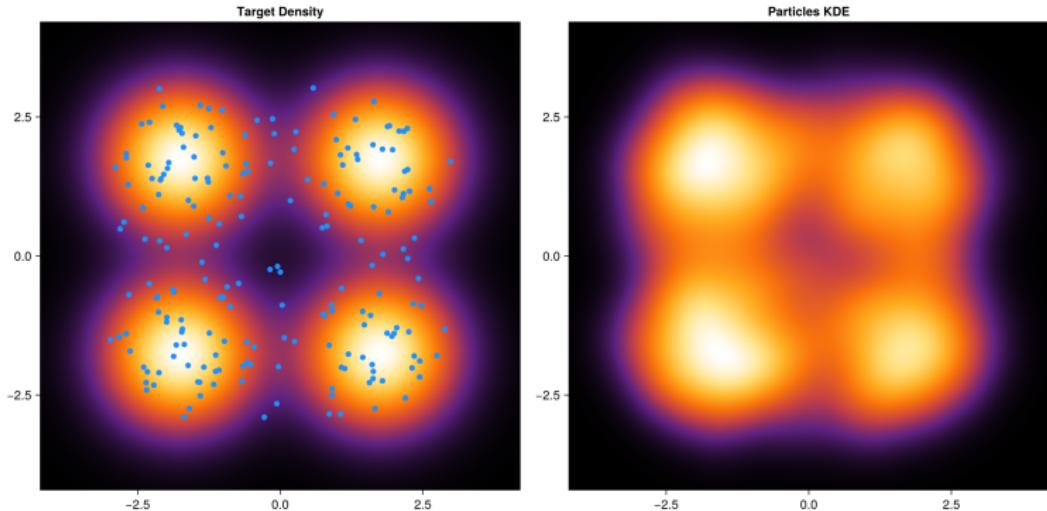
Approximate Inference : Particle-Based Inference



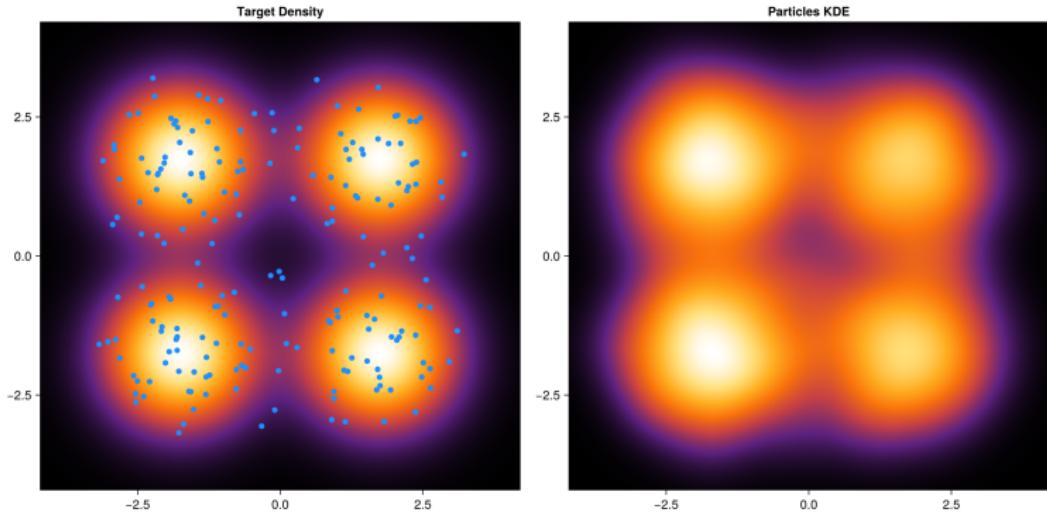
Approximate Inference : Particle-Based Inference



Approximate Inference : Particle-Based Inference



Approximate Inference : Particle-Based Inference



Conclusion

Thanks to advances in modeling and inference methods, Now we can:

- Explore a wide range of models

Conclusion

Thanks to advances in modeling and inference methods, Now we can:

- Explore a wide range of models
- Apply Bayesian methods to large data

Conclusion

Thanks to advances in modeling and inference methods, Now we can:

- Explore a wide range of models
- Apply Bayesian methods to large data
- Use Bayesian methods efficiently even for streaming data