Bayesian Inference with Probabilistic Programs

Fritz Obermeyer

Department of Mathematics
Carnegie Mellon University

2009:04:14
Overview

Aim Explain research in Multiple Target Tracking

Aim Mention research in Programming Languages

Theme Probabilistic Programs
Deterministic programs

\[ X = 2 \]
\[ Z = X + 1 \]

output \( Z \)
Probabilistic programs

\[ X \sim \text{unif}(0, 1) \]
\[ Z \sim \text{unif}(0, X) \]
\text{output } Z
Probabilistic programs

\[ X \sim \text{unif}(0, 1) \]
\[ Z \sim \text{unif}(0, X) \]

output \ Z
Probabilistic programs

\[ X \sim \text{unif}(0, 1) \]
\[ Z \sim \text{unif}(0, X) \]

output \ Z

\[ P(X, Z) = P(X) \ P(Z \mid X) \]
Bayesian networks use definitions

\[
\begin{align*}
X & \sim \phi_X \\
W & \sim \phi_W(X) \\
Y & \sim \phi_Y(X) \\
Z & \sim \phi_Z(W, Y)
\end{align*}
\]

output $Z$
Bayesian networks use definitions

\[
X \sim \phi_X \\
W \sim \phi_W(X) \\
Y \sim \phi_Y(X) \\
Z \sim \phi_Z(W, Y)
\]

output Z
Bayesian networks use definitions

\[ X \sim \phi_X \]
\[ Y \sim \phi_Y(X) \]
\[ W \sim \phi_W(X) \]
\[ Z \sim \phi_Z(W, Y) \]
output \( Z \)
HMMs, Bayes filters use loops

\[
X^0 \sim \phi_{\text{prior}} \\
\text{for } t \text{ in } [1, \ldots] :: \\
X^t \sim \phi_{\text{trans}}(X^{t-1}) \\
Z^t \sim \phi_{\text{obs}}(X^t) \\
\text{output } Z^t
\]
HMMs, Bayes filters use loops

\[
\begin{align*}
X^0 &\sim \phi_{\text{prior}} \\
\text{for } t \text{ in } [1, \ldots] : \\
X^t &\sim \phi_{\text{trans}}(X^{t-1}) \\
Z^t &\sim \phi_{\text{obs}}(X^t) \\
\text{output } Z^t
\end{align*}
\]
Multiple Target Tracking

\[
\begin{align*}
X^0_1 & \sim \phi_{\text{prior}} \\
X^0_2 & \sim \phi_{\text{prior}} \\
\text{for } t \text{ in } [1, \ldots] : \\
X^t_1 & \sim \phi_{\text{trans}}(X^{t-1}_1) \\
X^t_2 & \sim \phi_{\text{trans}}(X^{t-1}_2) \\
Z^t_1 & \sim \phi_{\text{obs}}(X^t_1) \\
Z^t_2 & \sim \phi_{\text{obs}}(X^t_2) \\
\text{output } Z^t_1, Z^t_2
\end{align*}
\]
Multiple Target Tracking

\[
\begin{align*}
X_1^t & \sim \phi_{\text{prior}} \\
X_2^0 & \sim \phi_{\text{prior}} \\
\text{for } t \text{ in } [1, \ldots] : \\
X_1^t & \sim \phi_{\text{trans}}(X_1^{t-1}) \\
X_2^t & \sim \phi_{\text{trans}}(X_2^{t-1}) \\
Z_1^t & \sim \phi_{\text{obs}}(X_1^t) \\
Z_2^t & \sim \phi_{\text{obs}}(X_2^t) \\
\text{output } Z_1^t, Z_2^t
\end{align*}
\]
...but sometimes association is ambiguous
...but sometimes association is ambiguous
...but sometimes association is ambiguous
...but sometimes association is ambiguous
**Multiple Hypothesis Tracking**

\[ \begin{align*}
\text{for } t \text{ in } [1, \ldots] : \\
A & \sim \phi_{\text{assoc}}(\ldots) \\
\text{branch on } A : \\
\text{case } =: \\
X_t^1 & \sim \phi_{\text{trans}}(X_t^{t-1}) \\
X_t^2 & \sim \phi_{\text{trans}}(X_t^{t-1}) \\
\text{case } x : \\
X_t^1 & \sim \phi_{\text{trans}}(X_t^{t-1}) \\
X_t^2 & \sim \phi_{\text{trans}}(X_t^{t-1}) \\
\ldots
\end{align*} \]
Long-term MHT is too expensive
Bayes Net Tracking Database
Bayes Net Tracking Database
for \( t \) in \([1, \ldots]\):

\[
\text{if } X^t_1 \text{ is close to } X^t_2:\n\]

\[
X^t \sim \phi_{\text{mix}}(X^{t-1})
\]

else:

\[
X^t_1 \sim \phi_{\text{trans}}(X^{t-1}_1)
\]

\[
X^t_2 \sim \phi_{\text{trans}}(X^{t-1}_2)
\]

\[
\ldots
\]
Iteratively compute state association

\[ P(\chi) = \frac{L(\chi)}{L(\bar{\chi}) + L(\chi) + L(\chi)} \]

Filter & Smooth

Mean Association
What about more general programs?

\[
F \sim \phi_{\text{program}} \\
\text{for } i \text{ in } [1, \ldots ]: \\
\text{input } X^i \\
Y^i \sim F(X^i) \\
\text{output } Y^i
\]
What is a random program?
What is a random program?

\[
\text{prog} ::= \begin{align*}
S \\
| 
K \\
| 
\text{prog}(\text{prog})
\end{align*}
\]
What is a random program?

\[
\text{prog} ::= \begin{cases} \text{S} \\ \text{K} \\ \text{prog}(\text{prog}) \end{cases}
\]

\[
\phi_{\text{prog}} = \frac{1}{4} \ (\text{return S}) \\
+ \frac{1}{4} \ (\text{return K}) \\
+ \frac{1}{2} \ (F \sim \phi_{\text{prog}} \\
\quad \quad X \sim \phi_{\text{prog}} \\
\quad \quad \text{return } F(X) )
\]
Research in programming languages

Developed tools for coarse equivalence
Research in programming languages

Developed tools for coarse equivalence

Simulated types in untyped languages
Research in programming languages

Developed tools for coarse equivalence

Simulated types in untyped languages

Implemented an equational theorem prover
Research in programming languages

Developed tools for coarse equivalence

Simulated types in untyped languages

Implemented an equational theorem prover
  Finds first 10,000 programs-$\text{mod}$-equivalence
Research in programming languages

Developed tools for coarse equivalence

Simulated types in untyped languages

Implemented an equational theorem prover
  Finds first 10,000 programs-mod-equivalence
  Generates conjectures from evidence
Research in programming languages

Developed tools for coarse equivalence

Simulated types in untyped languages

Implemented an equational theorem prover
  Finds first 10,000 programs-mod-equivalence
  Generates conjectures from evidence
  Optimizes language to fit examples
Questions?