Uncertain<T>: Porting library from C# to python and Cpp

Team:
Hammam wahab
Krishnateja Nallanukala
Manoj kumar Murugan
Vishnu Vardhan Dadi

Customer:
Deebul Nair

Uncertain<T>:

Uncertainty - difference between the estimate and the true value.

- Sensor data is uncertain.
- Many developers simply ignore them or sample a value from a distribution which is also a estimate.
- Developers have to avoid :
  - Using estimates as facts - contain random noise
  - Computation compound errors - decreases accuracy.
  - Using conditionals/ boolean question.
- Ignoring this creates three types of uncertainty bugs:
  - Random errors.
  - Compounding errors.
  - False positives and negatives.

Fig.1 Li, L. (2017). Spatial data uncertainty. *The Geographic Information Science & Technology Body of Knowledge* (4th Quarter 2017 Edition), John P. Wilson (ed). DOI: 10.22224/gistbok/2017.4.4

Uncertain <T> :(contd..)

Using Gaussian distribution for the approximation of the uncertain data is poor technique. This is because,

- From the adjacent figure, the uncetaindata is expressed in a gaussian distribution and a value is sampled.
- We can see that the selected estimate value ‘a’ is not close to the true value.
- In the cases like GPS data the, standard deviation will be very large and choosing a estimate from such distribution will introduce uncertainity.

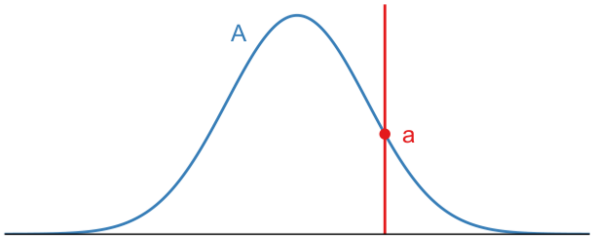


Fig2. James Bornholt, Todd Mytkowicz, Kathryn S. McKinley, Uncertain<T> : A First-Order Type for Uncertain Data, Architectural Support for Programming Languages and Operating Systems

Claims Made by Uncertain <T> & Their Demo

- **Claim-1:** Uncertain<T> improves accuracy of expressiveness of computations.
- **Case study:** GPS example.

- **Claim-2:** It makes use of prior knowledge to minimize random noise in digital sensors.
- **Case study:** Conway’s game of life

Case Study-1: GPS Example

GPS Sensor

- GPS sensors provide “estimated” location. [estimate = true value + uncertainty]

Motivation: Why consider uncertainty?

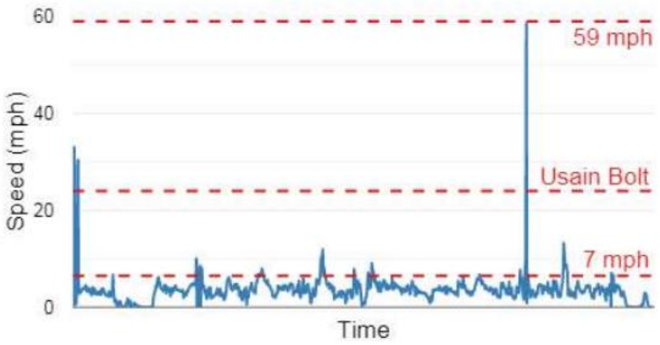
- Ignoring it leads to erroneous estimates insisting the user to walk through walls or drive through water.

(a) 95% CI, $\sigma = 33$ m

(b) 68% CI, $\sigma = 39$ m

Case Study-1: GPS Example

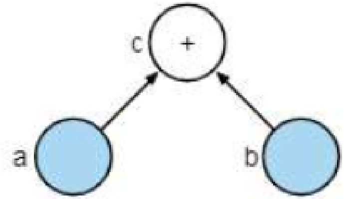
- **Effect:** Computation on estimates compounds uncertainty.
- **Example:** Calculate speeds using position estimates from GPS while walking.
- **Consequence:** Produces illogical results.



It is seen that at some instances the calculated speed is dramatically high.

Case Study-1: What Uncertain<T> Does?

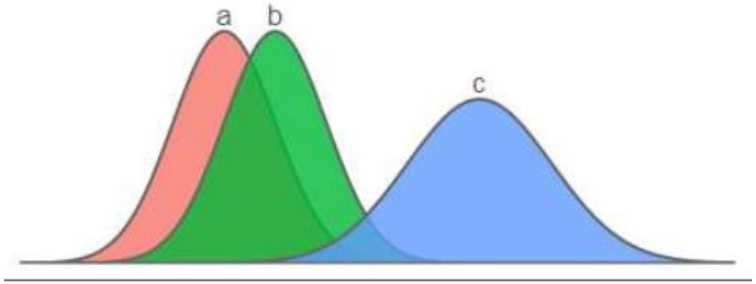
- **On a detailed level:**
  - It encapsulates random variable to include the measure of uncertainty.
  - It defines algebra which operates on random variables such that it propagates uncertainty through calculations leading to better estimates.
  - The operations over random variables are represented and carried out by creating Bayesian network.
  - The Bayesian network is created at runtime and evaluated using conditionals and hypothesis tests.
  - This ensures high performance.



An example Bayesian Network that sums two random variables ‘a’ & ‘b’ to produce ‘c’.

Case Study-1: What Uncertain<T> Does?

- **On a broader level:**
  - Compounds uncertainty.



c=a+b: The uncertainty in ‘c’ is much greater than that of ‘a’ and ‘b’ in other words it is compounded.

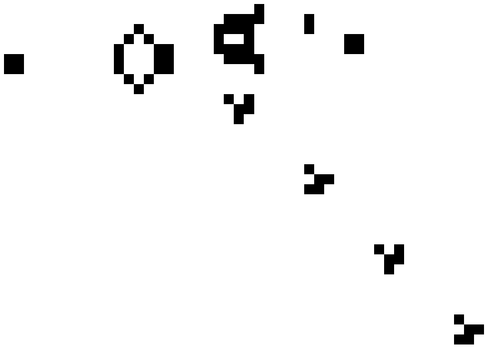
Case Study-1: ...and how does that help?

- Enables to reason about false positives and negatives and thereby make better decisions.
- **Example: Fitness application**
  - That insists users to walk faster than say 4mph.
  - When speed is lesser than 4mph the app alerts the users to walk faster.
  - When the user's speed is better estimated with uncertainty taken into account, the app would alert the user only when it detects with a high confidence that the user is walking with a speed lesser than 4mph.

- This therefore proves claim-1! √

Case Study-2: Conway’s Game of Life

- **The game**
  - Considers world to be an infinite 2 dimensional grid.
  - The 2D grid consist of cells.
  - A cell is either alive or dead.
  - The status of a cell depends on the status of its neighbouring cell.
- **Rules**
  - A live cell with fewer than two live neighbours dies. (Underpopulation)
  - A live cell with two or three live neighbours lives.
  - A live cell with more than three live neighbours dies (Overpopulation)
  - A dead cell with exactly three live neighbours becomes alive (Reproduction)
- **Significance**
  - Generates interesting patterns



Generated patterns

Case Study-2: How is it Related to Uncertainty?

- **Uncertainty comes into picture**
  - Each cell has 8 neighbouring cells.
  - A cell checks the status of its neighbors using digital sensors.
- **In Ideal Case**
  - A sensor outputs either 0 (if dead) or 1 (if alive).
- **In Practical Case**
  - A sensor outputs a real number!
  - This is due to noise say a zero mean Gaussian noise.
  - Therefore sensor output = ideal output + zero mean Gaussian noise
  - **Mathematically: $v = s + G(0, \sigma)$**
  - **s ->ideal output**
  - **G -> Gaussian noise**
  - **v-> actual output**

Case Study-2: Conway’s Game of Life

Game with noisy sensor:

- When the output of the noisy sensor is considered as facts i.e ignoring uncertainty causes incorrect cell status.
- Practically the cells decay faster.

Case Study-2: Conway's Game of Life

Game with noisy sensor: Improved Decision Using prior knowledge

- The nature of noise is known : zero mean Gaussian noise.
- Better decisions can be made using Bayes's rule.
- Probability of the nearby cell being alive:
  - $\mathbf{P(H_0|v) = P(v|H_0)xP(H_0) / Normalizer}$
- Probability of the nearby cell being dead:
  - $\mathbf{P(H_1|v) = P(v|H_1)xP(H_1) / Normalizer}$
- Where, **$P(H_0)$, and $P(H_1)$ are priors and $P(v|H_0)$, and $P(v|H_1)$ are likelihood functions.**

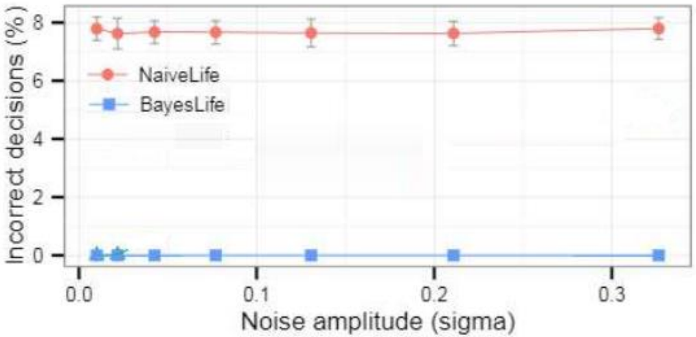
Case Study-2: Conway's Game of Life

Game with noisy sensor: Improved Decision Using prior knowledge

- $P(v|H_0)$ is a gaussian function centered at mean of 0.
- $P(v|H_1)$ is a gaussian function centered at mean of 1.

Case Study-2: ...and how does that help?

- Zero incorrect decision at all noise levels!
- This therefore proves claim-2! √



Why porting in python and Cpp?

- Python is very popular programming language, and there is a need for this library in python, as many open source projects use python.
- Another popular language is Cpp, in which there is no such library that deals with uncertain data exist.
- Porting this library in python and Cpp can be very useful for the developers who develop API which use noisy uncertain data.

Deliverables

- Development of an Uncertain <T> datatype.
- Implementation of data types for GPS and Game of life cases studies.
- Evaluation of two case studies.

Timeline

	Project Tasks	From	To	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
1	Uncertain <T> with tests on game of life	12/14/20	12/21/20					
2	Final version of Game of life	1/4/21	1/10/21					
3	GPS code without Uncertain <T>	1/11/21	1/17/21					
4	Update Uncertain<T> to include GPS	1/18/21	1/24/21					
5	Evaluation of both case studies	1/25/21	1/31/21					