24 mph
59 mph
GeoCoordinate PrevLocn = Get();
Sleep(5);
GeoCoordinate Location = Get();
GeoCoordinate PrevLocn = Get();
Sleep(5);
GeoCoordinate Location = Get();
double Dist =
    Distance(PrevLocn, Location);
double Speed = Dist / 5;
GeoCoordinate PrevLocn = Get();
Sleep(5);
GeoCoordinate Location = Get();
double Dist =
    Distance(PrevLocn, Location);
double Speed = Dist / 5;

if (Speed > 4)
    Alert("Keep it up!");
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Keep it up!
Uncertain<$T>$

an abstraction for reasoning about noise [ASPLOS’14]

adding domain knowledge

language constructs to make data more accurate
GeoCoordinate PrevLocn = Get();
Sleep(5);
GeoCoordinate Location = Get();
double Dist = Distance(PrevLocn, Location);
double Speed = Dist / 5;

if (Speed > 4)
    Alert("Keep it up!");
Uncertain<GeoCoordinate> PrevLocn = Get();
Sleep(5);
Uncertain<GeoCoordinate> Location = Get();
Uncertain<double> Dist =
    Distance(PrevLocn, Location);
Uncertain<double> Speed = Dist / 5;

if (Speed > 4) // Hypothesis test
    Alert("Keep it up!");
Uncertain<GeoCoordinate> PrevLocn = Get();
Sleep(5);
Uncertain<GeoCoordinate> Location = Get();
Uncertain<double> Dist =
    Distance(PrevLocn, Location);
Uncertain<double> Speed = Dist / 5;

if (Speed > 4)
    Alert("Keep it up!");
Pragmatics

- **Uncertain<T>** encapsulates probability distributions and hides statistical complexity
  - Computing over distributions
  - Deciding conditionals
Computations: Represent distributions by random samples
Operators build a Bayesian network rather than evaluating immediately.

\[ D = A / B \]
\[ E = D - C \]
Deciding at Conditionals

```java
if (Speed > 4)
    Alert("Keep it up!");
```
Deciding at Conditionals

```java
if (Speed > 4)
    Alert("Keep it up!");
```

More likely than not that Speed > 4?
Deciding at Conditionals

\[ \text{if } ((\text{Speed} > 4).\Pr(0.9)) \]
\[ \text{Alert("Keep it up!")} ; \]

At least 90% likely that \( \text{Speed} > 4 \)?

\[ \Pr[\text{Speed} > 4] \]
GeoCoordinate PrevLocn = Get();
Sleep(5);
GeoCoordinate Location = Get();
double Dist = Distance(PrevLocn, Location);
double Speed = Dist / 5;

if (Speed > 4)
    Alert("That’s crazy!");
Uncertain<GeoCoordinate> PrevLocn = Get();
Sleep(5);
Uncertain<GeoCoordinate> Location = Get();
Uncertain<double> Dist =
    Distance(PrevLocn, Location);
Uncertain<double> Speed = Dist / 5;

if (Speed > 4)
    Alert("That’s crazy!");
Uncertain<GeoCoordinate> PrevLocn = Get();
Sleep(5);
Uncertain<GeoCoordinate> Location = Get();
Uncertain<double> Dist =
    Distance(PrevLocn, Location);
Uncertain<double> Speed = Dist / 5;

if ((Speed > 4).Pr(0.9))
    Alert("That’s crazy!"),
Data and Inference Driven Programming Challenges

• What do we do when things go wrong?
  • Debugging large probabilistic systems is hard!
• How do developers easily express domain knowledge?
GPS Driving Application
Driving Application

- GPS
- Location = GPS # roads

Driving on a road (or not!)
Adding Domain Knowledge

--- GPS
Adding Domain Knowledge

--- GPS

--- roads
Adding Domain Knowledge

\[ \Pr[H \mid E] = \frac{\Pr[E \mid H] \Pr[H]}{\Pr[E]} \]

--- GPS
--- roads
--- location = GPS # roads
Uncertain<\(T\> 
Programming with estimates 

Thank you