Multivariate Source Use Case (Single DataFrameCase)

In this vignette I will represent a use case of the Source Multivariate Entropy Triangle with some individual Databases

Importing Libraries

As the functions for the entopies measures are stored in other domain, first we will need to access those modules with the functions and the import all the necessary functions

In [12]:

```
# Bring your packages onto the path
import sys,os
sys.path.append(os.path.abspath(os.path.join('..'))) #'entropytriangle main dire
ctory
```

In [13]:

```
from entropytriangle import * #importing all modules necessary for the plotting
```

Dowloading a set of Databases

In [14]:

```
#df = pd.read_csv('Arthritis.csv',delimiter=',',index_col='Unnamed: 0')
#df = pd.read_csv('Breast_data.csv',delimiter=',',index_col='Unnamed: 0').drop
(['Sample code number'],axis = 1).replace('?',np.nan) # in this DB the missing v
alues are represented as '?'
#df = pd.read_csv('Glass.csv',delimiter=',')
#df = pd.read_csv('Ionosphere.csv',delimiter=',')
df = pd.read_csv('Iris.csv',delimiter=',',index_col='Id')
#df = pd.read_csv('Wine.csv',delimiter=',').drop(['Wine'],axis = 1)
```

In [15]:

```
df.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150 entries, 1 to 150
Data columns (total 5 columns):
SepalLengthCm    150 non-null float64
SepalWidthCm     150 non-null float64
PetalLengthCm    150 non-null float64
PetalWidthCm     150 non-null float64
Species          150 non-null object
dtypes: float64(4), object(1)
memory usage: 7.0+ KB
```

In [16]:

```
df.head(10)
```

Out[16]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
Id					
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa

Discretizing the Data before entropy calculation

We have defined a function for discretizing a hole dataset, the function divides de entries in "NROWS(DF)^(1/3)" equally sized spaces, and turn the data types in "categories"

In [17]:

```
df = discretization(df)
```

```
/Users/jaime.de.los.rios/Documents/GitHub/entropytriangle/entropytri
angle/auxfunc.py:35: UserWarning: Discretizing data!
  warning("Discretizing data!")
```

In [18]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150 entries, 1 to 150
Data columns (total 5 columns):
SepalLengthCm    150 non-null category
SepalWidthCm     150 non-null category
PetalLengthCm    150 non-null category
PetalWidthCm     150 non-null category
Species          150 non-null category
dtypes: category(5)
memory usage: 2.8 KB
```

In [19]:

```
df.head(10)
```

Out[19]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
Id					
1	1	3	0	0	0
2	0	2	0	0	0
3	0	2	0	0	0
4	0	2	0	0	0
5	0	3	0	0	0
6	1	3	0	0	0
7	0	2	0	0	0
8	0	2	0	0	0
9	0	1	0	0	0
10	0	2	0	0	0

Source Entropies Measures calculation

Once we have our data discretized, we will start by calculating the values of the entropies for the posterior plots

In [20]:

```
'''
As the database is previosly discretized we won´t need the values of the bins
'Type variable select the entropy calculation:'
    Total: Total source entropy decomposition (CPx)
    Dual : Dual source entropy decomposition (DPx instead of CPx)
'''

edf = sentropies(df , type = 'total' , base = 2)
```

In [21]:

```
edf
```

Out[21]:

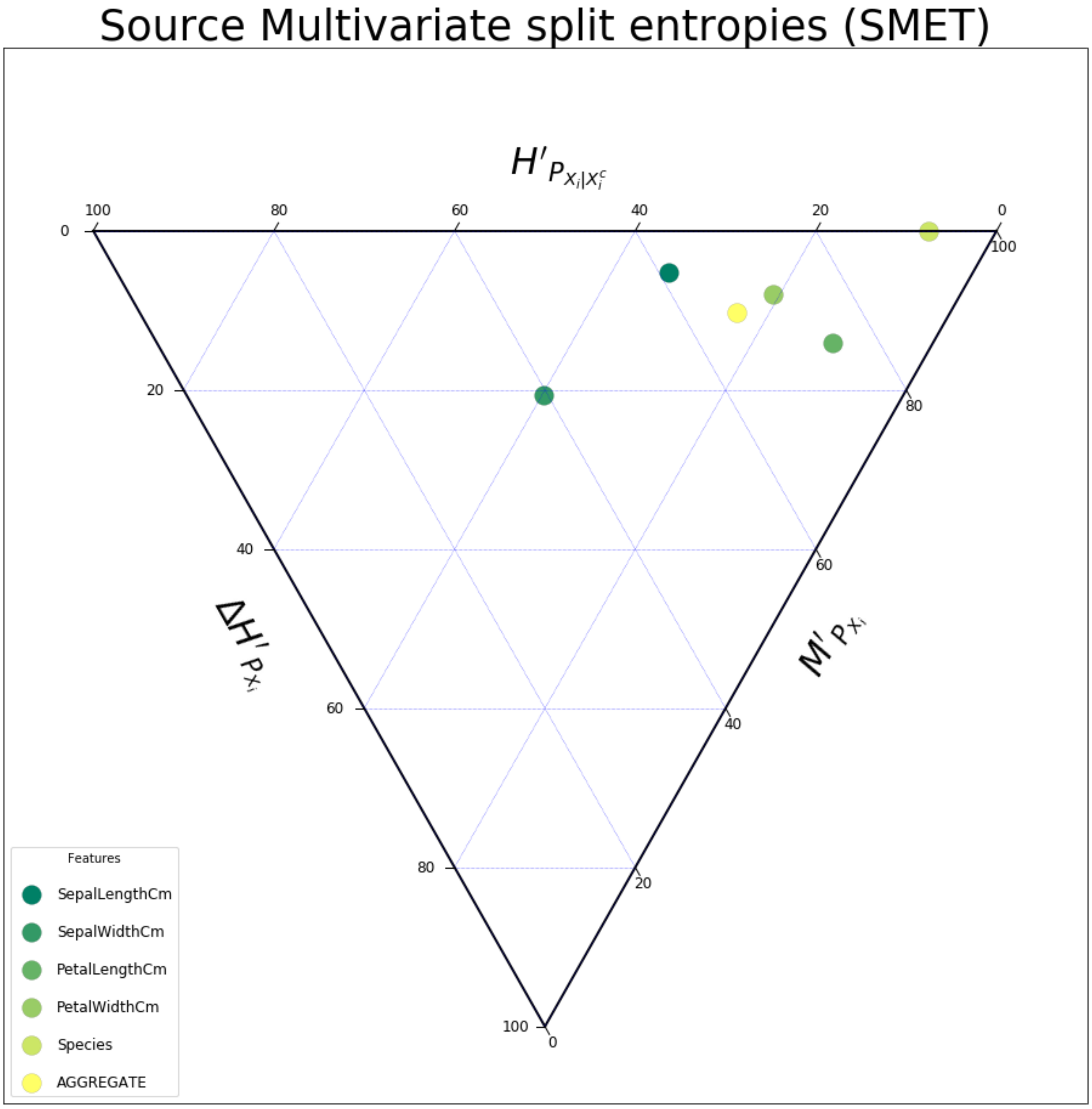
Name	H_Uxi	H_Pxi	DeltaH_Pxi	M_Pxi	VI_Pxi
SepalLengthCm	2.321928	2.200620	0.121308	1.417675	0.782945
SepalWidthCm	2.321928	1.841723	0.480205	0.917768	0.923955
PetalLengthCm	2.321928	1.995571	0.326357	1.738118	0.257453
PetalWidthCm	2.321928	2.137460	0.184468	1.654826	0.482635
Species	1.584963	1.584963	0.000000	1.465241	0.119721
AGGREGATE	10.872675	9.760337	1.112338	7.193628	2.566709

Source Entropies Entropy Triangle Plotting

The last step will be plotting the values calculated previously. The coordinates will be calculated multiplying the normalized values needed by the scale used for plotting the triangle, and will appear behind the triangle plot for comparission

In [28]:

```
entriangle(edf,s_mk=250,scale= 100, pltscale=16 , ticks_size=12, gridl = 20)
```


Source Multivariate split entropies (SMET)
$H'_{P_{X_i|X_i^c}}$
$\Delta H'_{P_{X_i}}$
$M'_{P_{X_i}}$
100
80
60
40
20
0
Features
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
Species
AGGREGATE