Data Analytics

Course Taught at IIFT

Day 11: Similarity Measures and Sensitivity Analysis

Dr. Tanujit Chakraborty

www.ctanujit.org

Today's discussion…

- Similarity and dissimilarity measures
- Sensitivity Analysis
  - Estimation Strategy
  - Accuracy Estimation
  - Performance Estimation

INTRODUCTION TO SIMILARITY MEASURES

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.
Webster's Dictionary

Similarity is hard to define, but…
"*We know it when we see it*"

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Similarity and Dissimilarity Measures

- In clustering techniques, similarity (or dissimilarity) is an important measurement.
- Informally, similarity between two objects (e.g., two images, two documents, two records, etc.) is a numerical measure of the degree to which two objects are alike.
- The dissimilarity on the other hand, is another alternative (or opposite) measure of the degree to which two objects are different.
- Both similarity and dissimilarity also termed as proximity.
- Usually, similarity and dissimilarity are non-negative numbers and may range from zero (highly dissimilar (no similar)) to some finite/infinite value (highly similar (no dissimilar)).

Note:

- Frequently, the term distance is used as a synonym for dissimilarity
- In fact, it is used to refer as a special case of dissimilarity.

Proximity Measures: Single-Attribute

- Consider an object, which is defined by a single attribute A (e.g., length) and the attribute A has n-distinct values $a_1, a_2, \ldots\ldots, a_n$.
- A data structure called "Dissimilarity matrix" is used to store a collection of proximities that are available for all pair of n attribute values.
  - In other words, the Dissimilarity matrix for an attribute A with n values is represented by an $n \times n$ matrix as shown below.

$$\begin{bmatrix} 0 & & & \\ p_{(2,1)} & 0 & & \\ p_{(3,1)} & p_{(3,2)} & 0 & \\ \vdots & \vdots & \vdots & \\ p_{(n,1)} & p_{(n,2)} & \ldots\ldots & 0 \end{bmatrix}_{n \times n}$$

  - Here, $p_{(i,j)}$ denotes the proximity measure between two objects with attribute values a_i and a_j.
- Note: The proximity measure is symmetric, that is, $p_{(i,j)} = p_{(j,i)}$

Proximity Calculation

- Proximity calculation to compute $p_{(i,j)}$ is different for different types of attributes according to NOIR topology.

Proximity calculation for Nominal attributes:

- For example, binary attribute, Gender = {Male, female} where Male is equivalent to binary 1 and female is equivalent to binary 0.
- Similarity value is 1 if the two objects contains the same attribute value, while similarity value is 0 implies objects are not at all similar.

Object	Gender
Ram	Male
Sita	Female
Laxman	Male

- Here, Similarity value let it be denoted by p, among different objects are as follows.

$$p(Ram, sita) = 0$$
$$p(Ram, Laxman) = 1$$

Note : In this case, if q denotes the dissimilarity between two objects i *and* j with single binary attributes, then $q_{(i,j)} = 1 - p_{(i,j)}$

Proximity Calculation

- Now, let us focus on how to calculate proximity measures between objects which are defined by two or more binary attributes.
- Suppose, the number of attributes be b. We can define the contingency table summarizing the different matches and mismatches between any two objects x and y, which are as follows.

Table 1: Contingency table with binary attributes

Object x \ Object y	1	0
1	f_{11}	f_{10}
0	f_{01}	f_{00}

Here, f_{11}= the number of attributes where x=1 and y=1.

f_{10}= the number of attributes where x=1 and y=0.

f_{01}= the number of attributes where x=0 and y=1.

f_{00}= the number of attributes where x=0 and y=0.

Note : $f_{00} + f_{01} + f_{10} + f_{11} = b$, the total number of binary attributes.

Now, two cases may arise: symmetric and asymmetric binary attributes.

Similarity Measure with Symmetric Binary

- To measure the similarity between two objects defined by symmetric binary attributes using a measure called symmetric binary coefficient and denoted as $\mathcal{S}$ and defined below

$$\mathcal{S} = \frac{Number\ of\ matching\ attribute\ values}{Total\ number\ of\ attributes}$$

or

$$\mathcal{S} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

The dissimilarity measure, likewise can be denoted as $\mathcal{D}$ and defined as

$$\mathcal{D} = \frac{Number\ of\ mismatched\ attribute\ values}{Total\ number\ of\ attributes}$$

or

$$\mathcal{D} = \frac{f_{01} + f_{10}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

Note that, $\mathcal{D} = 1 - \mathcal{S}$

Similarity Measure with Symmetric Binary

Example 1: Proximity measures with symmetric binary attributes

Consider the following two dataset, where objects are defined with symmetric binary attributes.

Gender = {M, F}, Food = {V, N}, Caste = {H, M}, Education = {L, I},
Hobby = {T, C}, Job = {Y, N}

Object	Gender	Food	Caste	Education	Hobby	Job
Hari	M	V	M	L	C	N
Ram	M	N	M	I	T	N
Tomi	F	N	H	L	C	Y

$$\mathcal{S}(\text{Hari, Ram}) = \frac{1+2}{1+2+1+2} = 0.5$$

Proximity Measure with Asymmetric Binary

- Such a similarity measure between two objects defined by asymmetric binary attributes is done by Jaccard Coefficient and which is often symbolized by $\mathcal{J}$ is given by the following equation

$$\mathcal{J}=\frac{Number\ of\ matching\ presence}{Number\ of\ attributes\ not\ involved\ in\ 00\ matching}$$

or

$$\mathcal{J}=\frac{f_{11}}{f_{01}+f_{10}+f_{11}}$$

Proximity Measure with Asymmetric Binary

Example 2: Jaccard Coefficient

Consider the following two dataset.

Gender = {M, F}, Food = {V, N}, Caste = {H, M}, Education = {L, I},
Hobby = {T, C}, Job = {Y, N}

Calculate the Jaccard coefficient between Ram and Hari assuming that all binary attributes are asymmetric and for each pair values for an attribute, first one is more frequent than the second.

Object	Gender	Food	Caste	Education	Hobby	Job
Hari	M	V	M	L	C	N
Ram	M	N	M	I	T	N
Tomi	F	N	H	L	C	Y

$$\mathcal{J}(\text{Hari, Ram}) = \frac{1}{2+1+1} = 0.25$$

Note: $\mathcal{J}(\text{Ram, Tomi}) = 0$ and $\mathcal{J}(\text{Hari, Ram}) = \mathcal{J}(\text{Ram, Hari})$, etc.

Example 3:

Consider the following two dataset.

Gender = {M, F}, Food = {V, N}, Caste = {H, M}, Education = {L, I},
Hobby = {T, C}, Job = {Y, N}

Object	Gender	Food	Caste	Education	Hobby	Job
Hari	M	V	M	L	C	N
Ram	M	N	M	I	T	N
Tomi	F	N	H	L	C	Y

? How you can calculate similarity if Gender, Hobby and Job are symmetric binary attributes and Food, Caste, Education are asymmetric binary attributes?

Obtain the similarity matrix with Jaccard coefficient of objects for the above, e.g.

$$\mathcal{J} = \begin{array}{c} \\ H \\ R \\ T \end{array} \begin{array}{c} \begin{array}{ccc} H & R & T \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ \mathcal{J}(R,H) & 0 & 0 \\ \mathcal{J}(T,H) & \mathcal{J}(T,R) & 0 \end{bmatrix} \end{array}$$

Proximity Measure with Categorical Attribute

- Binary attribute is a special kind of nominal attribute where the attribute has values with two states only.
- On the other hand, categorical attribute is another kind of nominal attribute where it has values with three or more states (e.g. color = {Red, Green, Blue}).
- If $s(x, y)$ denotes the similarity between two objects x *and* y, then

$$s(x, y) = \frac{Number\ of\ matches}{Total\ number\ of\ attributes}$$

and the dissimilarity $d(x, y)$ is

$$d(x, y) = \frac{Number\ of\ mismatches}{Total\ number\ of\ attributes}$$

- If m = number of matches and a = number of categorical attributes with which objects are defined as

$$s(x, y) = \frac{m}{a} \quad \text{and} \quad d(x, y) = \frac{a - m}{a}$$

Proximity Measure with Categorical Attribute

Example 4:

Object	Color	Position	Distance
1	R	L	L
2	B	C	M
3	G	R	M
4	R	L	H

The similarity matrix considering only color attribute is shown below

$$s = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Dissimilarity matrix, $d =$?

Obtain the dissimilarity matrix considering both the categorical attributes (i.e. color and position).

Proximity Measure with Ordinal Attribute

- Ordinal attribute is a special kind of categorical attribute, where the values of attribute follows a sequence (ordering) e.g. Grade = {Ex, A, B, C} where Ex > A >B >C.
- Suppose, A is an attribute of type ordinal and the set of values of $A = \{a_1, a_2, \ldots\ldots, a_n\}$. Let n values of A are ordered in ascending order as $a_1 < a_2 < .. < a_n$. Let *i-th* attribute value a_i be ranked as *i, i=1,2,..n.*
- The normalized value of a_i can be expressed as

$$\hat{a}_i = \frac{i-1}{n-1}$$

- Thus, normalized values lie in the range [0..1].
- As a_i is a numerical value, the similarity measure, then can be calculated using any similarity measurement method for numerical attribute.
- For example, the similarity measure between two objects x *and* y with attribute values a_i and a_j, then can be expressed as

$$s(x, y) = \sqrt{(\hat{a}_i - \hat{a}_j)^2}$$

where $\hat{a}_i$ and $\hat{a}_i$ are the normalized values of $\hat{a}_i$ and $\hat{a}_i$, respectively.

Proximity Measure with Ordinal Attribute

Example 5: Consider the following set of records, where each record is defined by two ordinal attributes size={S, M, L} and Quality = {Ex, A, B, C} such that S<M<L and Ex>A>B>C.

Object	Size	Quality
A	S (0.0)	A (0.66)
B	L (1.0)	Ex (1.0)
C	L (1.0)	C (0.0)
D	M (0.5)	B (0.33)

- Normalized values are shown in brackets.
- Their similarity measures are shown in the similarity matrix below.

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ ? & 0 & 0 & 0 \\ & ? & 0 & 0 \\ & & ? & 0 \end{bmatrix} \end{array}$$

Find the dissimilarity matrix, when each object is defined by only one ordinal attribute say size (or quality).

Proximity Measure with Interval Scale

- The measure called distance is usually referred to estimate the similarity between two objects defined with interval-scaled attributes.
- We first present a generic formula to express distance d between two objects x *and* y in n-dimensional space. Suppose, x_i *and* y_i denote the values of i^{th} attribute of the objects x *and* y respectively.

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

- Here, r is any integer value.
- This distance metric most popularly known as Minkowski metric.
- This distance measure follows some well-known properties. These are mentioned in the next slide.

Proximity Measure with Interval Scale

Properties of Minkowski metrics:

1. Non-negativity:
   a. $d(x, y) \geq 0 \; for \; all \; x \; and \; y$
   b. $d(x, y) = 0$ only if $x = y$. This is also called identity condition.

2. Symmetry:
   $d(x, y) = d(y, x) \;\; for \; all \; x \; and \; y$
   This condition ensures that the order in which objects are considered is not important.

3. Transitivity:
   $d(x, z) \leq d(x, y) + d(y, z) \; for \; all \; x \; , y \; and \; z.$
   - This condition has the interpretation that the least distance $d(x, z)$ between objects $x \; and \; z$ is always less than or equal to the sum of the distance between the objects $x \; and \; y$, and between $y \; and \; z$.
   - This property is also termed as Triangle Inequality.

Proximity Measure with Interval Scale

Depending on the value of r, the distance measure is renamed accordingly.

1. **Manhattan distance (L_1 Norm: $r = 1$)**

   The Manhattan distance is expressed as

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

where $|...|$ denotes the absolute value.

Example: x = [7, 3, 5] and y = [3, 2, 6].

The Manhattan distance is $|7 - 3| + |3 - 2| + |5 - 6| = 6$.

- As a special instance of Manhattan distance, when attribute values $\in [0, 1]$ is called Hamming distance.
- Alternatively, Hamming distance is the number of bits that are different between two objects that have only binary values (i.e. between two binary vectors).

Proximity Measure with Interval Scale

2. Euclidean Distance (L_2 Norm: $r = 2$)

This metric is same as Euclidean distance between any two points $x \; and \; y \; in \; \mathcal{R}^n$.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Example: x = [7, 3, 5] and y = [3, 2, 6].

The Euclidean distance between $x \; and \; y$ is

$$d(x, y) = \sqrt{(7-3)^2 + (3-2)^2 + (5-6)^2} = \sqrt{18} \approx 2.426$$

Proximity Measure with Interval Scale

3. Chebychev Distance ($L_\propto$ Norm: $r \in \mathcal{R}$)

This metric is defined as

$$d(x, y) = \max_{\forall i}\{|x_i - y_i|\}$$

- We may clearly note the difference between Chebychev metric and Manhattan distance. That is, instead of summing up the absolute difference (in Manhattan distance), we simply take the maximum of the absolute differences (in Chebychev distance). Hence, $L_\propto < L_1$

Example: x = [7, 3, 5] and y = [3, 2, 6].

The Manhattan distance = $|7 - 3| + |3 - 2| + |5 - 6| = 6.$

The chebychev distance = $\text{Max}\{|7 - 3|, |3 - 2|, |5 - 6|\} = 4.$

Proximity Measure with Interval Scale

4. Canberra metric:

$$d(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{(|x_i| + |y_i|)^q}$$

- where q is a real number. Usually $q = 1$, because numerator of the ratio is always $\leq$ denominator, the ratio ≤ 1, that is, the sum is always bounded and small.
- If $q \neq 1$, it is called Fractional Canberra metric.
- If $q > 1$, the oppositive relationship holds.

5. Hellinger metric:

$$d(x,y) = \sum_{i=1}^{n} (\sqrt{x_i} - \sqrt{y_i})^2$$

This metric is then used as either squared or transformed into an acceptable range [-1, +1] using the following transformations.

i. $d(x,y) = (1 - r(x,y))/2$

ii. $d(x,y) = 1 - r(x,y)$

Where $r(x,y)$ is correlation coefficient between x *and* y.

Note: Dissimilarity measurement is not relevant with distance measurement.

Proximity Measure for Ratio-Scale

The proximity between the objects with ratio-scaled variable can be carried with the following steps:

1. Apply the appropriate transformation to the data to bring it into a linear scale. (e.g. logarithmic transformation to data of the form $X = Ae^B$.

2. The transformed values can be treated as interval-scaled values. Any distance measure discussed for interval-scaled variable can be applied to measure the similarity.

Note:

There are two concerns on proximity measures:

- Normalization of the measured values.
- Intra-transformation from similarity to dissimilarity measure and vice-versa.

Proximity Measure for Ratio-Scale

Normalization:

- A major problem when using the similarity (or dissimilarity) measures (such as Euclidean distance) is that the large values frequently swamp the small ones.
- For example, consider the following data.

Make	Cost 1	Cost 2	Cost 3
X	2,00,000	70	10
Y	2,50,000	100	5

Here, the contribution of Cost 2 and Cost 3 is insignificant compared to Cost 1 so far the Euclidean distance is concerned.

- This problem can be avoided if we consider the normalized values of all numerical attributes.

- Another normalization may be to take the estimated values in a normalized range say [0, 1]. Note that, if a measure varies in the range, then it can be normalized as

$$s' = \frac{1}{1+s} \text{ where } s \in [0..\infty]$$

Proximity Measure for Ratio-Scale

Intra-transformation:

- Transforming similarities to dissimilarities and vice-versa is also relatively straightforward.
- If the similarity (or dissimilarity) falls in the interval [0..1], the dissimilarity (or similarity) can be obtained as

$$d = 1 - s$$

or

$$s = 1 - d$$

- Another approach is to define similarity as the negative of dissimilarity (or vice-versa).

Proximity Measure with Mixed Attributes

- The previous metrics on similarity measures assume that all the attributes were of the same type. Thus, a general approach is needed when the attributes are of different types.
- One straightforward approach is to compute the similarity between each attribute separately and then combine these attribute using a method that results in a similarity between 0 and 1.
- Typically, the overall similarity is defined as the average of all the individual attribute similarities.
- See the algorithm in the next slide for doing this.

Similarity Measure with Vector Objects

Suppose, the objects are defined with $A_1, A_2, \ldots\ldots, A_n$ attributes.

1. For the *k-th* attribute (k = 1, 2, . . , n), compute similarity $s_k(x, y)$ in the range [0, 1].
2. Compute the overall similarity between two objects using the following formula

$$similarity\ (x, y) = \frac{\sum_{i=1}^{n} s_i(x, y)}{n}$$

3. The above formula can be modified by weighting the contribution of each attribute. If the weight w_k is for the *k-th* attribute, then

$$w\_similarity\ (x, y) = \frac{\sum_{i=1}^{n} w_i s_i(x, y)}{n}$$

Such that $\sum_{i=1}^{n} w_i = 1$.

4. The definition of the Minkowski distance can also be modified as follows:

$$d(x, y) = \left(\sum_{i=1}^{n} w_i |x_i - y_i|^r \right)^{\frac{1}{r}}$$

Each symbols are having their usual meanings.

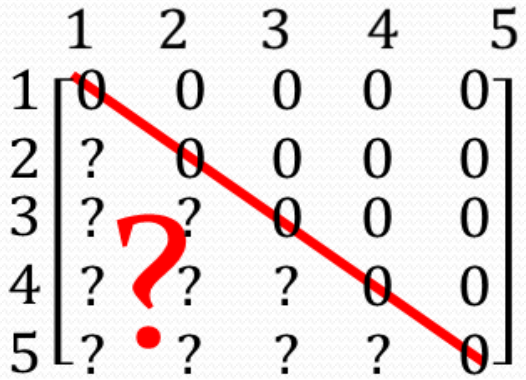
Similarity Measure with Mixed Attributes

Example 6:

Consider the following set of objects. Obtain the similarity matrix.

Object	A (Binary)	B (Categorical)	C (Ordinal)	D (Numeric)	E (Numeric)
1	Y	R	X	475	10^8
2	N	R	A	10	10^{-2}
3	N	B	C	1000	10^5
4	Y	G	B	500	10^3
5	Y	B	A	80	1

[For C: X>A>B>C]



How cosine similarity can be applied to this?

Non-Metric similarity

- In many applications (such as information retrieval) objects are complex and contains a large number of symbolic entities (such as keywords, phrases, etc.).
- To measure the distance between complex objects, it is often desirable to introduce a non-metric similarity function.
- Here, we discuss few such non-metric similarity measurements.

Cosine similarity

Suppose, *x* and *y* denote two vectors representing two complex objects. The cosine similarity denoted as $\cos(x, y)$ and defined as

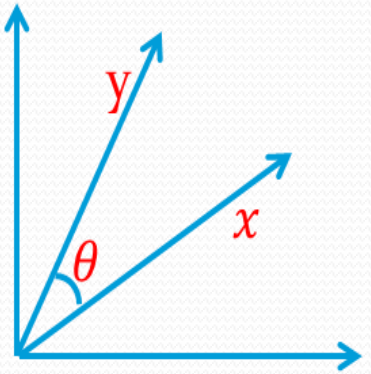
$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

- where $x \cdot y$ denotes the vector dot product, namely $x \cdot y = \sum_{i=1}^{n} x_i \cdot y_i$ such that $x = [x_1, x_2, .., x_n]$ and $y = [y_1, y_2, .., y_n]$.
- $\|x\|$ and $\|y\|$ denote the Euclidean norms of vector *x* and *y*, respectively (essentially the length of vectors *x* and *y*), that is

$$\|x\| = \sqrt{{x_1}^2 + {x_2}^2 + .. + {x_n}^2} \text{ and } \|y\| = \sqrt{{y_1}^2 + {y_2}^2 + .. + {y_n}^2}$$

Cosine Similarity

- In fact, cosine similarity essentially is a measure of the (cosine of the) angle between *x* and *y*.
- Thus if the cosine similarity is 1, then the angle between *x* and *y* is 0° and in this case, *x* and *y* are the same except for magnitude.
- On the other hand, if cosine similarity is 0, then the angle between *x* and *y* is 90° and they do not share any terms.
- Considering, this cosine similarity can be written equivalently



$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{x}{\|x\|} \cdot \frac{y}{\|y\|} = \hat{x} \cdot \hat{y}$$

where $\hat{x} = \frac{x}{\|x\|}$ and $\hat{y} = \frac{y}{\|y\|}$. This means that cosine similarity does not take the magnitude of the two vectors into account, when computing similarity.

- It is thus, one way normalized measurement.

Non-Metric Similarity

Example 7: Cosine Similarity

Suppose, we are given two documents with count of 10 words in each are shown in the form of vectors *x* and *y* as below.

x = [3, 2, 0, 5, 0, 0, 0, 2, 0, 0] and y = [1, 0, 0, 0, 0, 0, 0, 1, 0, 2]

Thus, $x \cdot y = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2$

$= 5$

$\|x\| = \sqrt{3^2 + 2^2 + 0 + 5^2 + 0 + 0 + 0 + 2^2 + 0 + 0} = 6.48$

$\|y\| = \sqrt{1^2 + 0 + 0 + 0 + 0 + 0 + 0 + 1^2 + 0 + 2^2} = 2.24$

$\therefore \quad \cos(x, y) = 0.31$

Extended Jaccard Coefficient

The extended Jaccard coefficient is denoted as EJ and defined as

$$EJ = \frac{x \cdot y}{\|x\|^2 \cdot \|y\|^2 - x \cdot y}$$

- This is also alternatively termed as Tanimoto coefficient and can be used to measure like document similarity.

Compute Extended Jaccard coefficient (EJ) for the above example 7.

Pearson's Correlation

- The correlation between two objects *x* and *y* gives a measure of the linear relationship between the attributes of the objects.
- More precisely, Pearson's correlation coefficient between two objects *x* and *y* is defined in the following.

$$P(x,y) = \frac{S_{xy}}{S_x \cdot S_y}$$

where $S_{xy} = covariance\ (x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$

$$S_x = Standard\ deviation\ (x) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$S_y = Standard\ deviation\ (y) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$\bar{x} = mean\ (x) = \frac{1}{n}\sum_{i=1}^{n} x_i$

$\bar{y} = mean\ (y) = \frac{1}{n}\sum_{i=1}^{n} y_i$ and *n* is the number of attributes in *x* and *y*.

Note 1: Correlation is always in the range of -1 to 1. A correlation of 1(-1) means that *x* and *y* have a perfect positive (negative) linear relationship, that is, $x_i = a \cdot y_i + b$ for some and b.

Example 8: Pearson's correlation

Calculate the Pearson's correlation of the two vectors *x* and *y* as given below.

x = [3, 6, 0, 3, 6]

y = [1, 2, 0, 1, 2]

Note: Vector components can be negative values as well.

Note:

If the correlation is 0, then there is no linear relationship between the attribute of the object.

Example 9: Non-linear correlation

Verify that there is no linear relationship among attributes in the objects *x* and *y* given below.

x = [-3, -2, -1, 0, 1, 2, 3]

y = [9, 4, 1, 0, 1, 4, 9]

$P(x, y) = 0$, and also note $x_i = {y_i}^2$ for all attributes here.

Mahalanobis Distance

- A related issue with distance measurement is how to handle the situation when attributes do not have the same range of values.
- For example, a record with two objects *Age* and *Income*. Here, two attributes have different scales. Thus, Euclidean distance is not a suitable measure to handle such situation.
- In the other way around, how to compute distance when there is correlation between some of the attributes, perhaps, in addition to difference in the ranges of values.
- A generalization of Euclidean distance, the mahalanobi's distance is useful when attributes are (partially) correlated and/or have different range of values.
- The Mahalanobi's distance between two objects (vectors) *x* and *y* is defined as

$$M(x, y) = (x - y)\sum^{-1}(x - y)^T$$

Here, $\sum^{-1}$ is inverse if the covariance matrix.

Set Difference and Time Difference

Set Difference

- Another non-metric dissimilarity measurement is set difference.
- Given two sets *A* and *B*, *A* – *B* is the set of elements of *A* that are not in *B*. Thus, if
  $A = \{1, 2, 3, 4\}$ and $B = \{2, 3, 4\}$ then $A - B = \{1\}$ and $B - A = \emptyset$.
- We can define *d* between two sets as *d(A, B)* as
  $d(A, B) = |A - B|$; where $|A|$ denotes the size of set *A*.

Note:

This measure does not satisfy the property of Non-negativity, Symmetric and Transitivity.

- Another modified definition however satisfies
  $$d(A, B) = |A - B| + |B - A|$$

Time Difference

- It defines the distance between times of the day as follows

$$d(t_1, t_2) = \begin{cases} t_2 - t_1 \ if \ t_1 \leq t_2 \\ 24 + (t_2 - t_1) \ if \ t_1 \geq t_2 \end{cases}$$

- Example: $d(1\ pm, 2\ pm) = 1\ hour$
  $d(2\ pm, 1\ pm) = 23\ hours.$

SENSITIVITY ANALYSIS

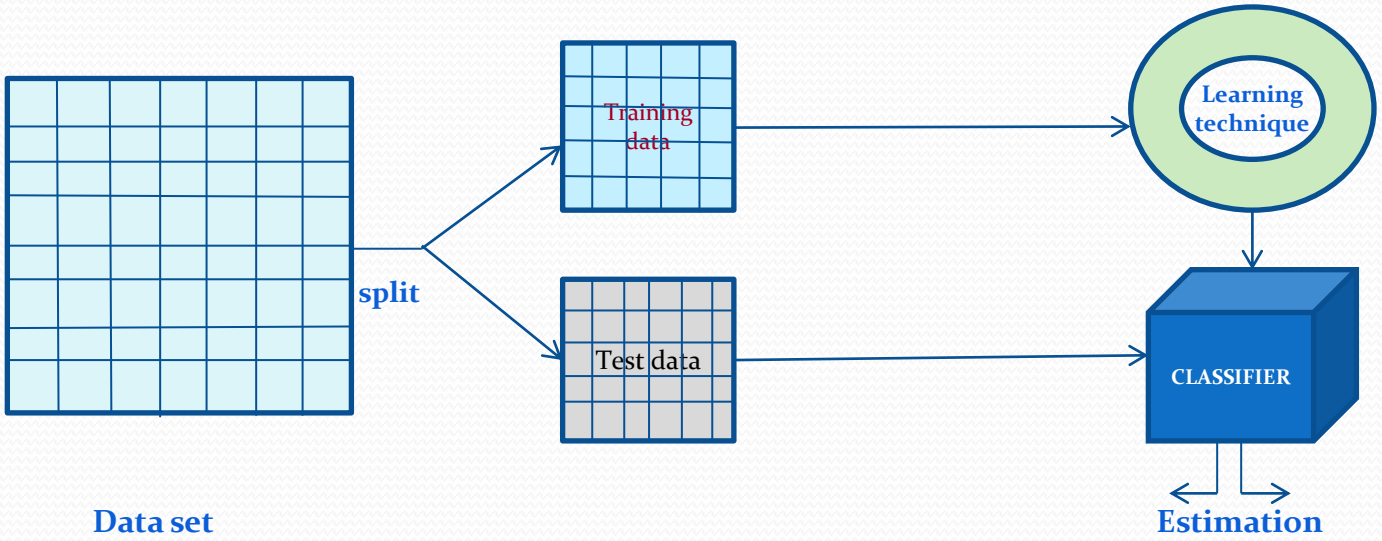
Introduction

- A classifier is used to predict an outcome of a test data
  - Such a prediction is useful in many applications
    - Business forecasting, cause-and-effect analysis, etc.
  - A number of classifiers have been evolved to support the activities.
    - Each has their own merits and demerits
- There is a need to estimate the accuracy and performance of the classifier with respect to few controlling parameters in data sensitivity
- As a task of sensitivity analysis, we have to focus on
  - Estimation strategy
  - Metrics for measuring accuracy
  - Metrics for measuring performance

Estimation Strategy

Planning for Estimation

- Using some "training data", building a classifier based on certain principle is called "learning a classifier".
- After building a classifier and before using it for classification of unseen instance, we have to validate it using some "test data".
- Usually training data and test data are outsourced from a large pool of data already available.



Estimation Strategies

Accuracy and performance measurement should follow a strategy. As the topic is important, many strategies have been advocated so far. Most widely used strategies are

- Holdout method
- Random subsampling
- Cross-validation
- Bootstrap approach

Holdout Method

This is a basic concept of estimating a prediction.

- Given a dataset, it is partitioned into two disjoint sets called training set and testing set.
- Classifier is learned based on the training set and get evaluated with testing set.
- Proportion of training and testing sets is at the discretion of analyst; typically **1:1 or 2:1**, and there is a trade-off between these sizes of these two sets.
- If the training set is too large, then model may be good enough, but estimation may be less reliable due to small testing set and vice-versa.

Random Subsampling

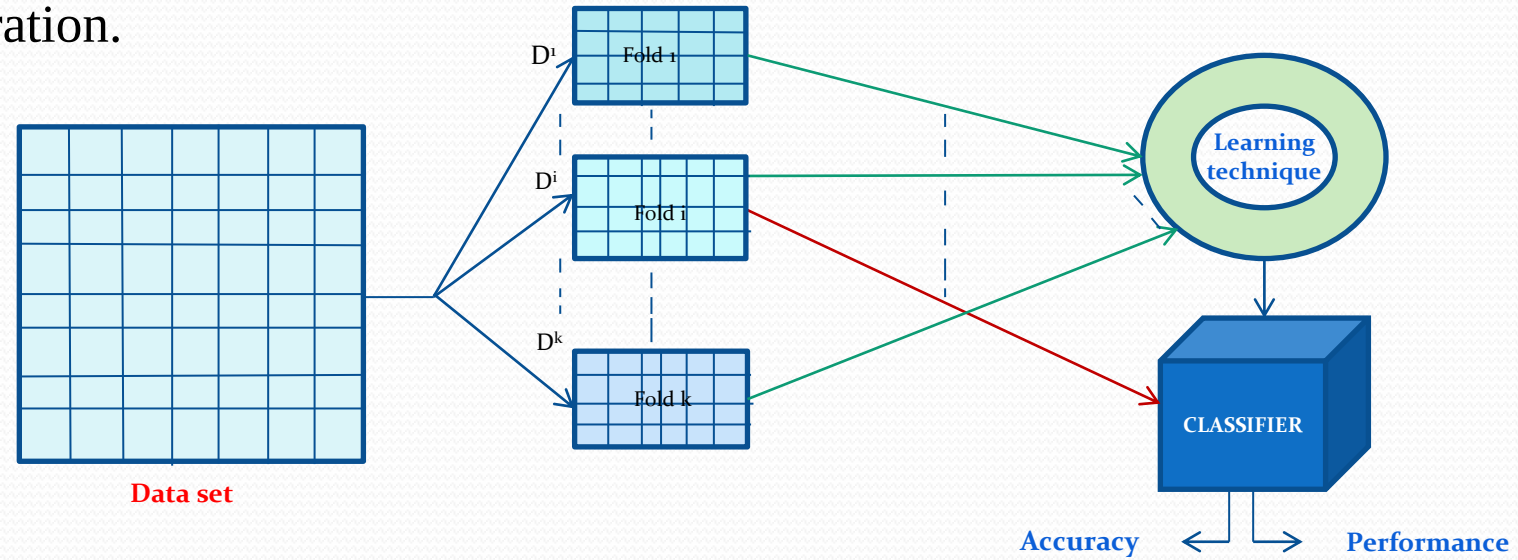
- It is a variation of Holdout method to overcome the drawback of over-presenting a class in one set thus under-presenting it in the other set and vice-versa.
- In this method, Holdout method is repeated ***k*** times, and in each time, two **disjoint sets are chosen at random** with a predefined sizes.
- Overall estimation is taken as the average of estimations obtained from each iteration.

Cross-Validation

- The main drawback of Random subsampling is, it does not have control over the number of times each tuple is used for training and testing.
- Cross-validation is proposed to overcome this problem.
- There are two variations in the cross-validation method.
  - k-fold cross-validation
  - *N*-fold cross-validation

k-fold Cross-Validation

- Dataset consisting of N tuples is divided into k (usually, 5 or 10) equal, mutually exclusive parts or folds $(D_1, D_2, \ldots, D_k)$, and if N is not divisible by k, then the last part will have fewer tuples than other (k-1) parts.
- A series of k runs is carried out with this decomposition, and in i^{th} iteration D_i is used as test data and other folds as training data
  - Thus, each tuple is used same number of times for training and once for testing.
- Overall estimate is taken as the average of estimates obtained from each iteration.



N-fold Cross-Validation

- In *k*-fold cross-validation method, $\frac{k-1}{N}$ part of the given data is used in training with *k*-tests.
- *N*-fold cross-validation is an extreme case of *k*-fold cross validation, often known as "Leave-one-out" cross-validation.
- Here, dataset is divided into as many folds as there are instances; thus, all most each tuple forming a training set, building *N* classifiers.
- In this method, therefore, *N* classifiers are built from *N-1* instances, and each tuple is used to classify a single test instances.
- Test sets are mutually exclusive and effectively cover the entire set (in sequence). This is as if trained by entire data as well as tested by entire data set.
- Overall estimation is then averaged out of the results of *N* classifiers.

N-fold Cross-Validation : Issue

- So far the estimation of accuracy and performance of a classifier model is concerned, the *N*-fold cross-validation is comparable to the others we have just discussed.
- The drawback of *N*-fold cross validation strategy is that it is computationally expensive, as here we have to repeat the run *N* times; this is particularly true when data set is large.
- In practice, the method is extremely beneficial with very small data set only, where as much data as possible to need to be used to train a classifier.

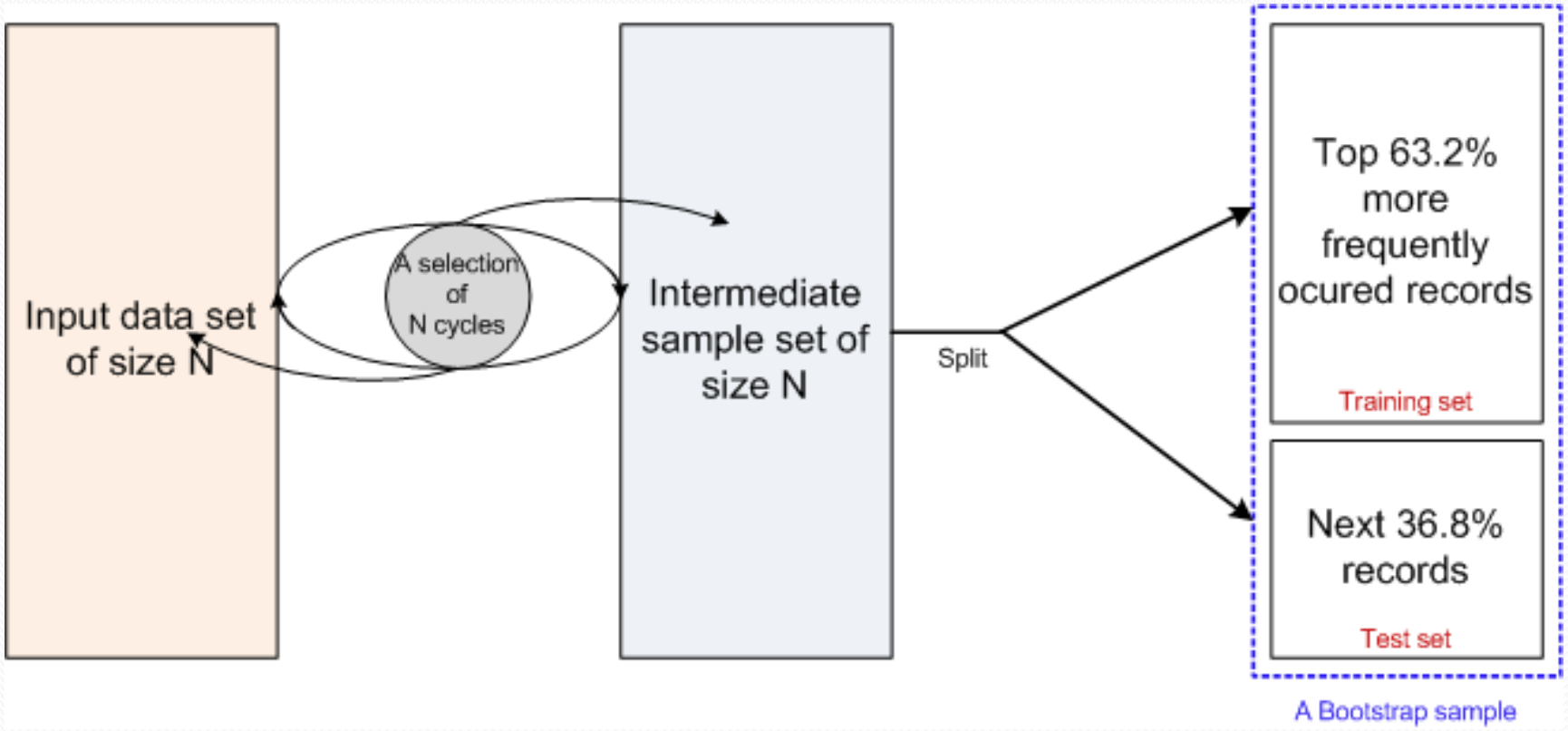
Bootstrap Method

- The Bootstrap method is a variation of repeated version of Random sampling method.
- The method suggests the sampling of training records with replacement.
  - Each time a record is selected for training set, is put back into the original pool of records, so that it is equally likely to be redrawn in the next run.
  - In other words, the Bootstrap method samples the given data set uniformly with replacement.
- The rational of having this strategy is that let some records be occur more than once in the samples of both training as well as testing.
  - What is the probability that a record will be selected more than once?

Bootstrap Method

- Suppose, we have given a data set of *N* records. The data set is sampled *N* times with replacement, resulting in a bootstrap sample (i.e., training set) of I samples.
  - Note that the entire runs are called a bootstrap sample in this method.
- There are certain chance (i.e., probability) that a particular tuple occurs one or more times in the training set
  - If they do not appear in the training set, then they will end up in the test set.
  - Each tuple has a probability of being selected $\frac{1}{N}$ (and the probability of not being selected is $\left(1-\frac{1}{N}\right)$.
  - We have to select N times, so the probability that a record will not be chosen during the whole run is $\left(1-\frac{1}{N}\right)^N$
  - Thus, the probability that a record is chosen by a bootstrap sample is $1-\left(1-\frac{1}{N}\right)^N$
  - For a large value of *N*, it can be proved that $\left(1-\frac{1}{N}\right)^N \approx e^{-1}$
  - Thus, the probability that a record chosen in a bootstrap sample is $1-e^{-1} = 0.632$

Bootstrap Method : Implication



- This is why, the Bootstrap method is also known as 0.632 bootstrap method

Accuracy Estimation

Accuracy Estimation

- We have learned how a classifier system can be tested. Next, we are to learn the metrics with which a classifier should be estimated.
- There are mainly to things to be measured for a given classifier
  - Accuracy
  - Performance
- **Accuracy estimation**
  - If *N* is the number of instances with which a classifier is tested and *p* is the number of correctly classified instances, the accuracy can be denoted as

$$\epsilon = \frac{p}{N}$$

  - Also, we can say the error rate (i.e., is misclassification rate) denoted by $\bar{\epsilon}$ is denoted by

$$\bar{\epsilon} = 1 - \epsilon$$

Accuracy : True and Predictive

- Now, this accuracy may be true (or absolute) accuracy or predicted (or optimistic) accuracy.
- True accuracy of a classifier is the accuracy when the classifier is tested with all possible unseen instances in the given classification space.
  - However, the number of possible unseen instances is potentially very large (if it is not infinite)
  - For example, classifying a hand-written character
  - Hence, measuring the true accuracy beyond the dispute is impractical.
- Predictive accuracy of a classifier is an accuracy estimation for a given test data (which are mutually exclusive with training data).
  - If the predictive accuracy for test set is $\in$ and if we test the classifier with a different test set it is very likely that a different accuracy would be obtained.
  - The predictive accuracy when estimated with a given test set it should be acceptable without any objection

Predictive Accuracy

- With the above-mentioned issue in mind, researchers have proposed two heuristic measures
  - Error estimation using **Loss Functions**
  - Statistical Estimation using **Confidence Level**
- In the next few slides, we will discus about the Loss estimation

Error Estimation using Loss Functions

- Let T be a matrix comprising with N test tuples

$$\begin{bmatrix} X_1 & y_1 \\ X_2 & y_2 \\ & \\ X_N & y_N \end{bmatrix}_{N\times(n+1)}$$

where X_i (i = 1, 2, …, N) is the n-dimensional test tuples with associated outcome y_i.

- Suppose, corresponding to (X_i, y_i), classifier produces the result (X_i, y_i')
- Also, assume that $(y_i - y_i')$ denotes a difference between y_i and y_i' (following certain difference (or similarity), (e.g., $(y_i - y_i') = 0$, if there is a match else 1)
- The two loss functions measure the error between y_i (the actual value) and y_i' (the predicted value) are

Absolute error:	$\lvert y_i - y_i' \rvert$
Squared error:	$\lvert y_i - y_i' \rvert^2$

Error Estimation using Loss Functions

- Based on the two loss functions, the test error (rate) also called generalization error, is defined as the average loss over the test set T. The following two measures for test errors are

Mean Absolute Error (MAE): $\frac{\sum_{i=1}^{N} |y_i - y_i'|}{N}$

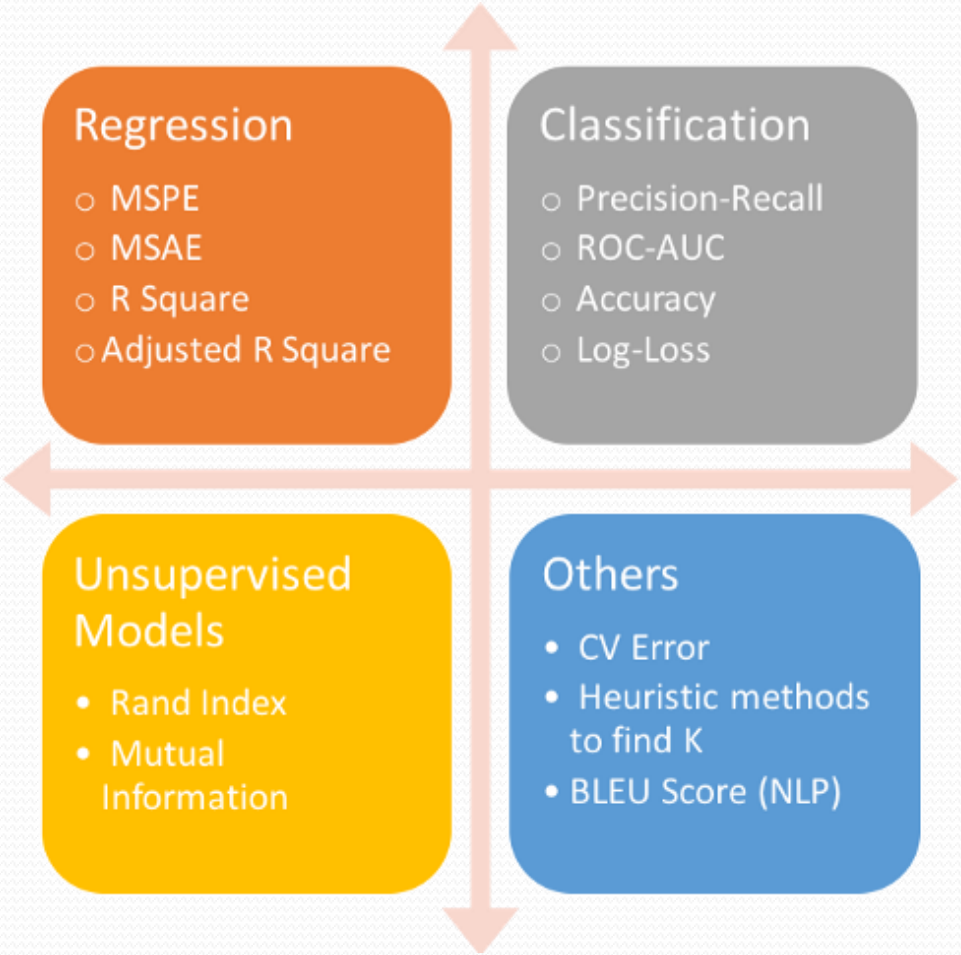
Mean Squared Error(MSE): $\frac{\sum_{i=1}^{N} (y_i - y_i')^2}{N}$

- Note that, MSE aggregates the presence of outlier.
- In addition to the above, a relative error measurement is also known. In this measure, the error is measured relative to the mean value $\tilde{y}$ calculated as the mean of y_i (i = 1, 2, …, N) of the training data say D. Two measures are

Relative Absolute Error (RAE): $\frac{\sum_{i=1}^{N} |y_i - y_i'|}{\sum_{i=1}^{N} |y_i - \tilde{y}|}$

Relative Squared Error (RSE): $\frac{\sum_{i=1}^{N} (y_i - y_i')^2}{\sum_{i=1}^{N} (y_i - \tilde{y})^2}$

Performance Estimation



Performance Estimation of a Classifier

- Predictive accuracy works fine, when the classes are balanced
  - That is, every class in the data set are equally important
- In fact, data sets with imbalanced class distributions are quite common in many real life applications
- When the classifier classified a test data set with imbalanced class distributions then, predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness.

Counter Example : Effectiveness of Predictive Accuracy

- Given a data set of stock markets, we are to classify them as "good" and "worst". Suppose, in the data set, out of 100 entries, 98 belong to "good" class and only 2 are in "worst" class.
  - With this data set, if classifier's predictive accuracy is 0.98, a very high value!
    - Here, there is a high chance that 2 "worst" stock markets may incorrectly be classified as "good"
  - On the other hand, if the predictive accuracy is 0.02, then none of the stock markets may be classified as "good"

Performance Estimation of a Classifier

- Thus, when the classifier classified a test data set with imbalanced class distributions, then predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness.
- This necessitates an alternative metrics to judge the classifier.
- Before exploring them, we introduce the concept of Confusion matrix.

Confusion Matrix

- A confusion matrix for a two classes (+, -) is shown below.

	C1	C2
C1	True positive	False negative
C2	False positive	True negative

	+	-
+	++	+-
-	-+	--

- There are four quadrants in the confusion matrix, which are symbolized as below.
  - True Positive (TP: f_{++}) : The number of instances that were positive (+) and correctly classified as positive (+v).
  - False Negative (FN: f_{+-}): The number of instances that were positive (+) and incorrectly classified as negative (-). It is also known as **Type 2 Error**.
  - False Positive (FP: f_{-+}): The number of instances that were negative (-) and incorrectly classified as (+). This also known as **Type 1 Error**.
  - True Negative (TN: f_{--}): The number of instances that were negative (-) and correctly classified as (-).

Confusion Matrix

Note:

- N_p = TP (f_{++}) + FN (f_{+-})
  = is the total number of positive instances.
- N_n = FP(f_{-+}) + Tn(f_{--})
  = is the total number of negative instances.
- $N = N_p + N_n$
  = is the total number of instances.
- (TP + TN) denotes the number of correct classification
- (FP + FN) denotes the number of errors in classification.
- For a perfect classifier FP = FN = 0, that is, there would be no Type 1 or Type 2 errors.

Confusion Matrix : Example

Example : Confusion matrix

A classifier is built on a dataset regarding Good and Worst classes of stock markets. The model is then tested with a test set of 10000 unseen instances. The result is shown in the form of a confusion matrix. The result is self explanatory.

Class	Good	Worst	Total	Rate(%)
Good	6954	46	7000	99.34
Worst	412	2588	3000	86.27
Total	**7366**	**2634**	**10000**	**95.52**

Predictive accuracy?

Confusion Matrix for Multiclass Classifier

- Having m classes, confusion matrix is a table of size $m \times m$, where, element at (i, j) indicates the number of instances of class i but classified as class j.
- To have good accuracy for a classifier, ideally most diagonal entries should have large values with the rest of entries being close to zero.
- Confusion matrix may have additional rows or columns to provide total or recognition rates per class.

Example : Confusion matrix with multiple class

Following table shows the confusion matrix of a classification problem with six classes labeled as C_1, C_2, C_3, C_4, C_5 and C_6. Calculate the predictive accuracy?

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	52	10	7	0	0	1
C_2	15	50	6	2	1	2
C_3	5	6	6	0	0	0
C_4	0	2	0	10	0	1
C_5	0	1	0	0	7	1
C_6	1	3	0	1	0	24

Performance Evaluation Metrics

- We now define a number of metrics for the measurement of a classifier.
  - In our discussion, we shall make the assumptions that there are only two classes: + (positive) and – (negative)
  - Nevertheless, the metrics can easily be extended to multi-class classifiers (with some modifications)
- **True Positive Rate (TPR):** It is defined as the fraction of the positive examples predicted correctly by the classifier.

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{f_{++}}{f_{++}+f_{+-}}$$

  - This metrics is also known as ***Recall***, ***Sensitivity*** or ***Hit rate***.
- **False Positive Rate (FPR):** It is defined as the fraction of negative examples classified as positive class by the classifier.

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} = \frac{f_{-+}}{f_{-+}+f_{--}}$$

  - This metric is also known as ***False Alarm Rate***.

Performance Evaluation Metrics

- **False Negative Rate (FNR):** It is defined as the fraction of positive examples classified as a negative class by the classifier.

$$FNR = \frac{FN}{P} = \frac{FN}{TP + FN} = \frac{f_{+-}}{f_{++} + f_{+-}}$$

- **True Negative Rate (TNR):** It is defined as the fraction of negative examples classified correctly by the classifier

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = \frac{f_{--}}{f_{--} + f_{-+}}$$

  - This metric is also known as ***Specificity***.

Performance Evaluation Metrics

- **Positive Predictive Value (PPV):** It is defined as the fraction of the positive examples classified as positive that are really positive

$$PPV = \frac{TP}{TP + FP} = \frac{f_{++}}{f_{++} + f_{-+}}$$

  - It is also known as *Precision*.

- **F_1 Score (F_1):** Recall (r) and Precision (p) are two widely used metrics employed in analysis, where detection of one of the classes is considered more significant than the others.

  - It is defined in terms of (r or TPR) and (p or PPV) as follows.

$$F_1 = \frac{2r.p}{r + p} = \frac{2TP}{2TP + FP + FN}$$

$$= \frac{2f_{++}}{2f_{++} + f_{+-} + f_{-+}} = \frac{2}{\frac{1}{r} + \frac{1}{p}}$$

Note

- F_1 represents the harmonic mean between recall and precision
- High value of F_1 score ensures that both Precision and Recall are reasonably high.

Performance Evaluation Metrics

- More generally, F_β score can be used to determine the trade-off between Recall and Precision as

$$F_\beta = \frac{(\beta + 1)rp}{r + \beta p} = \frac{(\beta + 1)TP}{(\beta + 1)TP + \beta FN + FP}$$

- Both, Precision and Recall are special cases of F_β when $\beta = 0$ and $\beta = 1$, respectively.

$$F_\beta = \frac{TP}{TP + FP} = Precision$$

$$F_\alpha = \frac{TP}{TP + FN} = Recall$$

Performance Evaluation Metrics

- A more general metric that captures Recall, Precision as well as is defined in the following.

$$F_\omega = \frac{\omega_1 TP + \omega_4 TN}{\omega_1 TP + \omega_2 FP + \omega_3 FN + \omega_4 TN}$$

Metric	ω_1	ω_2	ω_3	ω_4
Recall	1	1	0	1
Precision	1	0	1	0
F_β	β+1	β	1	0

Note

- In fact, given *TPR*, *FPR*, *p* and *r*, we can derive all others measures.
- That is, these are the universal metrics.

Predictive Accuracy & Error Rate

- It is defined as the fraction of the number of examples that are correctly classified by the classifier to the total number of instances.

$$\varepsilon = \frac{TP + TN}{P + N}$$
$$= \frac{TP + TN}{TP + FP + FN + TN}$$
$$= \frac{f_{++} + f_{--}}{f_{++} + f_{+-} + f_{-+} + f_{--}}$$

- This accuracy is equivalent to F_w with $w_1 = w_2 = w_3 = w_4 = 1$.

- The error rate $\bar{\varepsilon}$ is defined as the fraction of the examples that are incorrectly classified.

$$\bar{\varepsilon} = \frac{FP + FN}{P + N}$$
$$= \frac{FP + FN}{TP + TN + FP + FN}$$
$$= \frac{f_{+-} + f_{-+}}{f_{++} + f_{+-} + f_{-+} + f_{--}}$$

Note

$\bar{\bar{\varepsilon}} = 1 - \varepsilon$.

Accuracy, Sensitivity and Specificity

- Predictive accuracy (ε) can be expressed in terms of sensitivity and specificity.
- We can write

$$\varepsilon = \frac{TP + TN}{TP + FP + FN + TN}$$

$$= \frac{TP + TN}{P + N}$$

$$\varepsilon = \frac{TP}{P} \times \frac{P}{P + N} + \frac{TN}{N} \times \frac{N}{P + N}$$

Thus,

$$\varepsilon = \text{Sensitivity} \times \frac{P}{P+N} + \text{Specificity} \times \frac{N}{P+N}$$

Analysis with Performance Measurement Metrics

- Based on the various performance metrics, we can characterize a classifier.
- We do it in terms of TPR, FPR, Precision and Recall and Accuracy
- **Case 1: Perfect Classifier**

When every instance is correctly classified, it is called the perfect classifier. In this case, $TP = P$, $TN = N$ and CM is

$$TPR = \frac{P}{P} = 1$$

$$FPR = \frac{0}{N} = 0$$

$$Precision = \frac{P}{P} = 1$$

$$F_1\ Score = \frac{2 \times 1}{1+1} = 1$$

$$Accuracy = \frac{P+N}{P+N} = 1$$

		Predicted Class	
		+	-
Actual class	+	P	0
	-	0	N

Analysis with Performance Measurement Metrics

- **Case 2: Worst Classifier**

When every instance is wrongly classified, it is called the worst classifier. In this case, $TP = 0$, $TN = 0$ and the CM is

$$TPR = \frac{0}{P} = 0$$

$$FPR = \frac{N}{N} = 1$$

$$Precision = \frac{0}{N} = 0$$

F_1 *Score* = Not applicable as *Recall* + *Precision* = 0

$$\text{Accuracy} = \frac{0}{P+N} = 0$$

		Predicted Class	
		+	-
Actual class	+	0	P
	-	N	0

Analysis with Performance Measurement Metrics

- **Case 3: Ultra-Liberal Classifier**

The classifier always predicts the + class correctly. Here, the False Negative (FN) and True Negative (TN) are zero. The CM is

		Predicted Class	
		+	-
Actual class	+	P	0
	-	N	0

$$TPR = \frac{P}{P} = 1$$

$$FPR = \frac{N}{N} = 1$$

$$Precision = \frac{P}{P+N}$$

$$F_1\ Score = \frac{2P}{2P+N}$$

$$\text{Accuracy} = \frac{P}{P+N} = 0$$

Analysis with Performance Measurement Metrics

- **Case 4: Ultra-Conservative Classifier**

This classifier always predicts the - class correctly. Here, the False Negative (FN) and True Negative (TN) are zero. The CM is

		Predicted Class	
		+	-
Actual class	+	0	p
	-	0	N

$$TPR = \frac{0}{P} = 0$$

$$FPR = \frac{0}{N} = 0$$

$Precision =$ Not applicable

(as $TP + FP = 0$)

$F_1\ Score =$ Not applicable

$$\text{Accuracy} = \frac{N}{P+N} = 0$$

Predictive Accuracy versus TPR and FPR

- One strength of characterizing a classifier by its *TPR* and *FPR* is that they do not depend on the relative size of *P* and *N*.
  - The same is also applicable for *FNR* and *TNR* and others measures from CM.
- In contrast, the *Predictive Accuracy*, *Precision*, *Error Rate*, F_1 *Score*, etc. are affected by the relative size of *P* and *N*.
- *FPR*, *TPR*, *FNR* and *TNR* are calculated from the different rows of the CM.
  - On the other hand Predictive Accuracy, etc. are derived from the values in both rows.
- This suggests that *FPR*, *TPR*, *FNR* and TNR are more effective than *Predictive Accuracy*, etc.

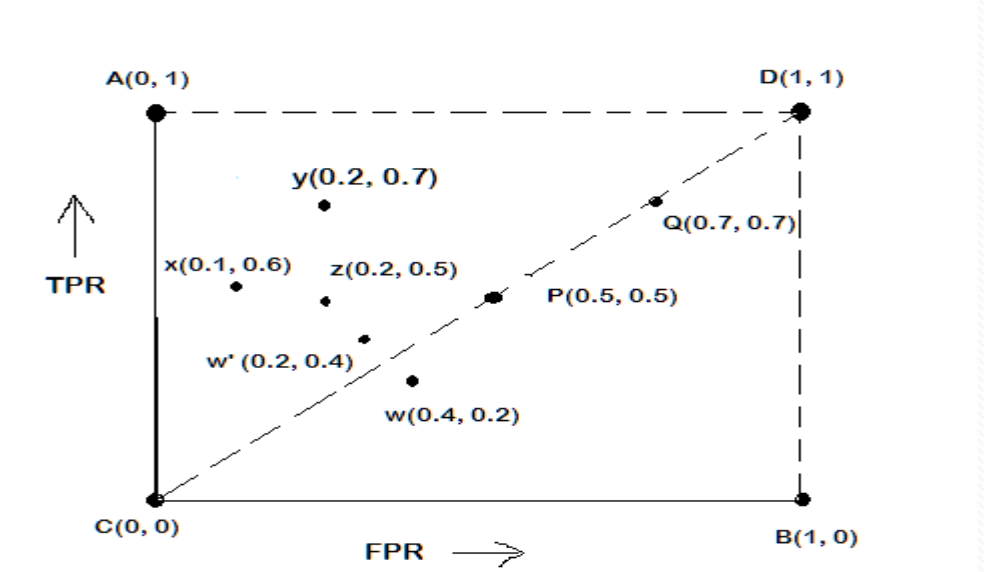
ROC Curves

ROC Curves

- ROC is an abbreviation of Receiver Operating Characteristic come from the signal detection theory, developed during World War 2 for analysis of radar images.
- In the context of classifier, ROC plot is a useful tool to study the behaviour of a classifier or comparing two or more classifiers.
- A ROC plot is a two-dimensional graph, where, X-axis represents FP rate (FPR) and Y-axis represents TP rate (TPR).
- Since, the values of FPR and TPR varies from 0 to 1 both inclusive, the two axes thus from 0 to 1 only.
- Each point *(x, y)* on the plot indicating that the FPR has value *x* and the TPR value *y*.

ROC Plot

- A typical look of ROC plot with few points in it is shown in the following figure.

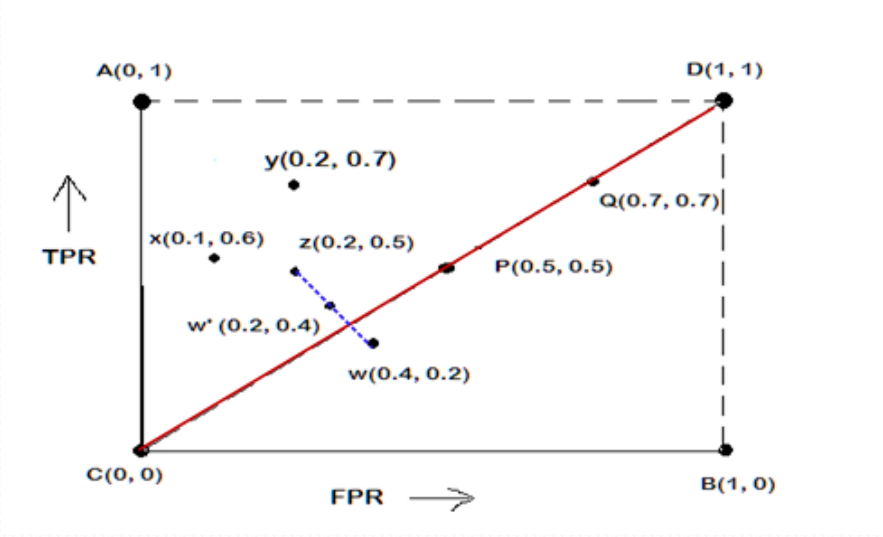


- Note the four cornered points are the four extreme cases of classifiers

Identify the four extreme classifiers.

Interpretation of Different Points in ROC Plot

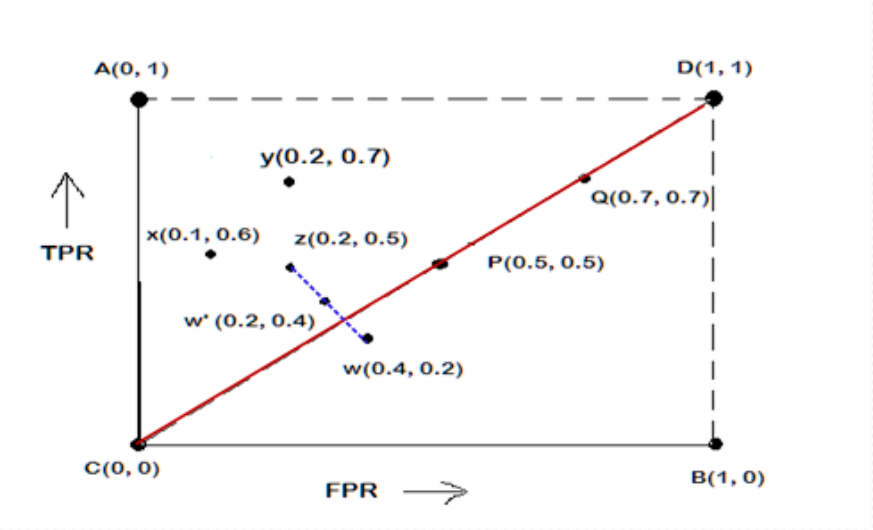
- Le us interpret the different points in the ROC plot.



- The four points (A, B, C, and D)
  - A: TPR = 1, FPR = 0, the ideal model, i.e., the perfect classifier, no false results
  - B: TPR = 0, FPR = 1, the worst classifier, not able to predict a single instance
  - C: TPR = 0, FPR = 0, the model predicts every instance to be a Negative class, i.e., it is an ultra-conservative classifier
  - D: TPR = 1, FPR = 1, the model predicts every instance to be a Positive class, i.e., it is an ultra-liberal classifier

Interpretation of Different Points in ROC Plot

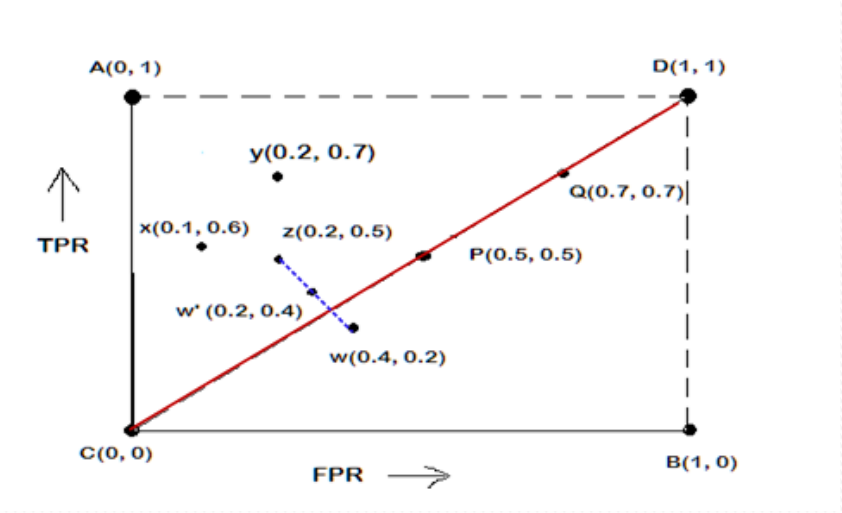
- Le us interpret the different points in the ROC plot.



- The points on diagonals
  - The diagonal line joining point C(0,0) and D(1,1) corresponds to random guessing
    - Random guessing means that a record is classified as positive (0r negative) with a certain probability
    - Suppose, a test set contacting N_+ positive and N_- negative instances. Suppose, the classifier guesses any instances with probability p
    - Thus, the random classifier is expected to correctly classify $p.N_+$ of the positive instances and $p.N_-$ of the negative instances
    - Hence, $TPR = FPR = p$
    - Since $TPR = FPR$, the random classifier results reside on the main diagonals

Interpretation of Different Points in ROC Plot

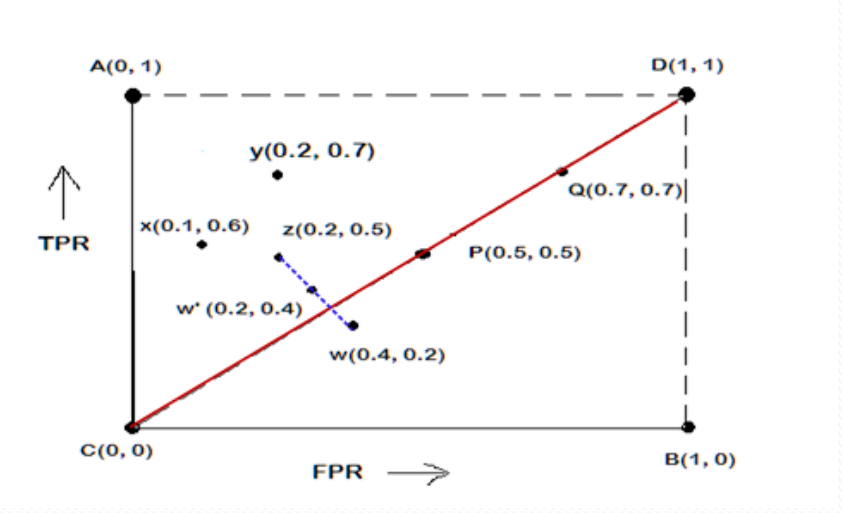
- Let us interpret the different points in the ROC plot.



- The points on the upper diagonal region
  - All points, which reside on upper-diagonal region are corresponding to classifiers "good" as their TPR is as good as FPR (i.e., FPRs are lower than TPRs)
  - Here, X is better than Z as X has higher TPR and lower FPR than Z.
  - If we compare X and Y, neither classifier is superior to the other

Interpretation of Different Points in ROC Plot

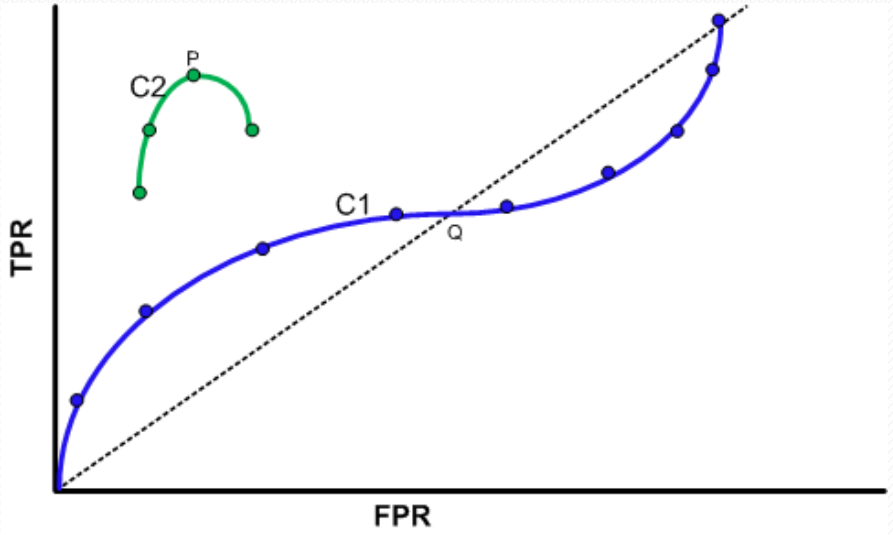
- Let us interpret the different points in the ROC plot.



- The points on the lower diagonal region
  - The Lower-diagonal triangle corresponds to the classifiers that are worst than random classifiers
  - Note: A classifier that is worst than random guessing, simply by reversing its prediction, we can get good results.
    - W'(0.2, 0.4) is the better version than W(0.4, 0.2), W' is a mirror reflection of W

Tuning a Classifier through ROC Plot

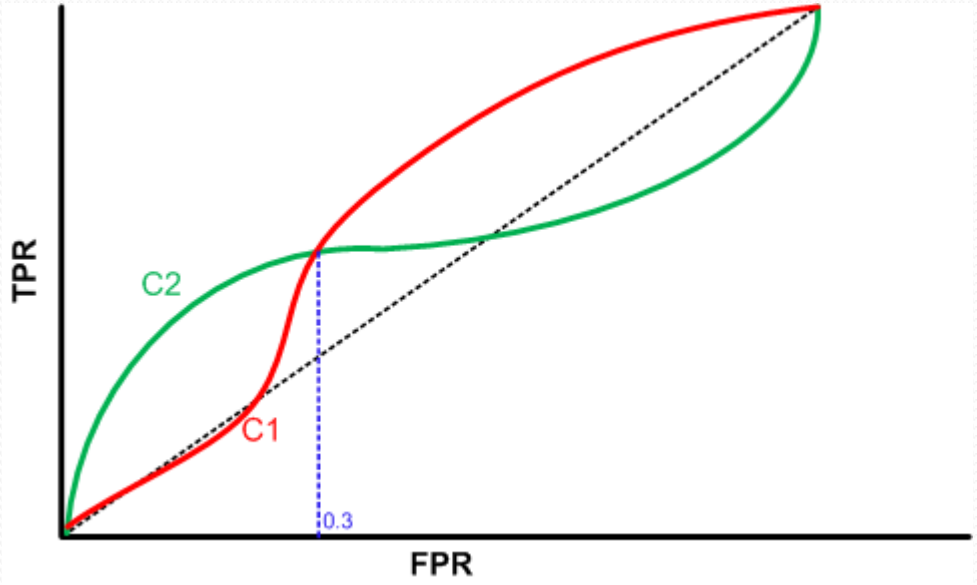
- Using ROC plot, we can compare two or more classifiers by their TPR and FPR values and this plot also depicts the trade-off between TPR and FPR of a classifier.



- Examining ROC curves can give insights into the best way of tuning parameters of classifier.
- For example, in the curve C2, the result is degraded after the point P. Similarly for the observation C1, beyond Q the settings are not acceptable.

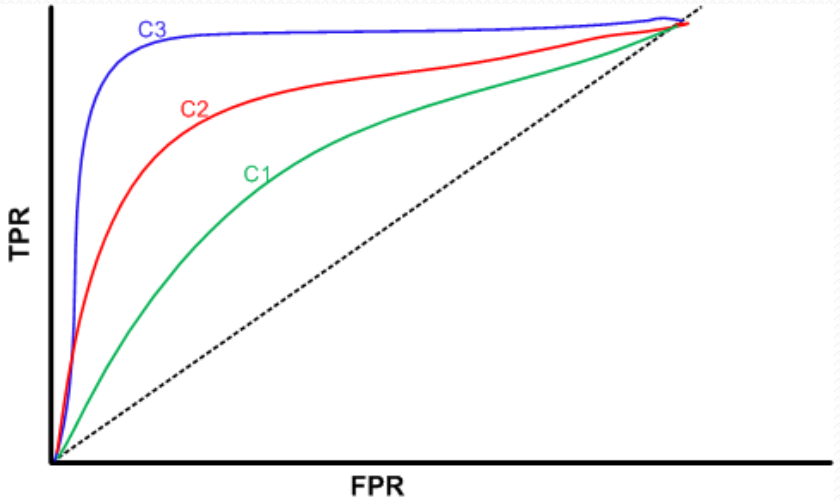
Comparing Classifiers trough ROC Plot

- Two curves C1 and C2 are corresponding to the experiments to choose two classifiers with their parameters.
- Here, C1 is better than C2 when FPR is less than 0.3.
- However, C2 is better, when FPR is greater than 0.3.
- Clearly, neither of these two classifiers dominates the other.



Comparing Classifiers trough ROC Plot

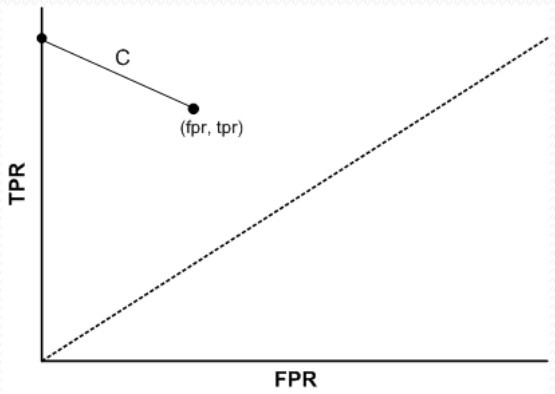
- We can use the concept of "area under curve" (AUC) as a better method to compare two or more classifiers.
- If a model is perfect, then its AUC = 1.
- If a model simply performs random guessing, then its AUC = 0.5
- A model that is strictly better than other, would have a larger value of AUC than the other.



- Here, C3 is best, and C2 is better than C1 as AUC(C3)>AUC(C2)>AUC(C1).

A Quantitative Measure of a Classifier

- The concept of ROC plot can be extended to compare quantitatively using Euclidean distance measure.
- See the following figure for an explanation.



- Here, *C(fpr, tpr)* is a classifier and $\boldsymbol{\delta}$ denotes the Euclidean distance between the best classifier (0, 1) and C. That is,
  - $\boldsymbol{\delta} = \sqrt{fpr^2 + (1 - tpr)^2}$

A Quantitative Measure of a Classifier

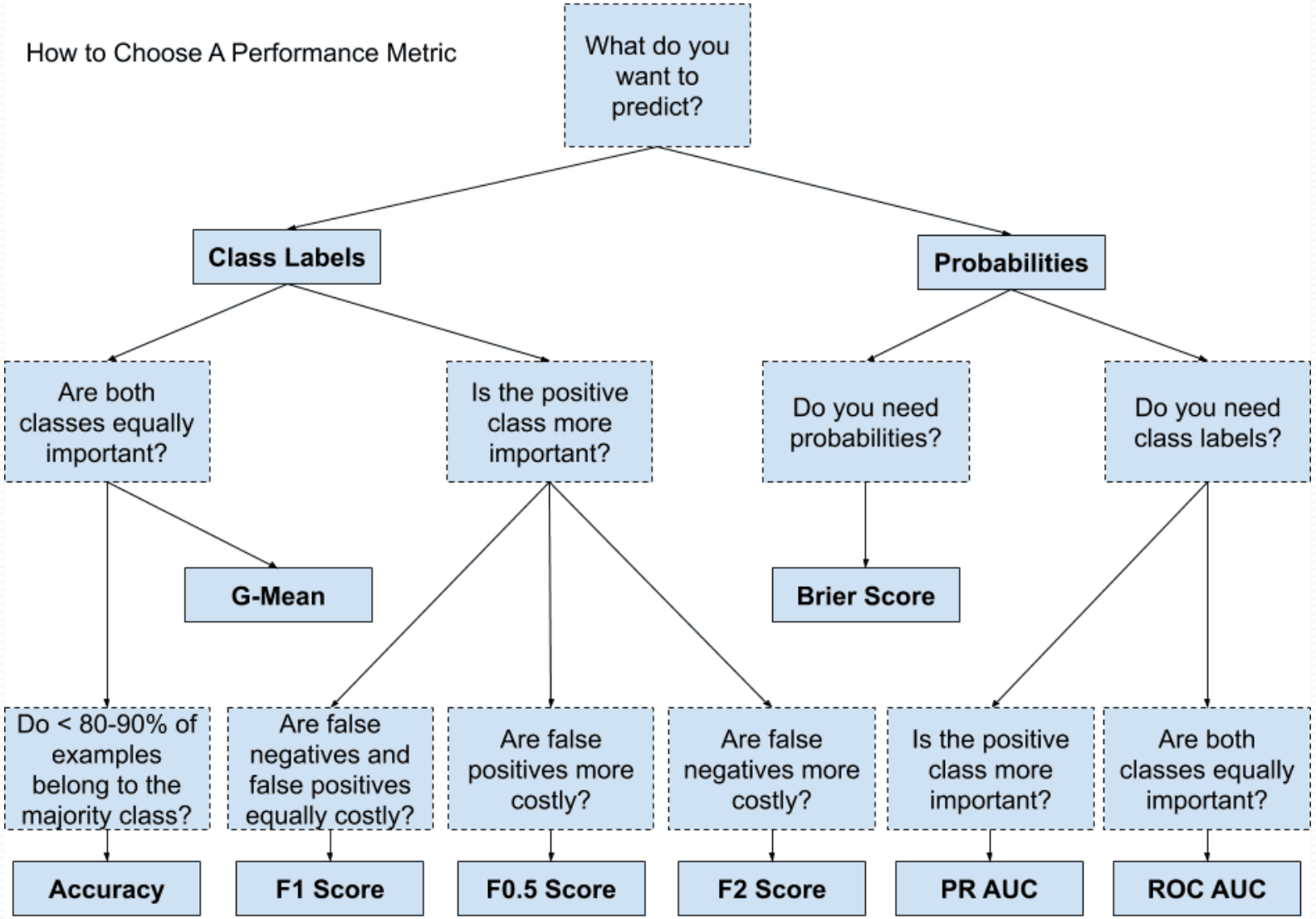
- The smallest possible value of $\boldsymbol{\delta}$ is 0
- The largest possible values of $\boldsymbol{\delta}$ is $\sqrt{2}$ (when (fpr = 1 and tpr = 0).
- We could hypothesise that the smaller the value of $\boldsymbol{\delta}$, the better the classifier.
- $\boldsymbol{\delta}$ is a useful measure, but does not take into account the relative importance of true and false positive rates.
- We can specify the relative importance of making TPR as close to 1 and FPR as close 0 by a weight *w* between 0 to 1.
- We can define weighted $\boldsymbol{\delta}$ (denoted by $\boldsymbol{\delta}_w$) as

$$\boldsymbol{\delta}_w = \sqrt{(1-w)fpr^2 + w(1-tpr)^2}$$

Note

- If $w = 0$, it reduces to $\boldsymbol{\delta}_w$= fpr, i.e., FP Rate.
- If $w = 1$, it reduces to $\boldsymbol{\delta}_w$ = 1 – tpr, i.e., we are only interested to maximizing TP Rate.

Cheat Sheet on Classification Performance Metrics



Reference

- The detail material related to this lecture can be found in

Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014

Any question?

ctanujit@gmail.com