MARKOV KERNELS V2: ALGORITHMS

6

- Kernels mathematically express algorithmic moves:

- Conversely, given a set of instructions for generating $X' \in \mathbb{X}$ to $X \in \mathbb{X}$, there exists a kernel K such that (1) holds

- A kernel K provides instructions to move samples $X \in \mathbb{X}$ to $X' \in \mathbb{X}$

$$X' \sim K(X, \mathrm{d}x') \tag{1}$$

- Both representations are important!

- Kernels allow us to formally study the correctness and mathematical properties of a given algorithm

- Instructions (i.e. pseudo-code) provide implementation details and analyse algorithmic complexity.

- Define $\mu \otimes K \in \mathscr{P}(\mathbb{X} \times \mathbb{X})$ as the joint law $X \sim \mu$ and $X' \sim K(X, \mathrm{d}x')$

$$\mu \otimes K(\mathrm{d}x, \mathrm{d}x') = \mu(\mathrm{dx})K(x, \mathrm{d}x')$$

MARKOV KERNELS V2: ALGORITHMS

- Kernels mathematically express algorithmic moves:
  - A kernel K provides instructions to move samples $X \in \mathbb{X}$ to $X' \in \mathbb{X}$

$$X' \sim K(X, \mathrm{d}x') \tag{1}$$

  - Conversely, given a set of instructions for generating $X' \in \mathbb{X}$ to $X \in \mathbb{X}$, there exists a kernel K such that (1) holds

- Both representations are important!
  - Kernels allow us to formally study the correctness and mathematical properties of a given algorithm
  - Instructions (i.e. pseudo-code) provide implementation details and analyse algorithmic complexity.

- Define $\mu \otimes K \in \mathscr{P}(\mathbb{X} \times \mathbb{X})$ as the joint law $X \sim \mu$ and $X' \sim K(X, \mathrm{d}x')$

$$\mu \otimes K(\mathrm{d}x, \mathrm{d}x') = \mu(\mathrm{dx})K(x, \mathrm{d}x')$$

EXAMPLES: