Proceedings of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, June 12-14, 2006 (pp120-125)

On Implementation of Nested Rectangular Decision Regions by Multi-layer Perceptrons III: Applications Using the Similarity and Dissimilarity

CHE-CHERN LIN
Department of Industrial Technology Education
National Kaohsiung Normal University
116 Ho Ping First Road,Kaohsiung, 802
TAIWAN, ROC

Abstract: - In this paper, we discuss the applications of the up-down algorithm. We focus on how to apply the algorithm to implement some special decision regions according to the similarity and dissimilarity. To generalize the algorithm to solve more complex decision regions, we modify the network structure by adding a logic layer into the original neural network and discuss the partitioning capability of the modified network structure.

Key-Words: - Classification; Multi-layer perceptron; Nested decision region; Similarity; Dissimilarity

1 Handling Dissimilarities

We demonstrate how to handle the dissimilarity parts for a decision region originally formed by a nested rectangular decision region. Two popular decision regions are discussed including five-square decision region and two-spiral decision region. Below we explain how to implement them using the properties of the similarity and dissimilarity. Both of the decision regions are originally taken from the previous studies [1,2].

1.1 Five-Square decision region

Fig 1(a) is a five-square decision region taken from [1]. It is not a convex recursive deletion region and therefore cannot be implemented by the algorithm proposed in [3]. We use illustrative figures to explain its similarity and dissimilarity with respect to the 2A-2B decision region and present a modified network to implement the decision region by adding intermediate nodes to handle the dissimilarity. Comparing Figure 1(a) with the 2A-2B case, we get four dissimilarity parts: DS1, DS2, DS3 and DS4, as indicated in Figure 1(b). All of the similarity parts are of 1A-1B case and can be solved by adding four intermediate nodes into the original neural network, as shown in Figure 1(c). The weights connecting the intermediate nodes with the output node are all set to –1.

It is very important to know that if a dissimilarity part belongs to class A in the original decision region, the weight connecting the associated intermediate node with the output node is 1.

1.2. Two-spiral decision region

Figure 2 is a two-spiral decision region taken from [2]. The decision region is divided into two individual spirals: spiral 1 and spiral 2. We explain how to implement the two spirals individually using the similarity and dissimilarity parts. We first use the 2A-2B case to form the similarity parts and then handle the dissimilarity parts. Figure 3(a) shows the decision region of spiral 1. Its similarity and dissimilarity parts are indicated in Figures 3(b) and 3(c), respectively. Using the same procedure, one can get the decision region of spiral 2 and its similarity and dissimilarity parts with respect to the nested decisions. The two-spiral decision region can be implemented if we use two output nodes in the output layer for the two spirals, respectively.

2 Generalization of the Similarity

We can implement some special decision regions by taking logical manipulations of two or more rectangular decision regions. To do this we add a logic layer in the original network structure.

The added logic layer in a multi-layer perceptron serves to collect the partitioning information of the associated rectangular decision regions and performs the desired logic manipulations such as union, intersection, …, etc. This promotes the partitioning capabilities of the proposed algorithm to implement more complicated decision regions.

Below, we show three examples to explain how to implement an octagonal, star-like, and gear-like decision regions by taking logic "***AND***", "***OR***", and "***XOR***" manipulations of two rectangular decision regions. Both of the two rectangular decision regions are the case of 1A-1B decision region. The first one is a regular 1A-1B case, and the second one is a 1A-1B case with a rotation of 45 degrees.

We also discuss the nested polygonal decision regions and apply them to solve the circle problems.

2.1 Octagonal decision region

An octagonal decision region can be formed by taking a logical "AND" of two 1A-1B decision regions. Figure 4 is an illustrative example to explain this. Figure 4(a) shows an octagonal decision region and Figure 4(b) shows the two 1A-1B decision regions forming the octagonal decision region. One can implement the octagonal decision region by adding a logical layer in the original network, functioning as a logical ***"AND"***. The associated network to implement the octagonal decision region is demonstrated in Figure 4(c). In Figure 4(c), two nodes (l_1 and l_2) are added to the logical layer. The weights connecting the two nodes with the output node are 1's. l_1 produces a '1' if an input pattern belongs to class A in the original 1A-1B decision region, and a '0' if it belongs to class B. Similarly, l_2 produces a '1' if the input pattern belongs to class A in the 1A-1B decision region with rotation of 45 degrees, and a '0' if it belongs to class B. let $v = l_1 + l_2$. v must be one of the values of 0, 1, or 2. The transfer function for the output node y is as follows:

$$y = f(v) = \begin{cases} 1 & \text{if } v = 2 \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

2.2 Star-like decision region

Based on the two 1A-1B decision regions displayed in Figure 4(b), a star–like decision region is solved by taking a logic ***"OR"*** manipulation of the two 1A-1B decision regions, as shown in Figure 5(a). The transfer function for the output node y is as follows:

$$y = f(v) = \begin{cases} 1 & \text{if } v \geq 1 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

2.3 Gear–like decision region

Similarly, a gear-like decision region is solved by taking a logic ***"XOR"*** manipulation of the two 1A-1B decision regions, as shown in Figure 5(b). The transfer function for the output node y is as follows:

$$y = f(v) = \begin{cases} 1 & \text{if } v = 1 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

2.4 Nested polygonal decision region

One can easily generalize the above procedure to add a logical layer to implement nested polygonal decision regions based on particular multi-A and multi-B decision regions (***i***A-***j***B cases). Figure 6(a) is an illustrative example of nested octagonal decision region formed by two 2A-2B decision regions.

2.5 Circle problem

The circle problem presented in [4] can be implemented by a nested octagonal decision region, as indicated in Figure 6(b).

3 Conclusions

We presented the up-down algorithm to obtain the weights of TLPs, by which to implement the nested rectangular decision regions. We explained how TLPs form the decision regions and discussed the properties of the nested rectangular decision regions implemented by the proposed algorithm. We studied on how to apply the algorithm to implement some special decision regions according to the similarity and dissimilarity. To generalize the algorithm to solve more complex decision regions, we modified the network structure by adding a logic layer into the original neural network and discussed the partitioning capability of the modified network structure.

The algorithm depicts a technique to implement the decision regions by the partitioning capabilities of the multi-layer perceptrons without any training procedure. It might be interesting research topics to realize more complicated decision regions by modifying the algorithm and the network structure.

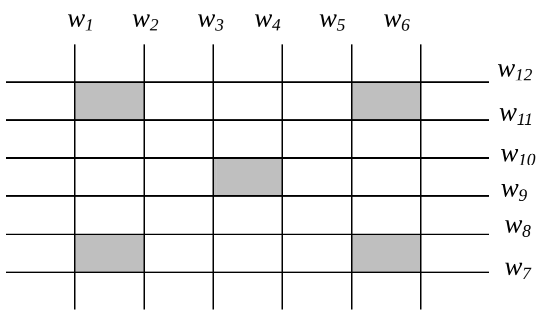
References:

[1] J. Wang, J. Rau, W. Liu, Two-Stage Clustering via Neural Networks, *IEEE Trans. on Neural Networks*, Vol. 14, No. 3, 2003, pp. 606-615.

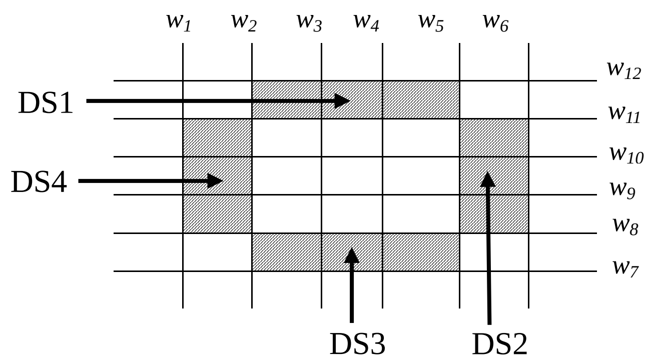
[2] K. Chen, D. Wang, Perceiving Geometric Patterns: Form Spirals to Inside-Outside Relations, *IEEE Trans. on Neural Networks*, Vol. 12, No. 5, 2001, pp. 1084-1102.

[3] C. Cabrelli, U. Molter, R. Shonkwiler, A Constructive Algorithm to Solve "Convex Recursive Deletion" (CoRD) Classification Problems via Two-Layer Perceptron Networks, *IEEE Trans. On Neural Networks*, Vol. 11, No. 3, 2000, pp. 811-816.

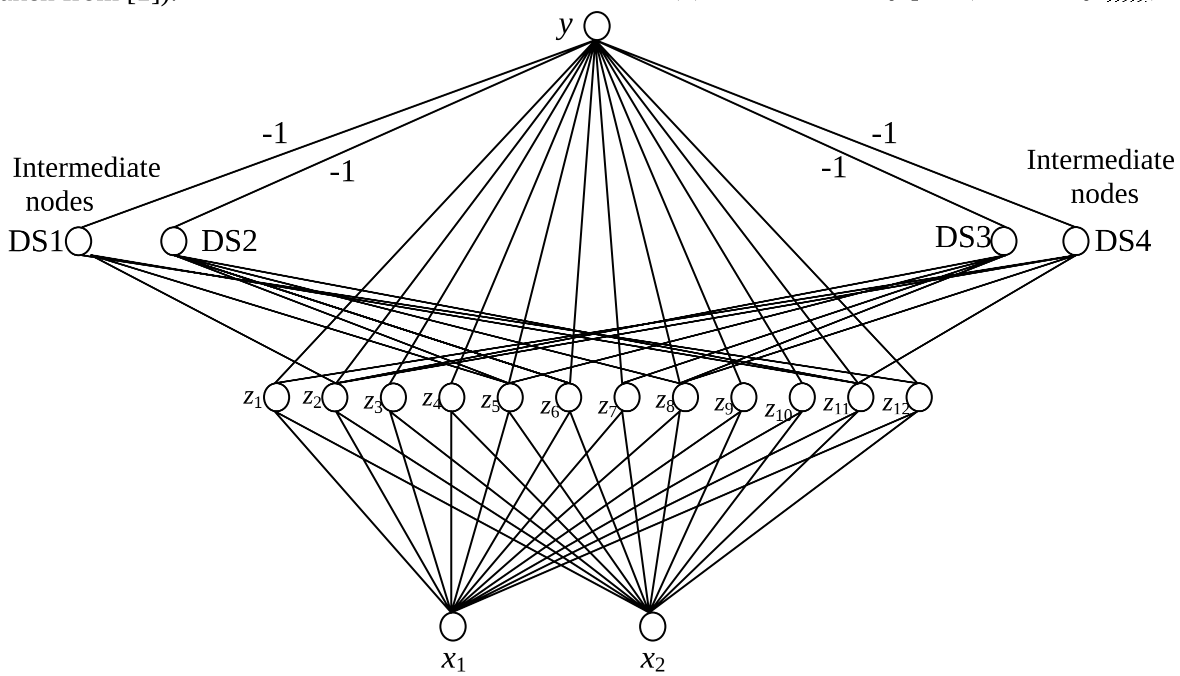
[4] R. D. Morris, A. D.. M. Garvin, Fast Probabilistic Self-Structuring of Generalized Single-Layer Networks, *IEEE Trans. on Neural Networks*, Vol. 7, No. 4, 1996, pp. 881-888.



(a) The five-square decision region (taken from [1]).



(b)The dissimilarity part (marked by ▨)



(c) The associated network to implement the five-square decision region

Figure 1: The five-square decision region and its associated network structure.

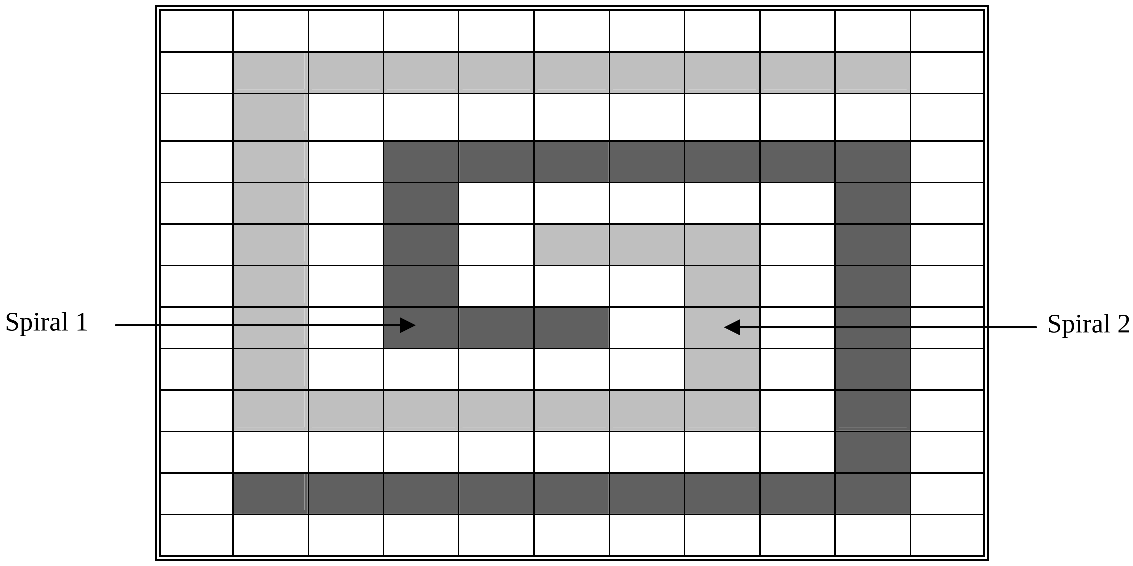
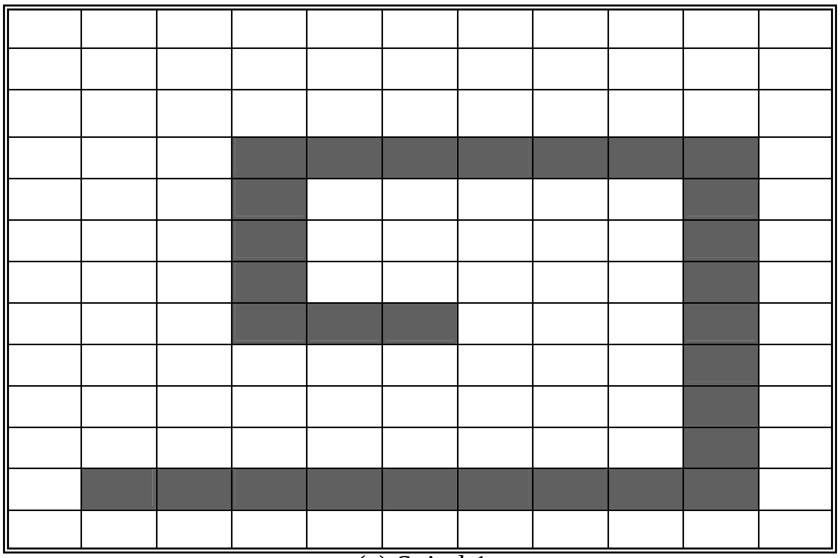
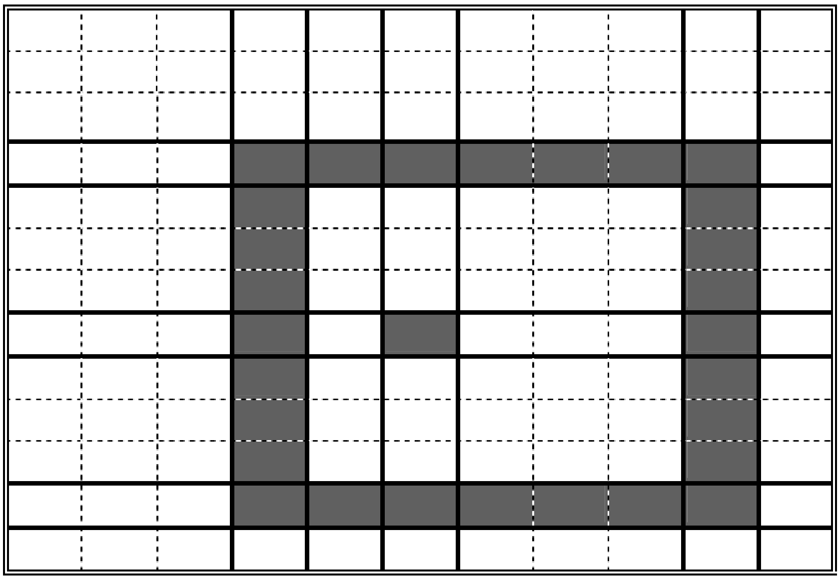


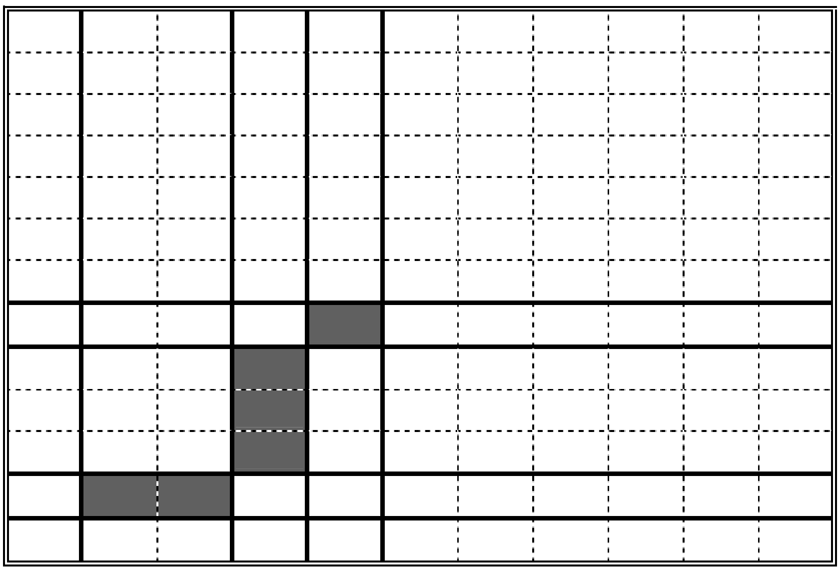
Figure 2: The two-spiral decision region (taken from [2]).



(a) Spiral 1

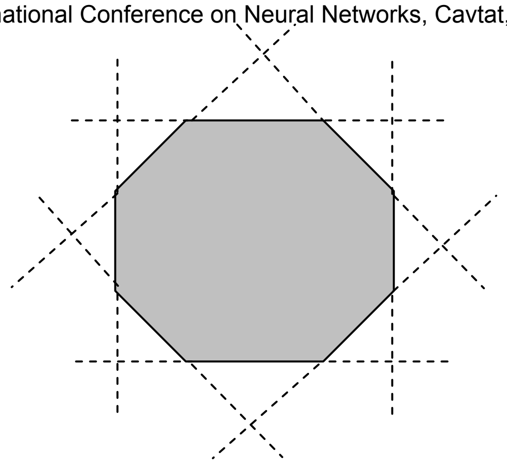


(b) The similarity parts (the 2A-2B case)

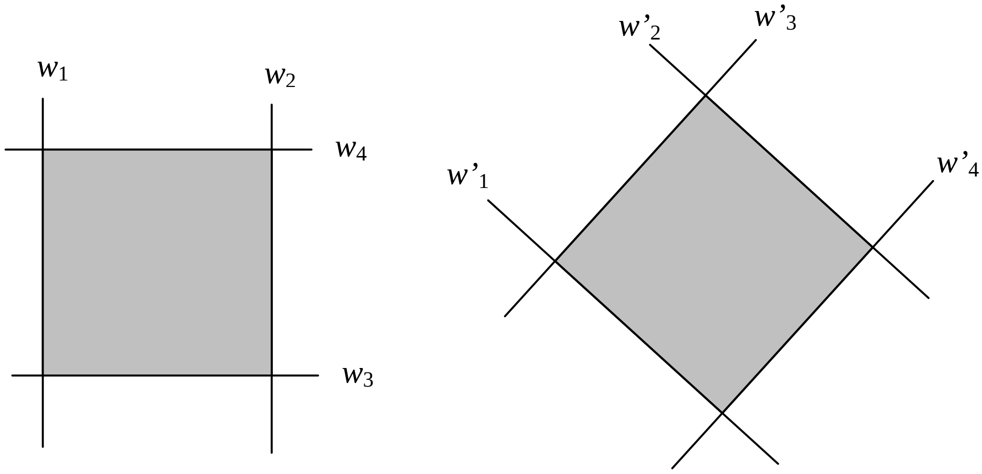


(c) The dissimilarity parts**.**

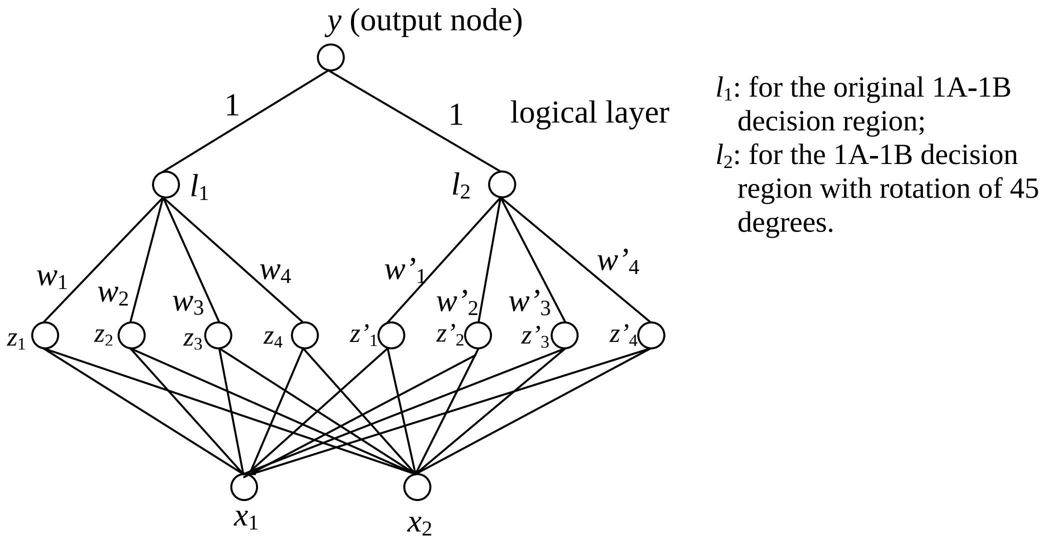
Figure 3: Spiral 1with similarity and dissimilarity parts.



(a) The octagonal decision region.

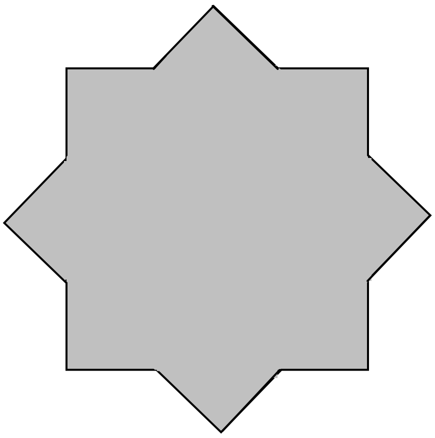


(b) The two 1A-1B decision regions used to form the octagonal decision region (left: the original 1A-1B decision region; right: the 1A-1B decision region with rotation of 45 degrees).

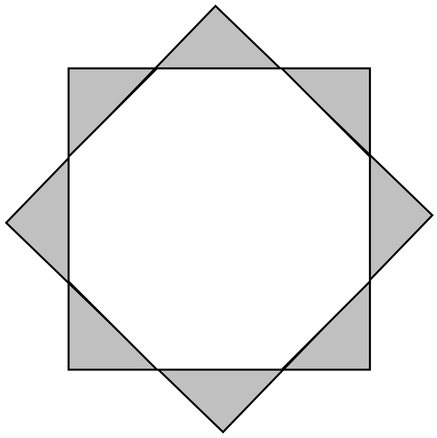


(c) The network to implement the octagonal decision region.

Figure 4: The octagonal decision region and the associated neural network.

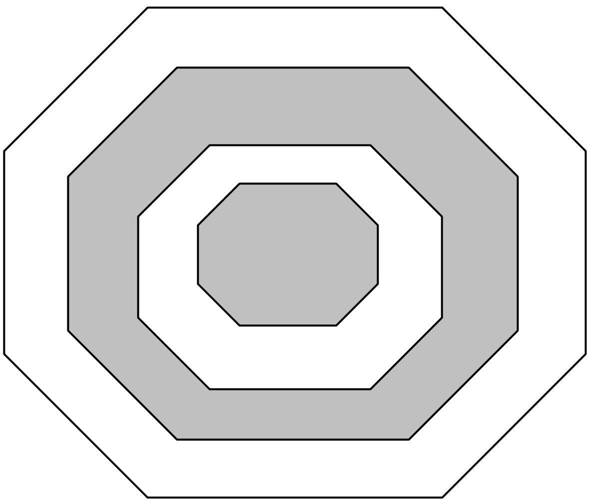


(a) The star-like decision region formed by taking a logical ***OR*** of the two 1A-1B decision regions shown in Figure 10 (b).

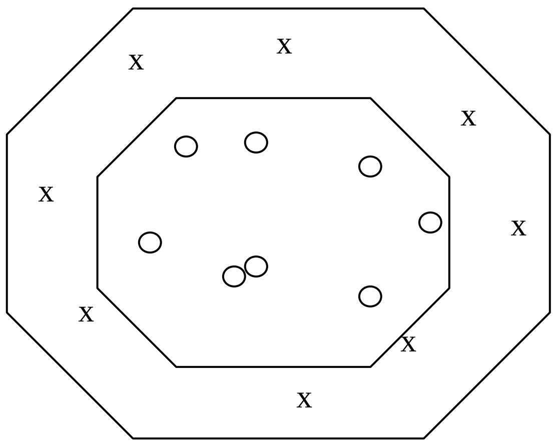


(b) The gear-like decision region formed by taking a logical ***XOR*** of the two 1A-1B decision regions shown in Figure 10 (b).

Figure 5: The stat-like and gear-like decision regions.



(a) The nested octagonal decision region



(b) The circle problem implemented by a nested octagonal decision region (pattern configuration originally from [4]).

Figure 6: The nested octagonal decision region and circle problem implemented by a nested octagonal decision region.