

Uncertain $\langle T \rangle$ Programming with Estimates

James Bornholt - University of Washington

Na Meng - University of Texas at Austin

Todd Mytkowicz - Microsoft Research

Kathryn S. McKinley - Microsoft Research

Microsoft Research

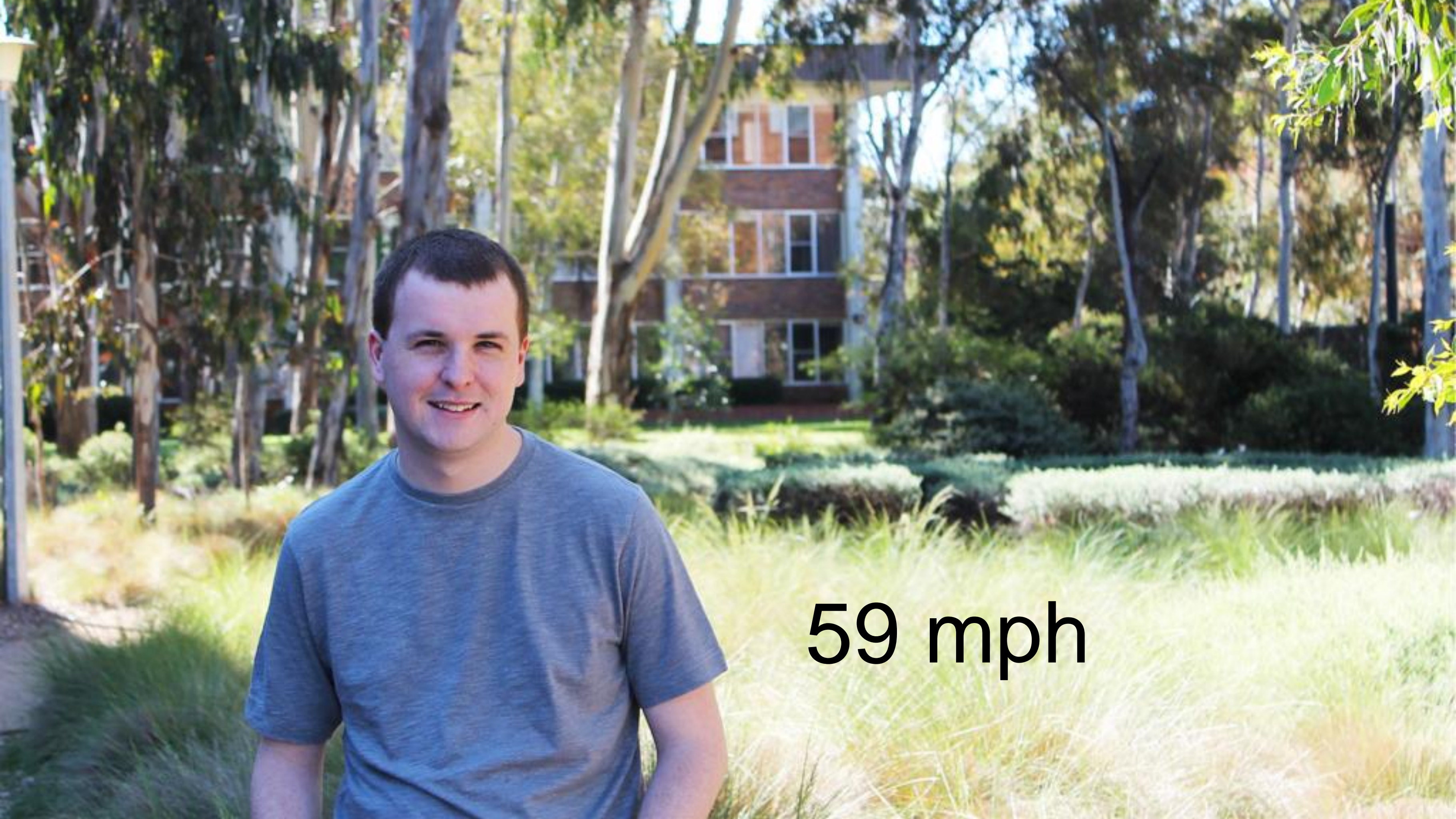
Faculty Summit
2015

July 8-9, 2015





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;
```

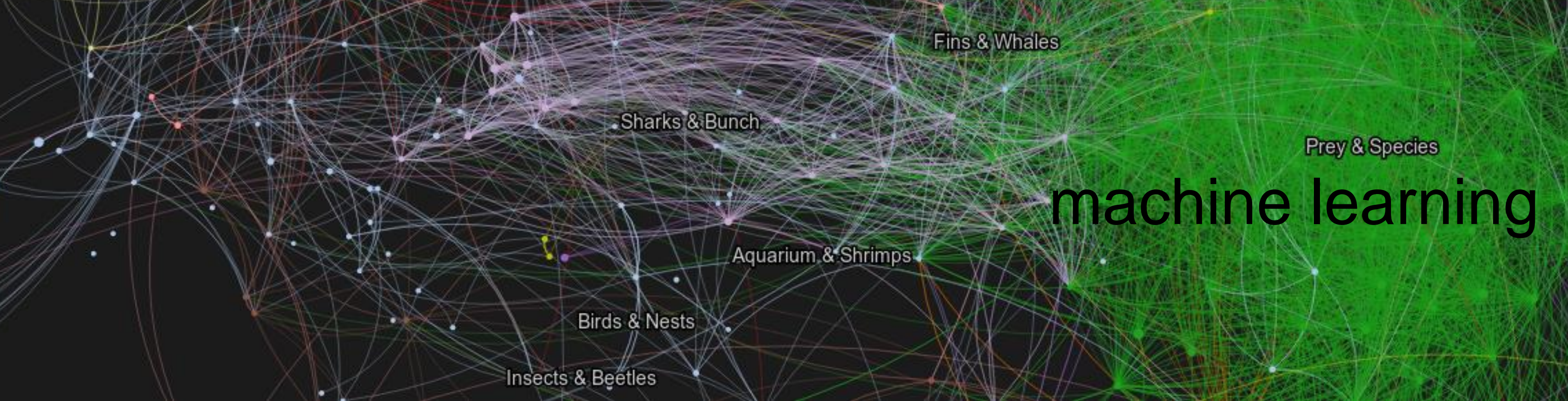
```
GeoCoordinate PrevLocn = Get();  
Sleep(5);  
GeoCoordinate Location = Get();  
double Dist =  
    Distance(PrevLocn, Location);  
double Speed = Dist / 5;  
  
if (Speed > 4)  
    Alert("Keep it up!");
```




59 mph



sensors



machine learning



approximate
computing

Uncertain $\langle T \rangle$

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!");
```



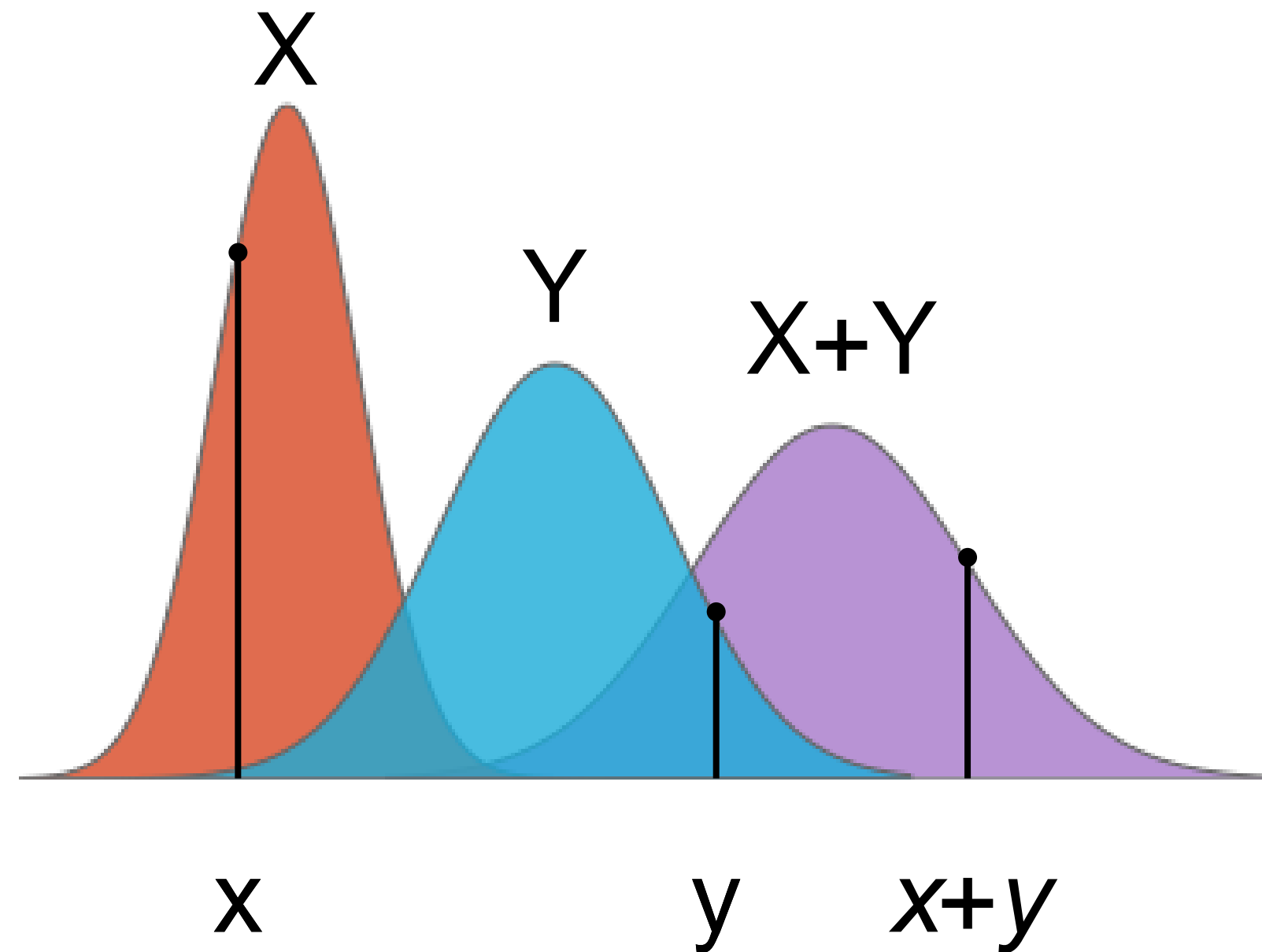
```
Uncertain<GeoCoordinate> PrevLocn = Get();  
Sleep(5);  
Uncertain<GeoCoordinate> Location = Get();  
Uncertain<double> Dist =  
    Distance(PrevLocn, Location);  
Uncertain<double> Speed = Dist / 5;  
  
if (Speed > 4)  
    Alert("Keep it up!");
```

86% fewer errors

Pragmatics

- **Uncertain<T>** encapsulates probability distributions and hides statistical complexity
- Computing over distributions
- Deciding conditionals

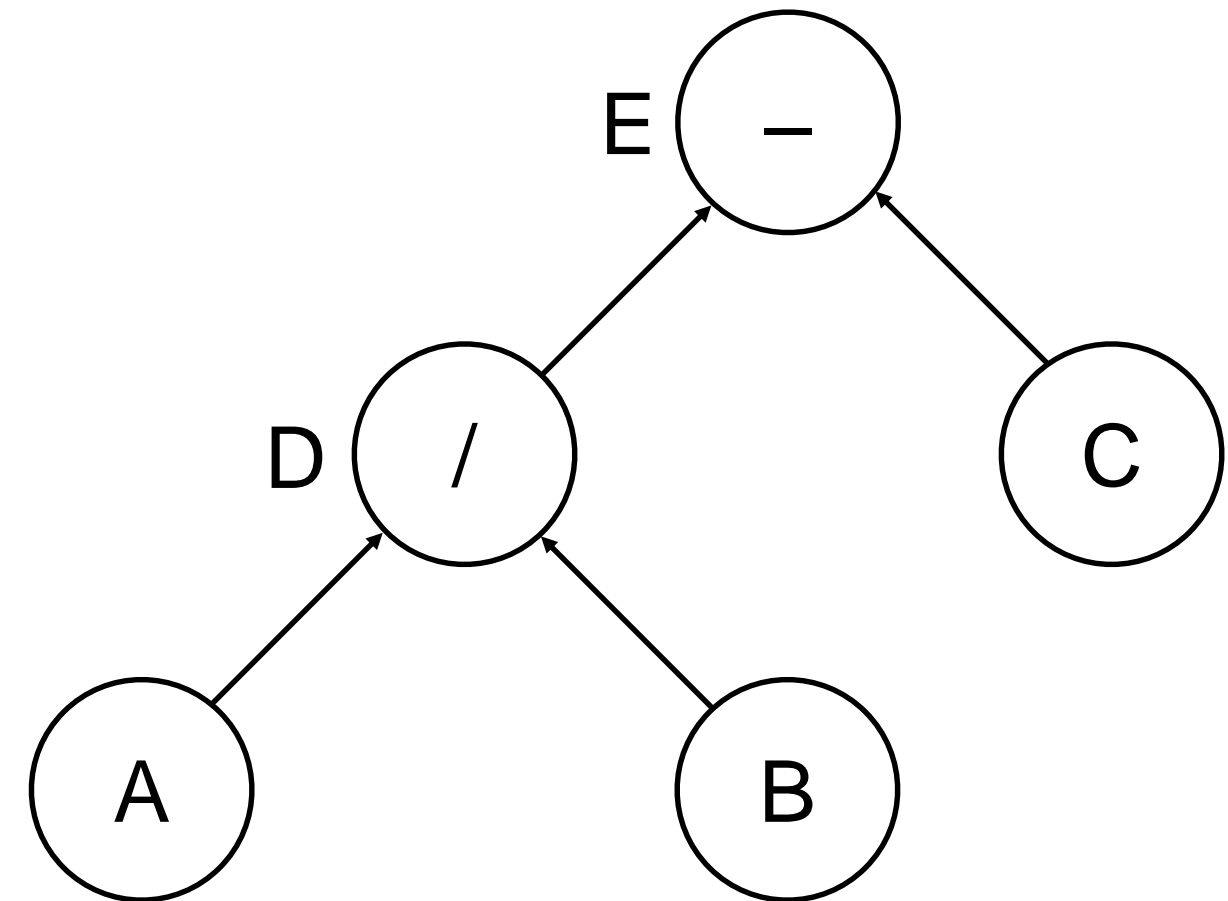
Computations: Represent distributions by random samples



Operators build a Bayesian network rather than evaluating immediately.

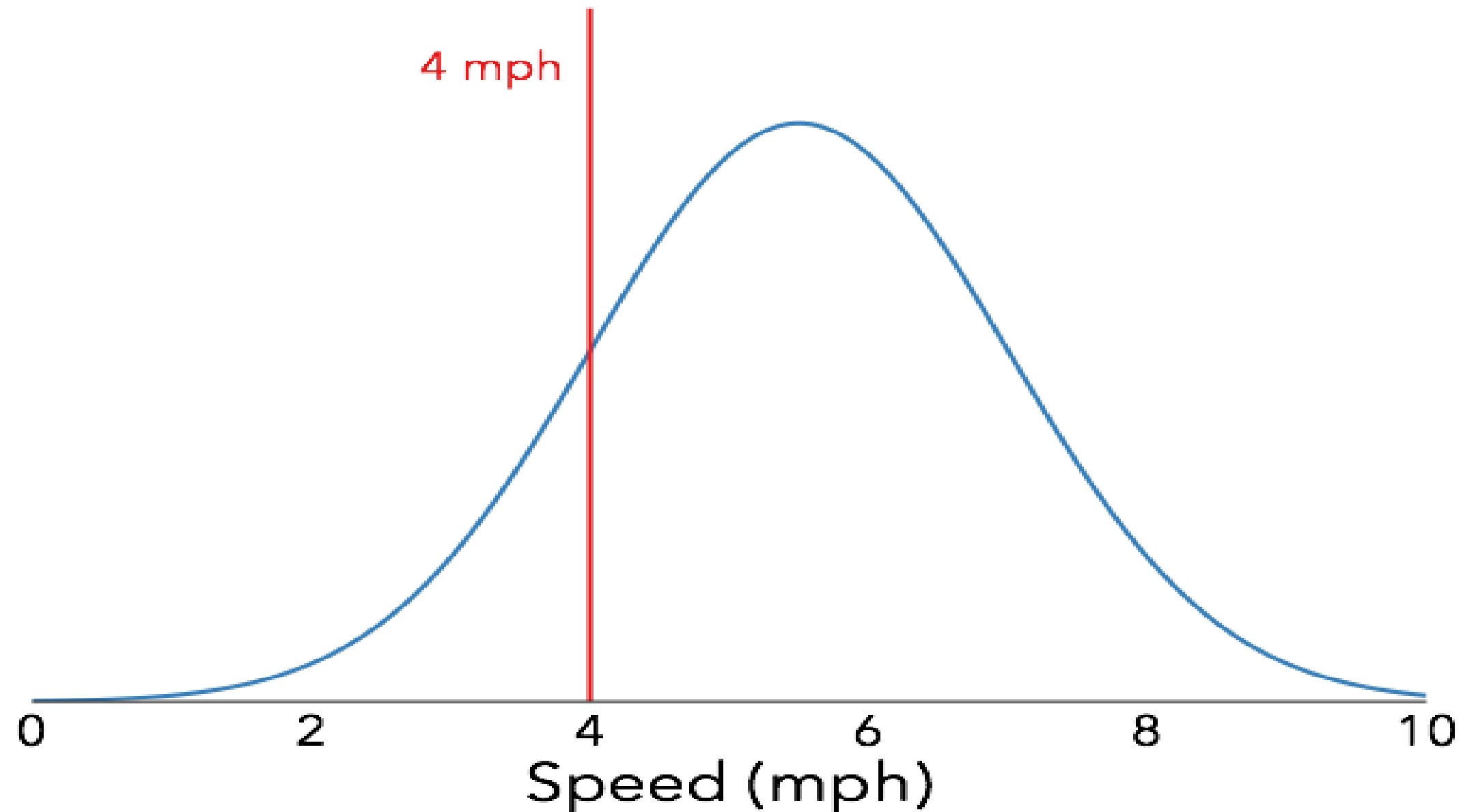
$$D = A / B$$

$$E = D - C$$



Deciding at Conditionals

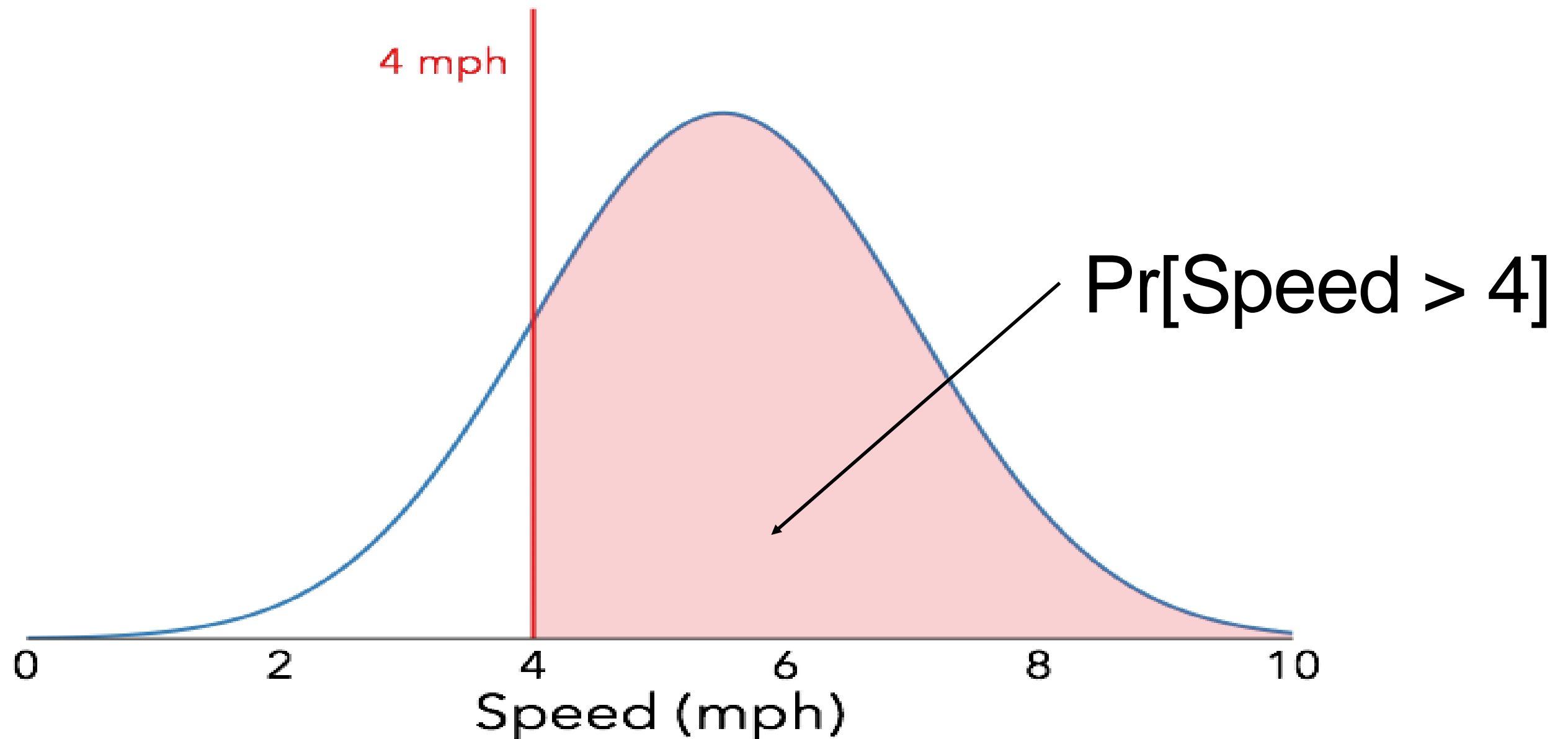
```
if (Speed > 4)  
    Alert("Keep it up!");
```



Deciding at Conditionals

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

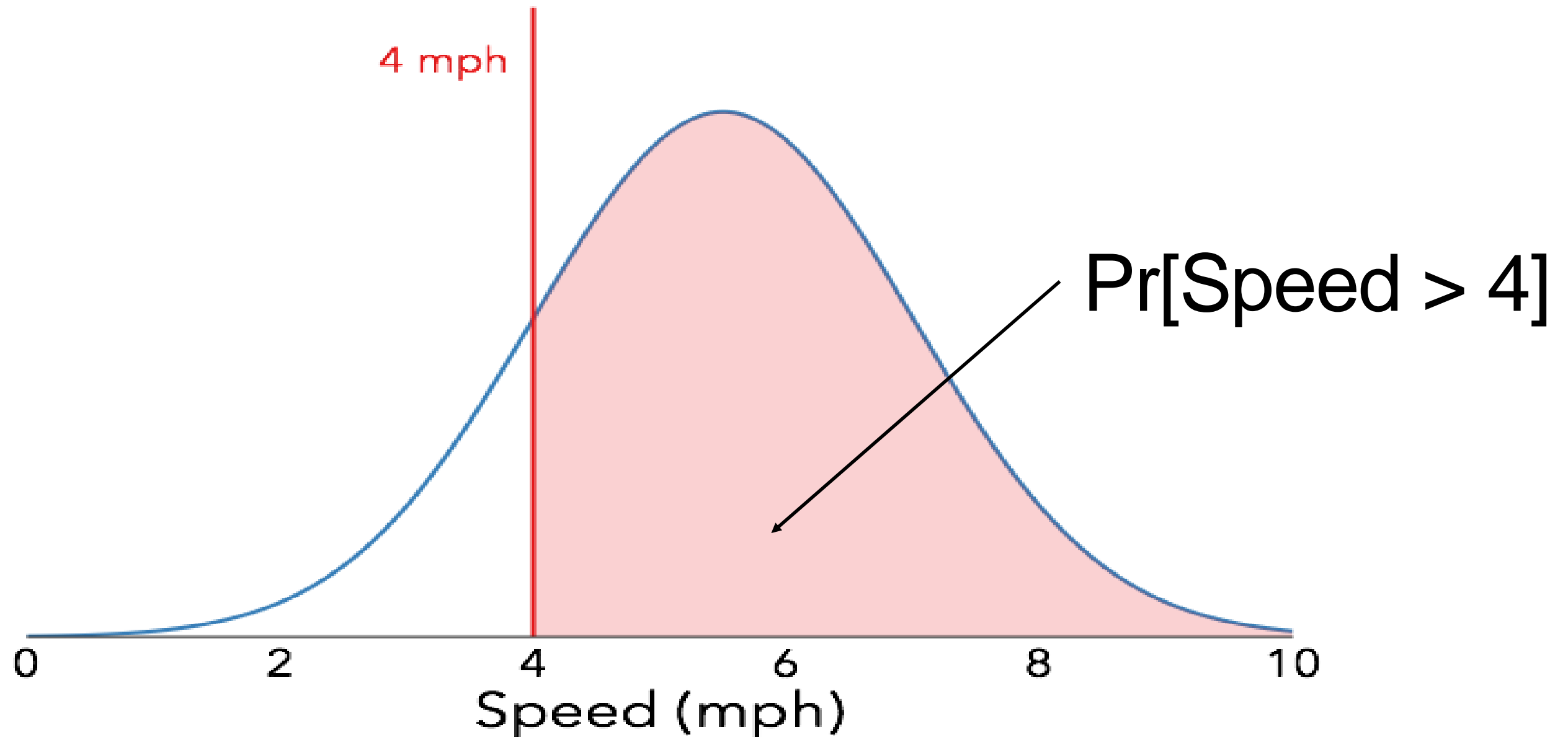
More likely than not
that $\text{Speed} > 4$?



Deciding at Conditionals

```
if ((Speed > 4).Pr(0.9))  
  Alert("Keep it up!");
```

At least 90% likely
that 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!");
```

Naïve: 30 times

```
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!");
```

*Naïve: 30 times
50%: 4 times*


```
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!");
```

*Naïve: 30 times
50%: 4 times
90%: never*

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?

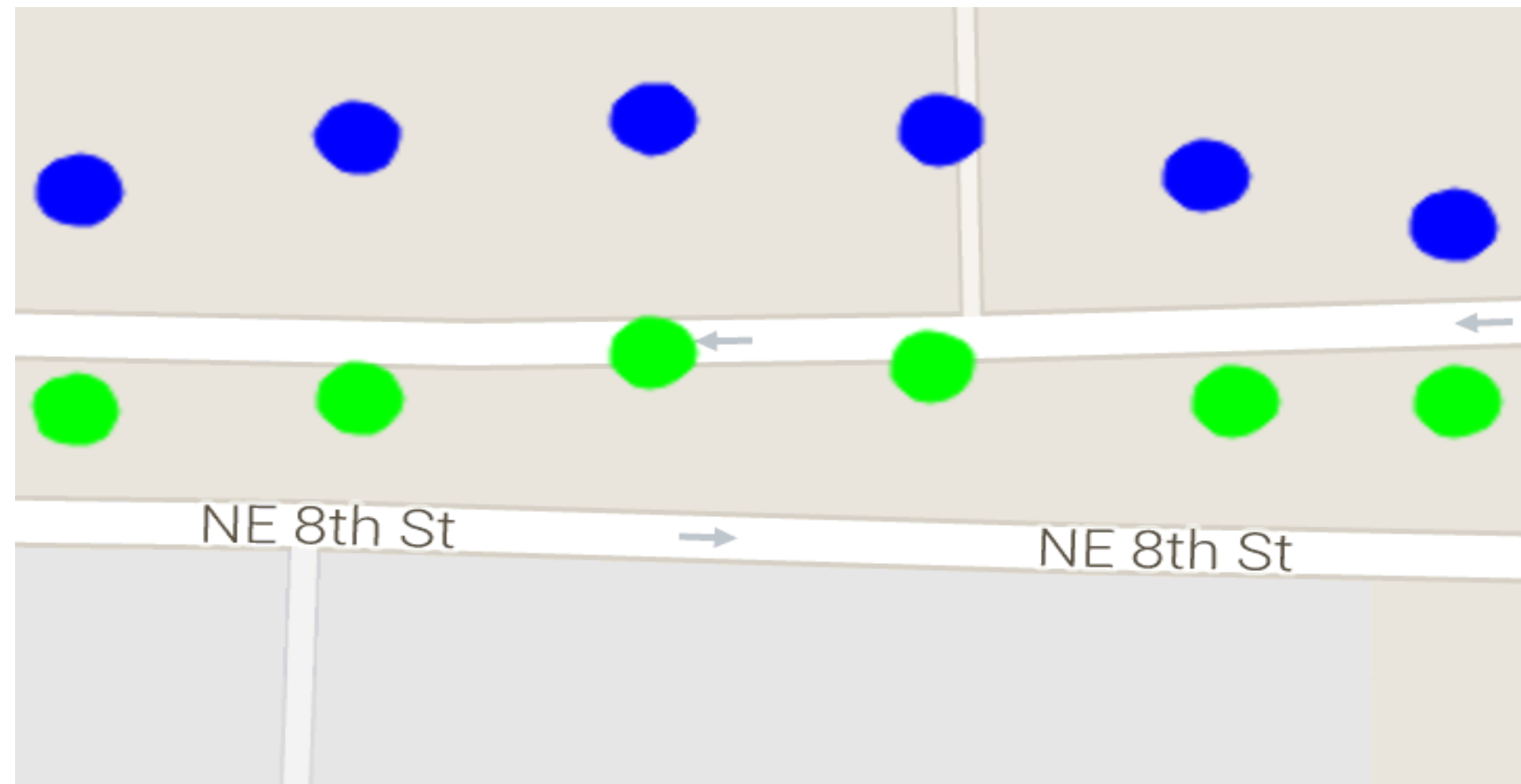
GPS Driving Application



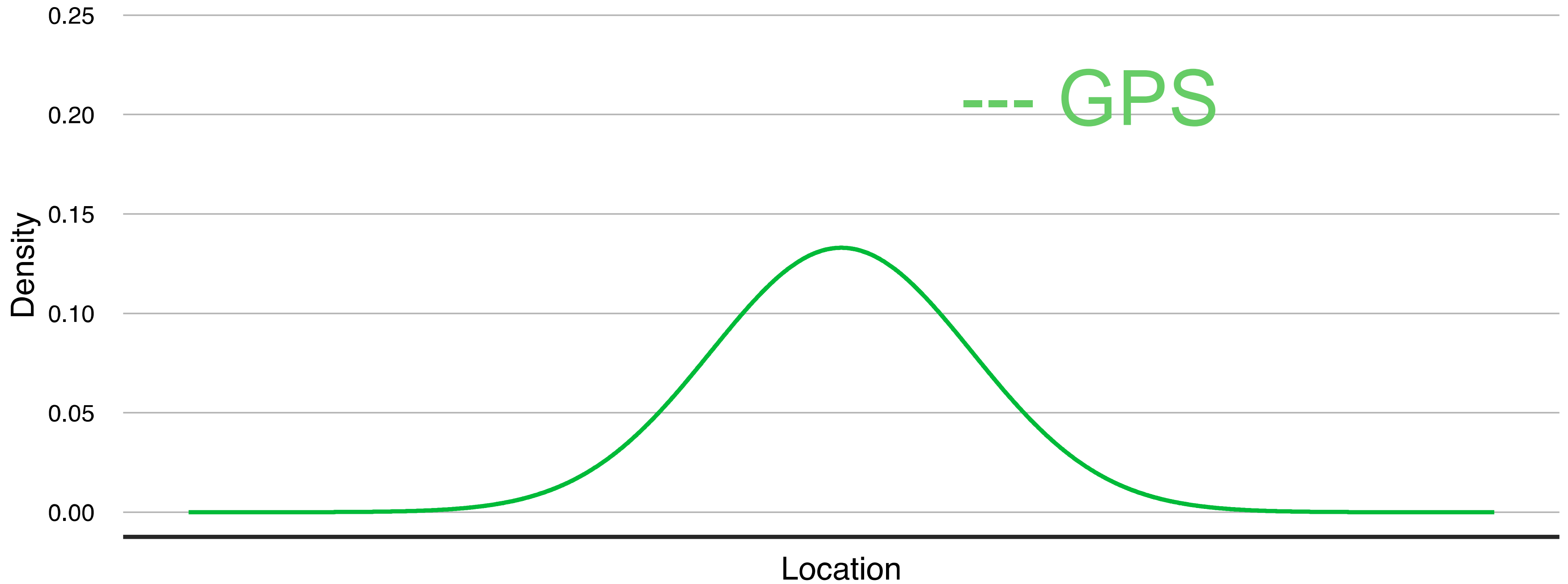
Driving Application

- GPS
- Location = GPS # roads

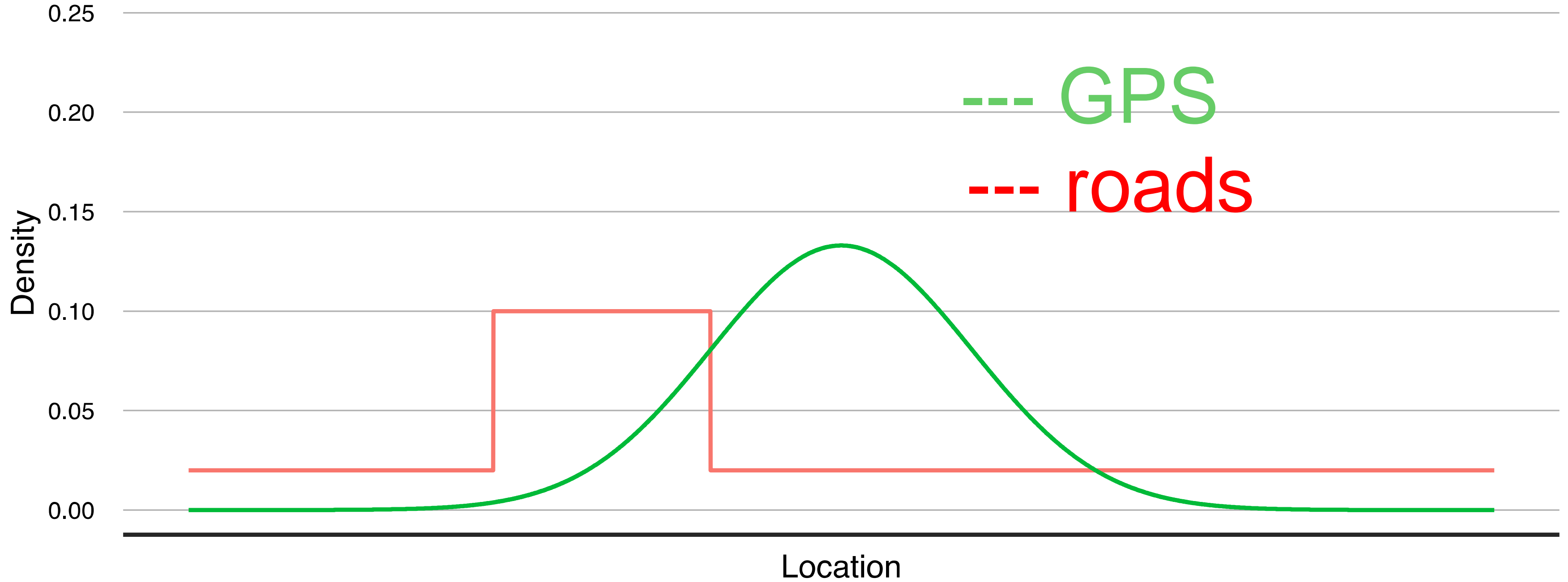
Driving on a road (or not!)



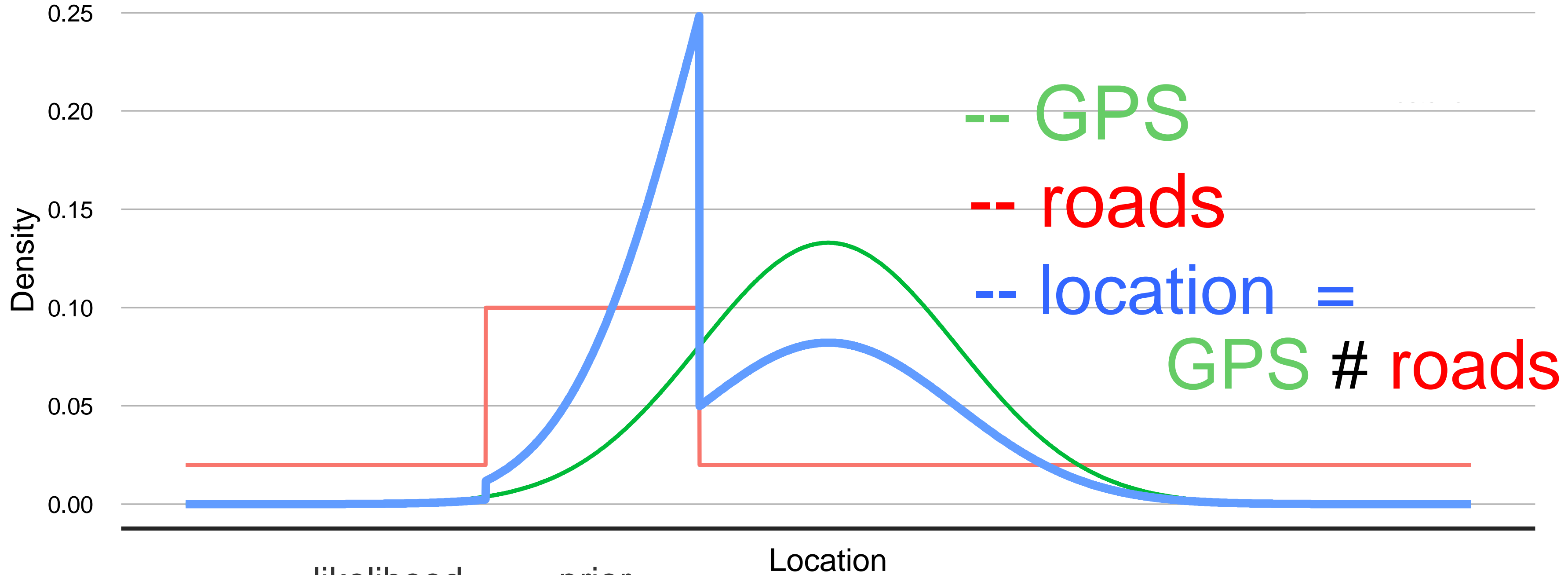
Adding Domain Knowledge



Adding Domain Knowledge



Adding Domain Knowledge



posterior likelihood prior

$$\Pr[H|E] = \frac{\Pr[E|H] \Pr[H]}{\Pr[E]}$$

Uncertain $\langle T \rangle$

Programming with estimates

Thank you