Chapter 1

Practical Application

In chapter **??** we introduce our own methodology for evaluating the similarity degree of fuzzy predicates. The method lacked a feature for realizing an error tolerance when working on knowledge bases with incomplete information. For that reason, three approaches are proposed in chapter **??**. The most elegant of these approaches, namely the *Hybrid Approach* inherits the best features of the preceding ones, while avoiding the shortcomings via merging the two by attaining them some weights. Previously we displayed how these weights could be decided in an intuitionistic way, and hinted the possibility of an atomized procedure. In this chapter we show our proposal for atomizing the process with a real-world example.

1.1 The Problem

In section **??** we mention about how today's search engines have limited querying capabilities. We give the example of the query *red car*, and say that an ideal frame work would return us the results of the query *orange car* in addition to the original one, with lower credibility values. In the light of this, as our real-world practical example, we inspect the similarity relations between the colors. We may observe colors as a domain of interest in many distinct fields, however one particular interesting example is the set of canonical colors that the web browsers use, namely the *X11 colors* [**?**]. The system makes use of the *RGB* framework, where colors are specified as triplets. Every color is depicted by three values which represent the *Red*, *Green* and *Blue* amount that the color consists of. The complete *X11* table is shown in the following page.

HTML name	Hex code			Decimal code		
	R	G	B	R	G	B
Red colors						
IndianRed	CD	5C	5C	205	92	92
LightCoral	F0	80	80	240	128	128
Salmon	FA	80	72	250	128	114
DarkSalmon	E9	96	7A	233	150	122
LightSalmon	FF	A0	7A	255	160	122
Red	FF	00	00	255	0	0
Crimson	DC	14	3C	220	20	60
FireBrick	B2	22	22	178	34	34
DarkRed	8B	00	00	139	0	0
Pink colors						
Pink	FF	C0	CB	255	192	203
LightPink	FF	B6	C1	255	182	193
HotPink	FF	69	B4	255	105	180
DeepPink	FF	14	93	255	20	147
MediumVioletRed	C7	15	85	199	21	133
PaleVioletRed	DB	70	93	219	112	147
Orange colors						
LightSalmon	FF	A0	7A	255	160	122
Coral	FF	7F	50	255	127	80
Tomato	FF	63	47	255	99	71
OrangeRed	FF	45	00	255	69	0
DarkOrange	FF	8C	00	255	140	0
Orange	FF	A5	00	255	165	0
Yellow colors						
Gold	FF	D7	00	255	215	0
Yellow	FF	FF	00	255	255	0
LightYellow	FF	FF	E0	255	255	224
LemonChiffon	FF	FA	CD	255	250	205
LightGoldenrodYellow	FA	FA	D2	250	250	210
PapayaWhip	FF	EF	D5	255	239	213
Moccasin	FF	E4	B5	255	228	181
PeachPuff	FF	DA	B9	255	218	185
PaleGoldenrod	EE	E8	AA	238	232	170
Khaki	F0	E6	8C	240	230	140
DarkKhaki	BD	B7	6B	189	183	107
Purple colors						
Lavender	E6	E6	FA	230	230	250
Thistle	D8	BF	D8	216	191	216
Plum	DD	A0	DD	221	160	221
Violet	EE	82	EE	238	130	238
Orchid	DA	70	D6	218	112	214
Fuchsia	FF	00	FF	255	0	255
Magenta	FF	00	FF	255	0	255
MediumOrchid	BA	55	D3	186	85	211
MediumPurple	93	70	DB	147	112	219
BlueViolet	8A	2B	E2	138	43	226
DarkViolet	94	00	D3	148	0	211
DarkOrchid	99	32	CC	153	50	204
DarkMagenta	8B	00	8B	139	0	139
Purple	80	00	80	128	0	128
Indigo	4B	00	82	75	0	130
DarkSlateBlue	48	3D	8B	72	61	139
SlateBlue	6A	5A	CD	106	90	205
MediumSlateBlue	7B	68	EE	123	104	238

HTML name	Hex code			Decimal code		
	R	G	B	R	G	B
Green colors						
GreenYellow	AD	FF	2F	173	255	47
Chartreuse	7F	FF	00	127	255	0
LawnGreen	7C	FC	00	124	252	0
Lime	00	FF	00	0	255	0
LimeGreen	32	CD	32	50	205	50
PaleGreen	98	FB	98	152	251	152
LightGreen	90	EE	90	144	238	144
MediumSpringGreen	00	FA	9A	0	250	154
SpringGreen	00	FF	7F	0	255	127
MediumSeaGreen	3C	B3	71	60	179	113
SeaGreen	2E	8B	57	46	139	87
ForestGreen	22	8B	22	34	139	34
Green	00	80	00	0	128	0
DarkGreen	00	64	00	0	100	0
YellowGreen	9A	CD	32	154	205	50
OliveDrab	6B	8E	23	107	142	35
Olive	80	80	00	128	128	0
DarkOliveGreen	55	6B	2F	85	107	47
MediumAquamarine	66	CD	AA	102	205	170
DarkSeaGreen	8F	BC	8F	143	188	143
LightSeaGreen	20	B2	AA	32	178	170
DarkCyan	00	8B	8B	0	139	139
Teal	00	80	80	0	128	128
Blue/Cyan colors						
Aqua	00	FF	FF	0	255	255
Cyan	00	FF	FF	0	255	255
LightCyan	E0	FF	FF	224	255	255
PaleTurquoise	AF	EE	EE	175	238	238
Aquamarine	7F	FF	D4	127	255	212
Turquoise	40	E0	D0	64	224	208
MediumTurquoise	48	D1	CC	72	209	204
DarkTurquoise	00	CE	D1	0	206	209
CadetBlue	5F	9E	A0	95	158	160
SteelBlue	46	82	B4	70	130	180
LightSteelBlue	B0	C4	DE	176	196	222
PowderBlue	B0	E0	E6	176	224	230
LightBlue	AD	D8	E6	173	216	230
SkyBlue	87	CE	EB	135	206	235
LightSkyBlue	87	CE	FA	135	206	250
DeepSkyBlue	00	BF	FF	0	191	255
DodgerBlue	1E	90	FF	30	144	255
CornflowerBlue	64	95	ED	100	149	237
RoyalBlue	41	69	E1	65	105	225
Blue	00	00	FF	0	0	255
MediumBlue	00	00	CD	0	0	205
DarkBlue	00	00	8B	0	0	139
Navy	00	00	80	0	0	128
MidnightBlue	19	19	70	25	25	112

HTML name	Hex code			Decimal code		
	R	G	B	R	G	B
Brown colors						
Cornsilk	FF	F8	DC	255	248	220
BlanchedAlmond	FF	EB	CD	255	235	205
Bisque	FF	E4	C4	255	228	196
NavajoWhite	FF	DE	AD	255	222	173
Wheat	F5	DE	B3	245	222	179
BurlyWood	DE	B8	87	222	184	135
Tan	D2	B4	8C	210	180	140
RosyBrown	BC	8F	8F	188	143	143
SandyBrown	F4	A4	60	244	164	96
Goldenrod	DA	A5	20	218	165	32
DarkGoldenrod	B8	86	0B	184	134	11
Peru	CD	85	3F	205	133	63
Chocolate	D2	69	1E	210	105	30
SaddleBrown	8B	45	13	139	69	19
Sienna	A0	52	2D	160	82	45
Brown	A5	2A	2A	165	42	42
Maroon	80	00	00	128	0	0
White colors						
White	FF	FF	FF	255	255	255
Snow	FF	FA	FA	255	250	250
Honeydew	F0	FF	F0	240	255	240
MintCream	F5	FF	FA	245	255	250
Azure	F0	FF	FF	240	255	255
AliceBlue	F0	F8	FF	240	248	255
GhostWhite	F8	F8	FF	248	248	255
WhiteSmoke	F5	F5	F5	245	245	245
Seashell	FF	F5	EE	255	245	238
Beige	F5	F5	DC	245	245	220
OldLace	FD	F5	E6	253	245	230
FloralWhite	FF	FA	F0	255	250	240
Ivory	FF	FF	F0	255	255	240
AntiqueWhite	FA	EB	D7	250	235	215
Linen	FA	F0	E6	250	240	230
LavenderBlush	FF	F0	F5	255	240	245
MistyRose	FF	E4	E1	255	228	225
Gray colors						
Gainsboro	DC	DC	DC	220	220	220
LightGrey	D3	D3	D3	211	211	211
Silver	C0	C0	C0	192	192	192
DarkGray	A9	A9	A9	169	169	169
Gray	80	80	80	128	128	128
DimGray	69	69	69	105	105	105
LightSlateGray	77	88	99	119	136	153
SlateGray	70	80	90	112	128	144
DarkSlateGray	2F	4F	4F	47	79	79
Black	00	00	00	0	0	0

1.1.1 Preliminaries

Before starting to proceed with the problem itself, we ought to mention about some problem specific concepts.

1.1.1.1 Real World Similarity

As stated in section 1.1 in the *RGB* scheme, all colors are represented by triplets of *Red*, *Green* and *Blue* values. With this in mind, when observing the similarity value between two colors, directly from the real-word data, we may consider them as if they were placed in *3D space*. Thus each of the *RGB* values acts as a coordinate for the corresponding color, and we may utilize a geometric distance formulation for this problem. In this scenario we adopt the *Euclidean distance*, which is indeed one of the common definitions in the field of *color science*. [**?**]

In Euclidean three-space, the distance between points (x_1, y_1, z_1) and (x_2, y_2, z_2) is

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Thus in our problem, the similarity distance between two colors C and S follows as:

$$similarity\_distance = \sqrt{(C_{Red} - S_{Red})^2 + (C_{Green} - S_{Green})^2 + (C_{Blue} - S_{Blue})^2}$$

We then follow by normalizing this metric distance value with the following algorithm

$$degree\_of\_similarity = 1 - \frac{similarity\_distance}{\sqrt{3 * 255^2}}$$

so that the resulting values fall between the *real number* interval of $[0, 1]$.

As one will notice the denominator of the function is the highest possible distance in the metric space. Real world correspondence of this scenario consists the color pair of *Black* and *White*. The algorithm would return 0 as a result for this case. Moreover as expected, the pairing of any color with itself would result the algorithm computing 1 as the similarity degree.

1.1.1.2 Evaluated Similarity

The gist of our methodology, which is introduced in chapter **??**, is conserved in the practical case with minor modifications.

The predicate trees are relatively simple in structural terms. All are depth one, have three children.

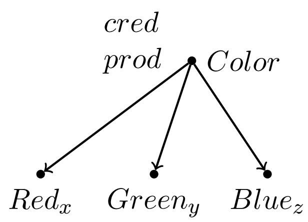

FIGURE 1.1: Example of a color predicate tree

As we know main focus of our algorithm is utilizing predefined similarity relations between the subconcepts. In this particular example, we assume the only predefined similarity relations are the ones between the same RGB main colors. So for every Red_α and Red_β a similarity relations is defined where α and β are two integer values from the interval $[0, 255]$.

The corresponding function is as follows:

$$similarity\_single\_color = 1 - \frac{|c_1 - c_2|}{255}$$

which informally equals to the normalization of the metric distance on a single coordinate of the colors, so that it falls into the interval of $[0, 1]$.

Lastly, the evaluation function follows the same procedure described in chapter **??**.

1.1.1.3 Pseudo Credibility Values

As discussed in chapter **??**, the main focus of interest is finding the weight constants in the *Hybrid Approach(HA)* in an automatic sense. As one might observe, the problem consists the necessary information for calculating the values of *VA* and *EA* thus for fixing the weights, solely the credibility value should also be known. So for training weights, there is a need for some initial *pseudo credibility values.*

In the end we want the product of our similarity and credibility estimation, close to the real degree of similarity.

$$degree\_of\_similarity \times degree\_of\_credibility \cong real\_similarity$$

Through the preceding sections, we know evaluating both of values concerning similarities. So by putting these values in the equation, we may compute the variable of interest.

1.1.1.4 Computing Weights

In section **??** we introduce the formula for the *Hybrid Approach* as follows:

$$\mathit{Credibility} = (w \times [\textbf{Vertex\_Approach}]) + ((1 - w) \times [\textbf{Edge\_Approach}]) \tag{1.1}$$

where $w \in [0..1]$.

As mentioned earlier, when one has the value of credibility, getting the value of the weights from the formula is trivial.

One noteworthy concept here is the *Global Weight*. The focus of this methodology is training the weights from some pre-given info, then fixing the weights through some procedure so the credibility values for new fuzzy rules can be determined.

For maintaining the Multi-Adjoint properties as shown in [**?**], we adopt the minimum operator for evaluating the global weight. Informally, we train a single weight for every similarity pair in our training set, then set the minimum of those values as our global weight.

1.1.1.5 The Credibility

When the weights are computed, the credibility values are easily evaluated via the function we introduce in section **??**. This consists the last step of the methodology, which could just be followed by checking both the credibility value and the similarity value together in comparison to the real degree of similarity.

Here the desired outcome is getting a conservative results, so to speak a value which does not exceed the original degree of similarity, but still finding one which is relatively close.

1.2 A Practical Example

Suppose we have the following six color pairings in our training set:

```
Salmon Coral
Tomato Peru
YellowGreen MediumAquamarine
FineBrick LightSeaGreen
Crimson RoyalBlue
Snow MidnightBlue
```

Listing 1.1: The training set

For the sake of an interesting display, let us select a pairing from the training set and one completely outside of the initial knowledge base, in order to construct our test set:

```
Tomato Peru
Pink Violet
```

Listing 1.2: The training set

With these information in hand, we are ready for pursuing with the calculations.

1.2.1 Training Process for Salmon and Coral

In this section we show step by step the process of training a weight via utilizing the info of a color pairing from the training set. The task is simply following the explicit process described in section 1.1.1.

The first step is finding the real-world similarity proximity between the colors, namely *Salmon* and *Coral*. And for that, one needs to find out the metric distance of similarity of the colors.

We input the corresponding RGB values of the colors into the modified euclidian function, defined in section 1.1.1.1.

$$similarity\_distance = \sqrt{(C_{Red} - S_{Red})^2 + (C_{Green} - S_{Green})^2 + (C_{Blue} - S_{Blue})^2}$$

$$\begin{aligned} similarity\_distance &= \sqrt{(C_{Red} - S_{Red})^2 + (C_{Green} - S_{Green})^2 + (C_{Blue} - S_{Blue})^2} \\ &= \sqrt{(C_{250} - S_{255})^2 + (C_{128} - S_{127})^2 + (C_{114} - S_{80})^2} \\ &\cong \mathbf{34.38} \end{aligned} \tag{1.2}$$

Then use this metric distance value in the similarity evaluation function:

$$\begin{aligned} degree\_of\_similarity &= 1 - \frac{similarity\_distance}{\sqrt{3 * 255^2}} \\ &= 1 - \frac{34.48}{\sqrt{3 * 255^2}} \\ &\cong \mathbf{92.21} \end{aligned} \tag{1.3}$$

Second step of the process is finding the degree of similarity between the colors via using our own evaluation algorithm. As we know first thing to do here is handling with the calculation of the similarity proximities of subconcepts. There will be three such relations that have to be taken into account. Let us compute one of them, namely the one concerning the *Red* value of the colors:

$$\begin{aligned} similarity\_single\_color &= 1 - \frac{|c_1 - c_2|}{255} \\ &= 1 - \frac{|250 - 255|}{255} \\ &\cong \mathbf{0.98} \end{aligned} \tag{1.4}$$

After doing the same computation for the subconcept similarity relations regarding *Green* and *Blue*, and putting them in the evaluation function, we get:

$$degree\_of\_similarity = 0.947$$

In section 1.1.1.3 we have stated that one we have the both results of the similarity evaluations, we may conclude the credibility value which will be used for training the weights. The corresponding formula and the computation is as follows:

$$\begin{aligned} degree\_of\_similarity \times degree\_of\_credibility &= real\_similarity \\ degree\_of\_credibility &= \frac{real\_similarity}{degree\_of\_similarity} \\ &= 1 - \frac{0.947}{0.98} \\ &\cong \mathbf{0.973} \end{aligned} \tag{1.5}$$

The last computation enables us to continue with the last step of training, which is evaluating the weight. In addition to the credibility value, the *VA* and *EA* values should also be calculated. Since all of the subconcepts are included in some similarity relation, the *VA* value is 1. Moreover as there are three predefined subconcept similarity relations out of nine possible relations(i.e. $3 \times 3 = 9$), the corresponding *EA* value is $1/3$.

With the help of all these information, the so-called *trained weight value* is:

$$
\begin{aligned}
\mathit{Credibility} &= (w \times [\textbf{Vertex\_Approach}]) + ((1-w) \times [\textbf{Edge\_Approach}]) \\
w &= \frac{\mathit{Credibility} - [\textbf{Edge\_Approach}]}{[\textbf{Vertex\_Approach}] - [\textbf{Edge\_Approach}]} \\
&= \frac{0.973 - 0.33}{1 - 0.33} \\
&\cong \mathbf{0.959}
\end{aligned}
\tag{1.6}
$$

And this concludes our calculation for the first color pair of the training set.

1.2.2 The Outcome

A complete implementation of the process in *C++* programming language is displayed at the appendix.

Regarding this example, the console output of the program is depicted below:

```
Please select your choice of program: a
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~TRAINING SET~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Color_pair[0], Salmon and Coral:
_______________________________________
sim1:   0.947712        sim2:   0.922159
cred2:  0.973037        weight: 0.959555


Color_pair[1], Tomato and Peru:
_______________________________________
sim1:   0.879739        sim2:   0.8615
cred2:  0.979268        weight: 0.968902


Color_pair[2], YellowGreen and MediumAquamarine:
_______________________________________
sim1:   0.775163        sim2:   0.703893
cred2:  0.908058        weight: 0.862087


Color_pair[3], FineBrick and LightSeaGreen:
_______________________________________
sim1:   0.443137        sim2:   0.44288
cred2:  0.999418        weight: 0.999128


Color_pair[4], Crimson and RoyalBlue:
_______________________________________
sim1:   0.470588        sim2:   0.4525
cred2:  0.961562        weight: 0.942343


Color_pair[5], Snow and MidnightBlue:
_______________________________________
sim1:   0.224837        sim2:   0.207335
cred2:  0.92216         weight: 0.88324


************************************************
        Global weight: 0.862087
************************************************


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~TEST SET~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Color_pair[6], Tomato and Peru:
_______________________________________
sim1:   0.879739        sim2:   0.8615

cred1:  0.908058
simFin: 0.798854

************************************************
        0.798854    =<     0.8615
************************************************


Color_pair[7], Pink and Violet:
_______________________________________
sim1:   0.85098         sim2:   0.83427

cred1:  0.908058
simFin: 0.77274

************************************************
        0.77274     =<     0.83427
************************************************
```