CIS501 – Lecture 7

Woon Wei Lee
Fall 2013, 10:00-11:15am,
Sundays and Wednesdays

Masdar INSTITUTE

For today:

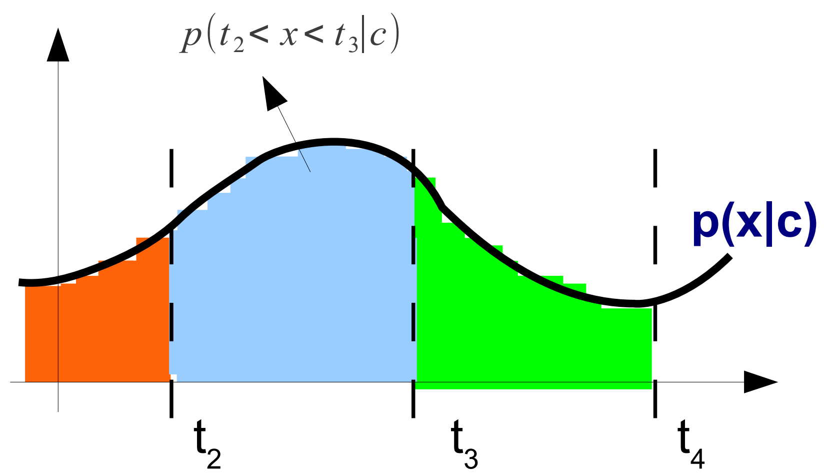
- Administrative stuff
    - Lab (next week!)
    - Project
- NBC wrap-up
    - Discretization techniques
- Assorted topics
    - Model selection
    - Evaluating classifiers

NBC for continuous data

- **We've only seen a couple cases:**
  - Spam classification, cooking example
  - These are examples of discrete features
- **Extend NBC to handle numeric/continous features**
  - Solution: *feature discretization*
  - Basic idea – divide each input feature into two or more bins
  - p(x_i|c) can now take multiple values:

$$p(x|c) \in \{p(x < t_1|c), p(t_1 < x < t_2|c), \ldots, p(t_n < x < t_{n+1}|c)\}$$

  - (n → number of bins, t_i → *i*th threshold)
  - Distribution should be relatively smooth for this to work (why?)



Discretization strategies

- **Problem is now → determine the t_i's**
  - Various strategies, we will study a few examples

1. **Equal Width Discretization (EWD)**
   - Divide axis into bins of equal size (histogram approach).
   - p(f|c) for each bin is proportional to number of points
   - “Alright” but has the same shortcomings as we discussed before
2. **Equal Frequency Discretization (EFD)**
   - Division of axis into bins where each bin has equal number of points
   - Allows scaling of bin-sizes to fit data density.
   - (Analogous to k-NN approach)

Masdar INSTITUTE

Continued

3. Fuzzy discretization (FD)

- Uses a fuzzy assignment scheme to generate the likelihood terms
- Procedure as follows:

  i. Form *k* equally spaced bins (like with EWD)

  ii. However, for a given bin $[t_i, t_{i+1})$, the corresponding likelihood term includes contributions from every training instance:

$$p(t_i \leqslant x < t_{t+1} | c; v) = \int_{t_i}^{t_{i+1}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\left\{\frac{x-v}{\sigma}\right\}^2} dx$$

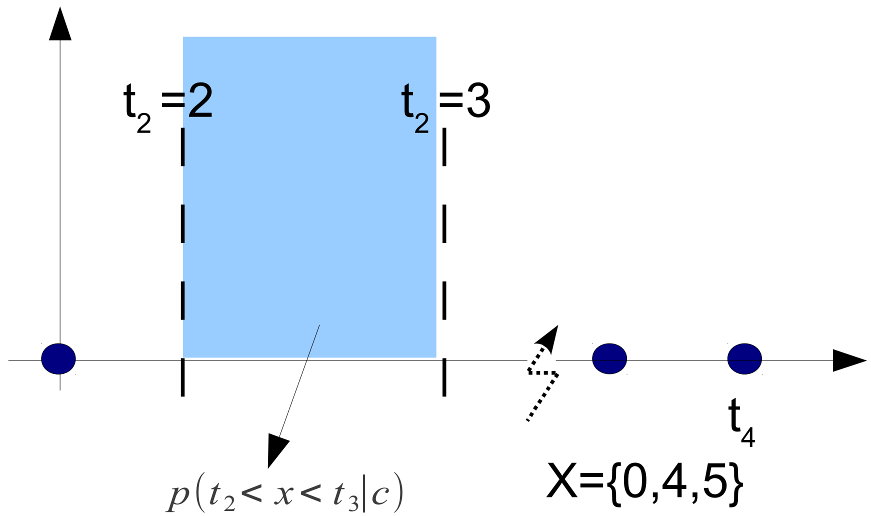
  (*v* is the location of the training instance)

  iii. Finally, calculate p(*f*|*c;v*) as follows:

$$p(t_i \leqslant x < t_{t+1} | c) = \sum_{v \in V} p(t_i \leqslant x < t_{t+1} | c; v)$$

  (*V* is the set of all training points)

Example..



- **EWD:**

  $$p(t_2 < x < t_3 | c) = 0$$

  (probably) unsatisfactory

- **EFD**

  $$p(t_2 < x < t_3 | c) > 0 \, (some\, value)$$

  - Better but you have problem of poor resolution for low density areas, etc..

- **FD:**

  - Assuming σ = 1, this can be calculated using:

$$p(t_2 < x < t_3 | c) = \sum_{v \in \{0,4,5\}} p(t_2 < x < t_3 | c ; v)$$

$$= \frac{1}{2} \sum_{v \in \{0,4,5\}} \left| erf\left(\frac{3-v}{\sqrt{2}}\right) - erf\left(\frac{2-v}{\sqrt{2}}\right) \right|$$

$$= \frac{1}{2} [0.00428 + 0.27181 + 0.00428]$$

$$= 0.17871$$

<u>Technical Note</u>

- erf → "error function", defined as:

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_{x=0}^{z} e^{-x^2} dx$$

- Hence to find Gaussian integral:

$$\frac{1}{\sigma\sqrt{2\pi}} \int_{x=0}^{z} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx$$

(change of variable) $$= \frac{1}{\sigma\sqrt{2\pi}} \int_{x=0}^{\frac{x-\mu}{\sigma\sqrt{2}}} e^{-t^2} . (\sigma\sqrt{2}\, dt)$$

$$= \frac{1}{2} erf\left(\left[\frac{x-\mu}{\sigma\sqrt{2}}\right]\right)$$

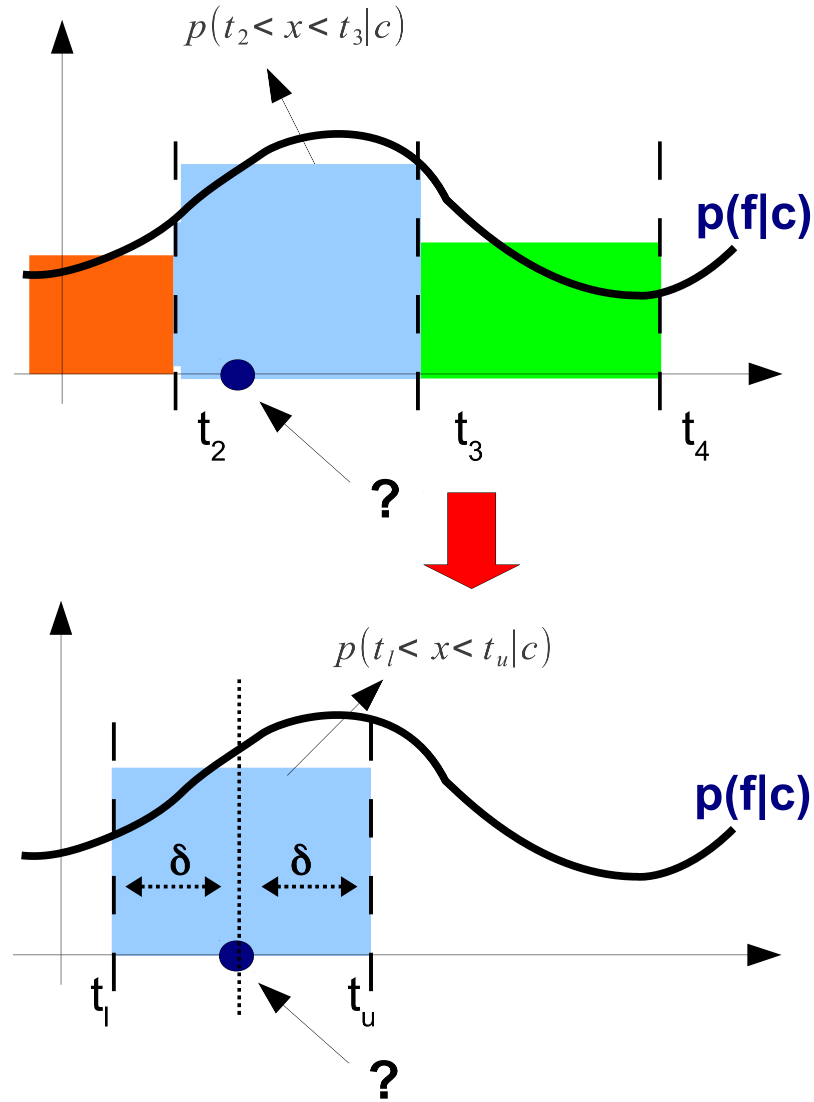
Masdar INSTITUTE

“Flexible threshold” strategies..

- All of the three previous techniques use fixed thresholds
- If the query value is at the end of a bin, this could reduce accuracy
- Let's look at two more techniques which address this:

4. Lazy discretization (LD)

- With LD, the determination of p(x|c) is “postponed”
- When query point *x* is presented, place it at center of bin
  - Set the thresholds at [x-δ,x-δ]
  - p(x|c) is proportional to the number of training instances within these threshold values.
  - δ is set as in the case of EWD.



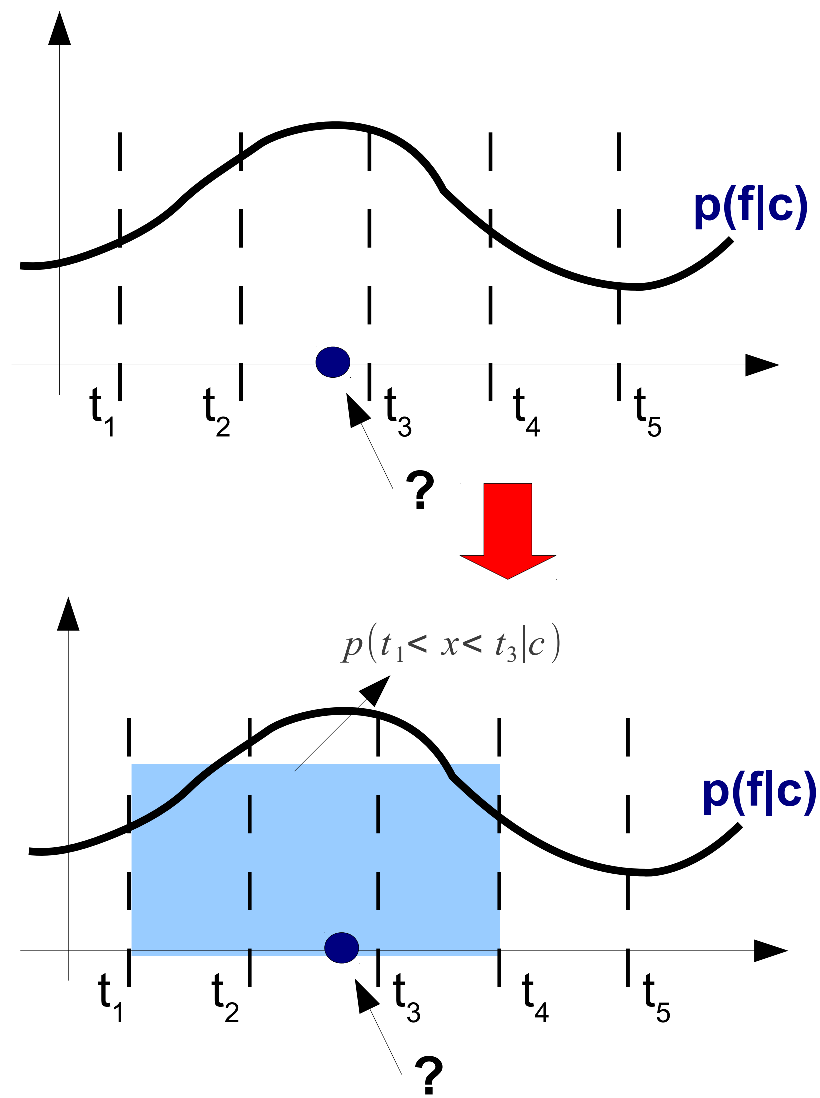
Non-disjoint discretization

5. Non-disjoint discretization (NDD)

- LD is good but high memory requirements.
- NDD → set bins in advance, but are overlapping

  * Query point is always close to center!
- In practice, create a set of “atomic bins” using EFD

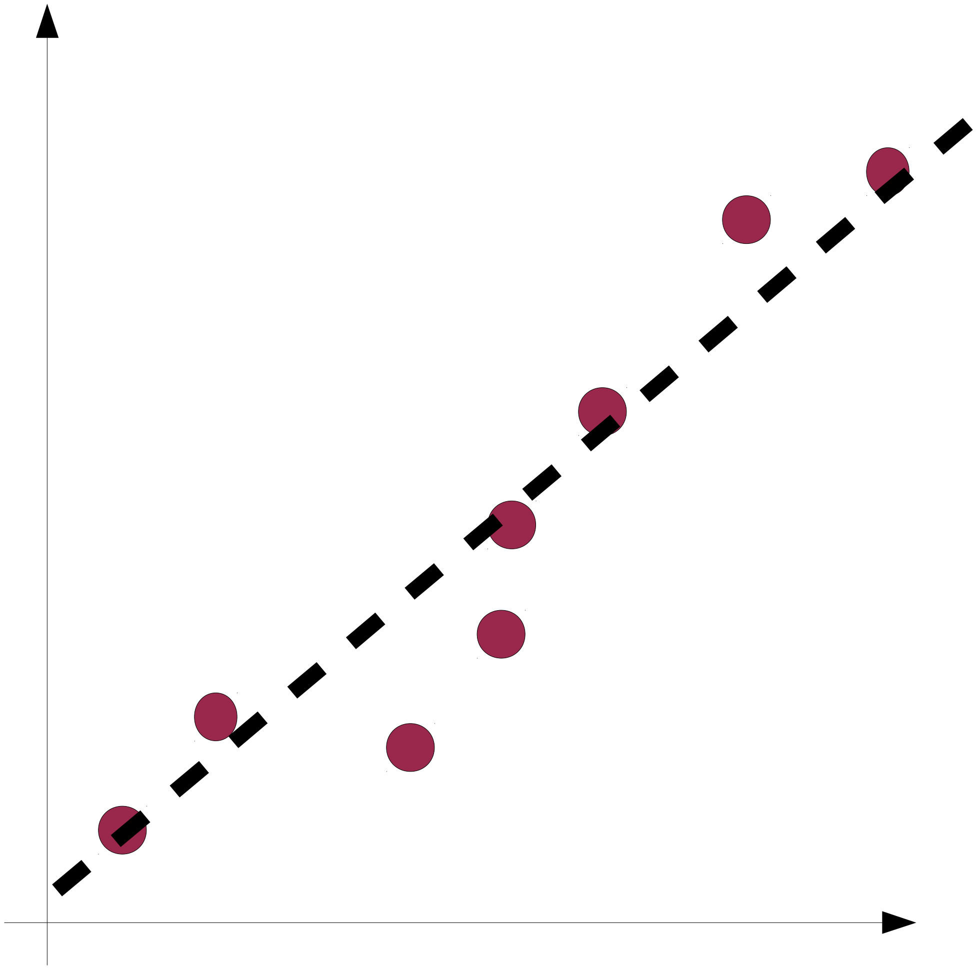
  (like normal bins but smaller)
- When query point is received:
  - Say it falls in atomic bin *i*
  - Actual bin → combination of atomic bins *i*-1,*i* and *i*+1.



Masdar INSTITUTE

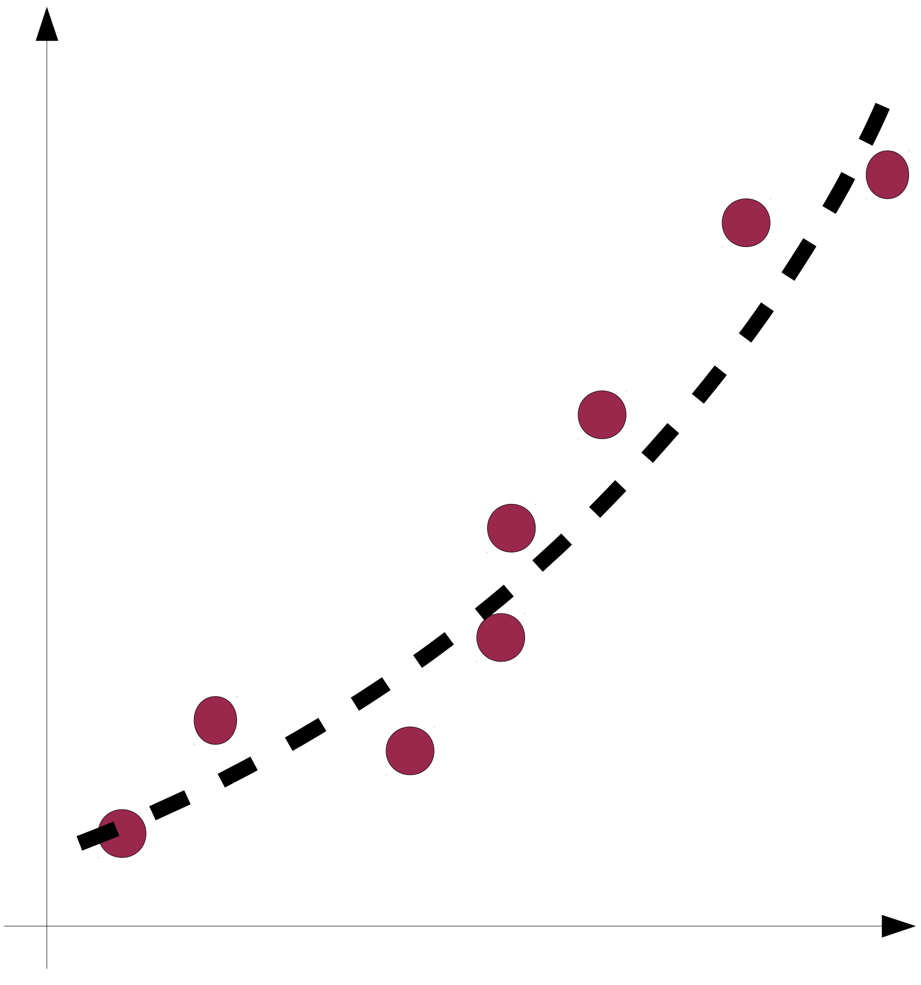
Model selection

- Frequently, there is more than one model that can fit a data set..
- How do we choose any one model over the other?
  - Accuracy of fit/minimum error?
  - (Above) will always favour high complexity models!
  - Overfitting!
- Occam's Razor!
  - Preference for simpler models over complex
- A ***fundamental*** principle in Science/Engineering



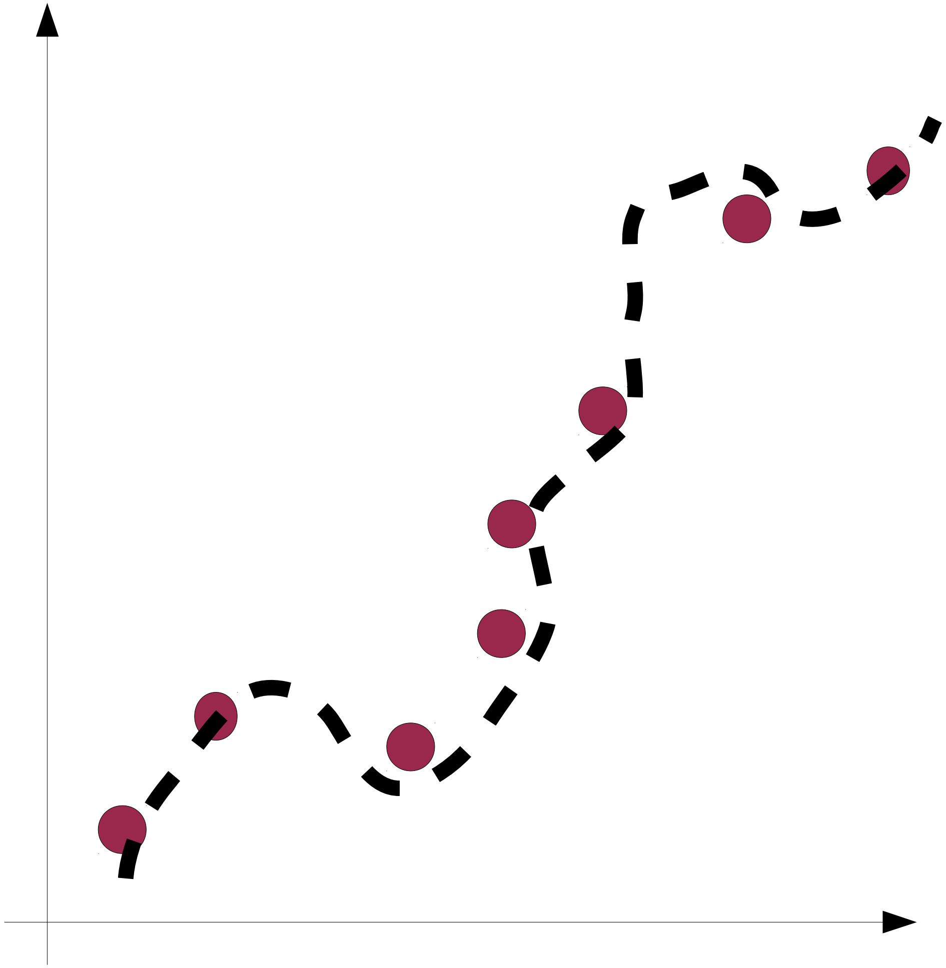
Model selection

- Frequently, there is more than one model that can fit a data set..
- How do we choose any one model over the other?
    - Accuracy of fit/minimum error?
    - (Above) will always favour high complexity models!
    - Overfitting!
- Occam's Razor!
    - Preference for simpler models over complex
- A ***fundamental*** principle in Science/Engineering



Model selection

- Frequently, there is more than one model that can fit a data set..
- How do we choose any one model over the other?
  - Accuracy of fit/minimum error?
  - (Above) will always favour high complexity models!
  - Overfitting!
- Occam's Razor!
  - Preference for simpler models over complex
- A ***fundamental*** principle in Science/Engineering



Evaluating classifiers...

- As I've noted before, you can always build a classifier to classify anything..!
  - (e.g. lottery numbers, stock markets, etc...)
- Critical step in the process → evaluation!
  - i.e. how do you know if your classifier is doing well..
- There are a number of methods but in general we would like to test some notion of the *correctness* of the classifier.
- For a classifier, this is normally in terms of classification accuracy.
- Loosely defined as:

$$Accuracy\,(\%) = \frac{|\{\,Correctly\ classified\ objects\,\}|}{|\{\,Total\ number\ of\ objects\,\}|} \times 100$$

Evaluating classifiers (Cont'd)

- However, this is rather general, and may miss details
- Alternative performance metrics (mostly borrowed from *information retrieval)*:

$$\text{Precision}(\%) = \frac{|\{\mathit{True\ Positives}\}|}{|\{\mathit{True\ Positives} + \mathit{False\ Positives}\}|} \times 100$$

$$\text{Recall}(\%) = \frac{|\{\mathit{True\ Positives}\}|}{|\{\mathit{True\ Positives} + \mathit{False\ Negatives}\}|} \times 100$$

$$\text{Fall-out}(\%) = \frac{|\{\mathit{True\ Negatives}\}|}{|\{\mathit{True\ Negatives} + \mathit{False\ Positives}\}|} \times 100$$

$$\text{F-Measure} = \frac{2 \times (\mathit{Precision} \times \mathit{Recall})}{(\mathit{Precision} + \mathit{Recall})}$$

- Another commonly used tool is a "confusion matrix":

	c_1	c_2	c_3
c_1	99	1	3
c_2	2	92	5
c_3	5	1	95

Terminology Alert!

- Precision ↔ *Specificity,*
- Recall ↔ *Sensitivity, Hit Rate*
- F-Measure ↔ *F_1 Score*
- False Positives ↔ *Type I Error*
- False Negatives ↔ *Type II Error*

Masdar INSTITUTE