2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Fuzzy Perceptron with Pocket Algorithm in Postoperative Patient Data Set

Suwannee Phitakwinai,
Computer Engineering Department,
Faculty of Engineering,
Chiang Mai University,
Chiang Mai, Thailand
suwannee.phi@gmail.com

Sansanee Auephanwiriyakul*,
Senior Member, IEEE,
Computer Engineering Department,
Faculty of Engineering,
Biomedical Engineering Center,
Chiang Mai University,
Chiang Mai, Thailand
sansanee@ieee.org

Nipon Theera-Umpon,
Senior Member, IEEE,
Electrical Engineering Department,
Faculty of Engineering,
Biomedical Engineering Center,
Chiang Mai University,
Chiang Mai, Thailand
nipon@ieee.org

***Abstract*—Classification is one of the problems in pattern recognition. Most of the time this problem will deal with data sets that are in numeric form and represented by vectors of numbers. Since there might be uncertainties embedded in a data set, it is more natural to represent the data set as fuzzy vectors. Hence, in this paper, we develop a fuzzy perceptron with pocket algorithm for fuzzy vectors. This algorithm is based on the extension principle and the decomposition theorem. We implement this algorithm on both synthetic and a real-world data set, i.e., the postoperative patient data. We also compare the result from the fuzzy perceptron with pocket algorithm with that from the regular perceptron with pocket algorithm. The comparison is done on the fuzzy perceptron with and without pocket as well.**

Keywords—Fuzzy Pocket; Fuzzy Perceptron; Postoperative Patient Data set

I. INTRODUCTION

Normally, each sample in classification problem can be in the form of binary values or real numbers. The vector of numbers corresponding to an appropriate linear space is a representation of this data. Most of the classifiers are developed based on numeric mathematical model. However, there are usually uncertainties in data resulting from several reasons, e.g., imprecision measurement in the data collection, natural language, and etc. This uncertainty can be best modeled by a fuzzy set. A vector of fuzzy subsets in the Euclidean space is called a fuzzy vector or a linguistic vector.

There have been several studies on extending the numeric neural networks to work with fuzzy inputs. However, some of them did not show any calculation or perform any experiment [1, 2]. Some other algorithms [3–9] were to calculate numeric weights or triangular weights, but not general fuzzy weights from fuzzy inputs. There were some other works on fuzzy rule-based systems with fuzzy inputs [10–13]. However, our interest in this paper is on the extension of regular algorithm using the extension principle theorem.

In our previous works [14–16], we described the generalized Choquet fuzzy integral (GCFI) and applied it to pattern recognition and information fusion. We introduced a linguistic fuzzy C-means (LFCM) [17] that worked with fuzzy inputs without any defuzzification methods. It is our belief that developing algorithms that deal with linguistic vectors without using defuzzification methods produces meaningful results. In [18, 19], we introduced a fuzzy perceptron that is a fuzzy version of the regular perceptron. However, we only performed experiments on synthetic data sets. The result showed that it was worth doing further investigation on real-world data sets. However, the nature of real-world data set is not usually linear separable. Hence, we propose a fuzzy perceptron with pocket algorithm in this work. We test the algorithm with a real-world data set, i.e., the postoperative patient data.

II. FUZZY PERCEPTRON WITH POCKET ALGORITHM

A. Background

A fuzzy number $\boldsymbol{A}$ is a normal convex fuzzy set defined on the real line, R [20]. The support of $\boldsymbol{A}$ ($\Gamma_{\boldsymbol{A}} = [a_1, a_2]$) is bounded in R. If $0 < a_1 \le a_2$ holds for $\Gamma_{\boldsymbol{A}}$ then $\boldsymbol{A}$ is called positive [21]. Similarly, $\boldsymbol{A}$ is called negative if $a_1 \le a_2 < 0$ and zero if $a_1 \le 0 < a_2$ [21]. Or $\boldsymbol{A}$ is called non-negative if $\mu_{\boldsymbol{A}}(x) = 0$ for all $x < 0$, [22]. Suppose that $\boldsymbol{A}$ and $\boldsymbol{B}$ are two fuzzy numbers. Let the symbol $\otimes$ denote any of the algebraic operations +, –, × or ÷. According to the extension principle [23], the algebraic operation will map fuzzy sets in the fuzzy power set of R ($\Im(\mathrm{R})$) to a fuzzy subset in $\Im(\mathrm{R})$ producing another fuzzy number $\boldsymbol{Z}$. The result's membership function is given by:

$$\mu_{\boldsymbol{Z}}(y) = \sup_{x_1 \otimes x_2 = y} \min\left(\mu_{\boldsymbol{A}}(x_1), \mu_{\boldsymbol{B}}(x_2)\right), \qquad (1)$$

where x_1 and x_2 satisfy the mapping constraint. However, the resultant fuzzy set is often irregular and inaccurate compared with the exact result if the continuous support is discretized into finite number of points and approximate the fuzzy set on the discrete domain [24–26]. Hence, the operation is perform based on interval arithmetic [21, 24–26] and the decomposition theorem [20]. For a fuzzy set ($\boldsymbol{A} \otimes \boldsymbol{B}$), its

α-cut ($[\boldsymbol{A}\otimes\boldsymbol{B}]_\alpha$) equals $[\boldsymbol{A}]_\alpha \otimes [\boldsymbol{B}]_\alpha$ for all $\alpha\in(0, 1]$ [20]. Then, by the decomposition theorem,

$$\boldsymbol{A}\otimes\boldsymbol{B} = \bigcup_{\alpha\in[0,1]}[\boldsymbol{A}\otimes\boldsymbol{B}]_\alpha . \quad (2)$$

Since, for any of these operations, $[\boldsymbol{A}\otimes\boldsymbol{B}]_\alpha$ is a closed interval for each $\alpha\in(0, 1]$ and $\boldsymbol{A}$, $\boldsymbol{B}$ are fuzzy numbers, $\boldsymbol{A}\otimes\boldsymbol{B}$ is also a fuzzy number.

For the collection of non-interactive fuzzy vectors, Mares [27] has shown that with $\vec{\mathbf{0}}$ (the vector of singleton fuzzy number **0**), $\vec{\mathbf{1}}$ (vector of singleton fuzzy number **1**), and component-wise addition and scalar multiplication, this forms a vector space. Also, with appropriate definitions of distance, these spaces exhibit the properties of metric spaces [22, 28].

B. Fuzzy Perceptron with Pocket Algorithm

To deal with fuzzy vectors, a fuzzy perceptron is developed by extending the perceptron algorithm [29–31]. Now, we will briefly describe our fuzzy perceptron [18, 19]. Let $\mathbf{X} = \{\vec{\boldsymbol{X}}_j \mid 1 \le j \le N\}$ be a set of non-interactive fuzzy vectors in p-dimensional space, $\vec{\boldsymbol{X}}_j = (\boldsymbol{X}_{j1},\ldots,\boldsymbol{X}_{jp})^t \in [\Im(\mathrm{R})]^p$. Suppose there exists a fuzzy weight vector $\vec{\boldsymbol{W}} = (\boldsymbol{W}_1,\ldots,\boldsymbol{W}_{p+1})^t$ such that

$$\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}}_j = \left(\boldsymbol{W}_1\boldsymbol{X}_{j1} + \boldsymbol{W}_2\boldsymbol{X}_{j2} + \ldots + \boldsymbol{W}_p\boldsymbol{X}_{jp} + \boldsymbol{W}_{p+1}\boldsymbol{X}_{jp+1}\right) > 0 \quad \text{if } \vec{\boldsymbol{X}}_j \text{ is in class 1} \quad (3a)$$

$$\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}}_j = \left(\boldsymbol{W}_1\boldsymbol{X}_{j1} + \boldsymbol{W}_2\boldsymbol{X}_{j2} + \ldots + \boldsymbol{W}_p\boldsymbol{X}_{jp} + \boldsymbol{W}_{p+1}\boldsymbol{X}_{jp+1}\right) \le 0 \quad \text{if } \vec{\boldsymbol{X}}_j \text{ is in class 2} \quad (3b)$$

where $\vec{\boldsymbol{X}}_j = (\boldsymbol{X}_{j1},\ldots,\boldsymbol{X}_{jp}, \mathbf{1})^t$ is an augmented feature vector j with a singleton fuzzy number **1**. From the decomposition theorem, the left side of equations (3a) and (3b) is transformed to

$$\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}}_j = \bigcup_{\alpha\in[0,1]}\left([\boldsymbol{W}_1]_\alpha[\boldsymbol{X}_{j1}]_\alpha + \ldots + [\boldsymbol{W}_p]_\alpha[\boldsymbol{X}_{jp}]_\alpha + [\boldsymbol{W}_{p+1}]_\alpha[\mathbf{1}]_\alpha\right) \quad (4)$$

The inner product between a fuzzy weight vector and fuzzy input vector produces a positive fuzzy number if $\vec{\boldsymbol{X}}_j$ is in class 1, and if $\vec{\boldsymbol{X}}_j$ is in class 2, $\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}}_j$ is a negative or zero fuzzy number. For the simplicity of the training process, we also multiplied each class 2 fuzzy vectors with a numeric number -1, i.e.,

$$\vec{\boldsymbol{X}}_j' = -1\times\vec{\boldsymbol{X}}_j \quad \text{for all } \vec{\boldsymbol{X}}_j \text{ that are in class 2}. \quad (5)$$

Since the input feature vector is a non-interactive fuzzy vector, this multiplication is done in each dimension separately. For each dimension k and the decomposition theorem,

$$-1\times\boldsymbol{X}_{jk} = \bigcup_{\alpha\in[0,1]}\left(-1\times[\boldsymbol{X}_{jk}]_\alpha\right) \quad \text{for all } \vec{\boldsymbol{X}}_j \text{ that are in class 2}. \quad (6)$$

Hence, for all fuzzy input vectors,

$$\vec{\mathbf{W}}^t\vec{\mathbf{X}}_j > \mathbf{0} \quad \text{for all } j = 1,\ldots,N, \quad (7)$$

i.e., the inner product in equation (7) produces a positive fuzzy number.

The fuzzy hyperplane ($\boldsymbol{H}$) is defined by $\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}} = \mathbf{0}$ where all class 1 feature vectors lie on one side and all class 2 feature vectors lie on the other side of the hyperplane. In order to understand this definition better, let us first consider the 2-dimensional case. If fuzzy hyperplane is a fuzzy number $\in [\Im(\mathrm{R})]^2$, then it is a fuzzy line [32], i.e.,

$$\boldsymbol{H} = \vee\{\alpha|(x_1, x_2)\in\Omega\}, \quad (8)$$

where $\Omega = \{(x_1, x_2) \mid [\boldsymbol{W}_1]_\alpha x_1 + [\boldsymbol{W}_2]_\alpha x_2 + [\boldsymbol{W}_3]_\alpha = 0\}$, $\forall\alpha\in[0,1]$ and $(x_1, x_2)\in\mathrm{R}^2$. If the fuzzy hyperplane is a fuzzy number $\in [\Im(\mathrm{R})]^p$, then

$$\boldsymbol{H} = \vee\{\alpha|(x_1,\ldots, x_p)\in\Omega\}, \quad (9)$$

where $\Omega = \{(x_1,\ldots, x_2) \mid [\boldsymbol{W}_1]_\alpha x_1 + [\boldsymbol{W}_2]_\alpha x_2 + \ldots + [\boldsymbol{W}_p]_\alpha x_p + [\boldsymbol{W}_{p+1}]_\alpha = 0\}$, $\forall\alpha\in[0,1]$ and $(x_1,\ldots, x_p)\in\mathrm{R}^p$. The perceptron algorithm is then to find a fuzzy weight vector as follows:

1. If $\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}} \le \mathbf{0}$ or $\vec{\boldsymbol{W}}^t\vec{\boldsymbol{X}}$ is a negative or zero fuzzy number then

$$\vec{\boldsymbol{W}}(n+1) = \vec{\boldsymbol{W}}(n) + \eta\vec{\boldsymbol{X}}(n), \quad (10)$$

where η is a positive constant (numeric number) and n is a time step. For each dimension k,

$$\boldsymbol{W}_k(n+1) = \boldsymbol{W}_k(n) + \eta\boldsymbol{X}_k(n). \quad (11a)$$

From the decomposition theorem, equation (11a) is transformed to

$$\boldsymbol{W}_k(n) + \eta\boldsymbol{X}_k(n) = \bigcup_{\alpha\in[0,1]}\left([\boldsymbol{W}_k(n)]_\alpha + \eta[\boldsymbol{X}_k(n)]_\alpha\right). \quad (11b)$$

2. Otherwise, $\vec{\boldsymbol{W}}$ remains the same.

We use the similarity [33] of the weight to compute the stopping criteria of our fuzzy perceptron algorithm. Let $\overrightarrow{\boldsymbol{OW}}$ be the old weight, $\vec{\boldsymbol{W}}$ be the new weight; $\overrightarrow{\boldsymbol{OW}}$ and $\vec{\boldsymbol{W}} \in [\Im(\mathrm{R})]^p$ and ε be a small positive number. The dissimilarity between the old weight and the new weight is:

$$Q = \max_{1\le j\le p+1}\left(1 - \frac{card(\boldsymbol{OW}_j \cap \boldsymbol{W}_j)}{card(\boldsymbol{OW}_j \cup \boldsymbol{W}_j)}\right). \quad (12)$$

The algorithm will stop if:

$$Q < \varepsilon \quad (13)$$

or the weight is not updated for N fuzzy feature vectors.

For a data set that is not linearly separable data set, Gallant [34] introduced the pocket algorithm that helps the perceptron to converge to an optimal solution. Therefore, we utilize the pocket algorithm in our fuzzy perceptron as well. However, the postoperative patient data set has an unbalanced number of samples in each class. To reduce the bias from unbalanced numbers, we modify the pocket algorithm as follows:

1. The correct classification of either class has to be increasing, and

2. The total number of correct classification has to be increasing.

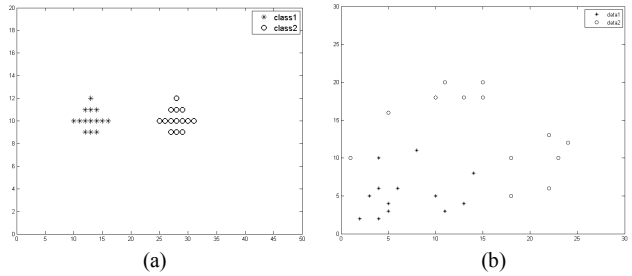
The algorithm of fuzzy perceptron with the pocket algorithm is as follows:

Initial weight fuzzy vector
Define a stored weight ($\vec{W}_s$)
Define history of total correct classification (h_s)
Define history of class 1 correct classification (h_{s1})
Define history of class 2 correct classification (h_{s2})
Do {
  Set Update Flag to false.
  For all vector $\vec{X}_j$, j = 1,…,N
    If $\vec{W}^t \vec{X}_j$ is negative or zero fuzzy number **then**
      Update weight fuzzy vector using equations (10), (11a) and (11b)
      Set Update Flag to true.
  End For
  Compute class 1 correct classification (P_{c1})
  Compute class 2 correct classification (P_{c2})
  Compute total correct classification (P_c)
  If (P_{c1}> h_{s1} or P_{c2}> h_{s2}) and (P_c> h_s) **then**
    h_{s1} = P_{c1}
    h_{s2} = P_{c2}
    h_s = P_c
    $\vec{W}_s = \vec{W}$
}**Until** weight stabilize using equations (12) and (13) or Update Flag is false.

III. Experimental Results

A. Synthetic Data Set

In this section, we perform experiments on two synthetic examples shown in figure 1. For each data set in figure 1, we fuzzify each dimension of each data point to be a trapezoid as shown in figure 2. We generate the width on the right and left of the core (w_r, w_l) using Gaussian distribution N(μ, σ^2). The width of the spread (a) is equal to the mean of the distribution (μ) that generates w_r and w_l of the core [17]. The first data set is fuzzified into fuzzy vector data set using N(0.1,0.5), while the second data set is fuzzified using N(0.01,0.01).



(a) (b)

Fig. 1. (a) First synthetic data set and (b) Second synthetic data set.

We apply the regular perceptron with pocket on the peaks (the coordinates in figure 1) followed by the fuzzy perceptron with pocket on the fuzzy vectors. In all the computer simulations, we randomly initialize weight to be a singleton fuzzy vector. The learning rate is set to 0.0001 for all experiments of both data sets. The stopping criterion for both algorithms is set to a maximum iteration count of 1000 or $\varepsilon=10^{-4}$. To make a comparison, we implement fuzzy perceptron without pocket on both data sets with the same setting, as well.

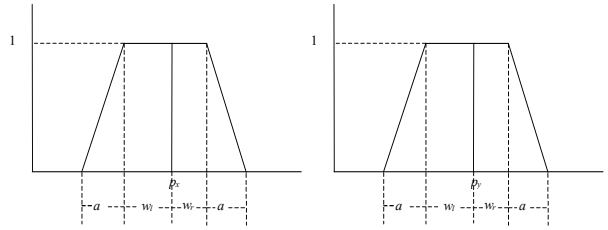
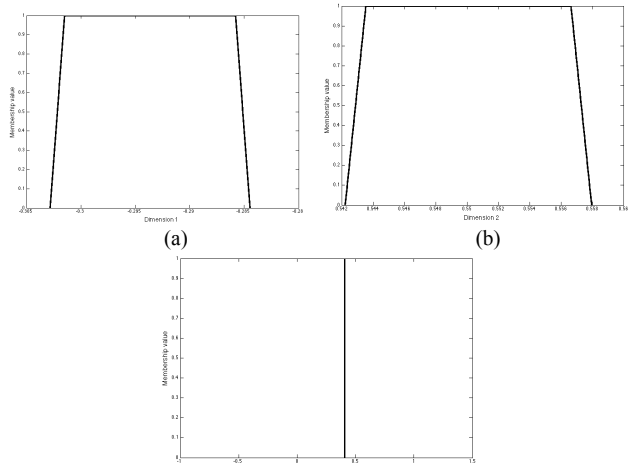


Fig. 2. Data with peak (p_x, p_y) and (w_r, w_l) generated from normal distribution.

In all of the experiments, we randomly initialize weight to be a singleton fuzzy vector. For the first data set, the initialized weight vector is [0.0765, 0.6857, 0.4191]. Table 1 shows the confusion matrix of fuzzy peceptron with pocket on the first data set. The confusion matrix of the regular perceptron with pocket and the fuzzy perceptron without pocket are similar to the one shown in table 1. All of the algorithms yield 100% correct classification rate. Both fuzzy perceptron with and without pocket are converged at iteration 133. The final weight vector from the fuzzy perceptron with pocket is shown in figure 3. Because the input fuzzy vector from this data set is $\in [\Im(R)]^2$, the fuzzy hyperplane is a fuzzy line. The support of this fuzzy line and the support of the fuzzy input vectors are shown in figure 4. The final weight vector ([−0.2566,0.5635,0.4072]) from the regular one is from iteration 120.

TABLE I. Confusion Matrix of Fuzzy Perceptron with Pocket on the First Data Set

		Desired Output	
		Class 1	Class 2
Program Output	Class 1	14	0
	Class 2	0	14



(a) (b)

(c)

Fig. 3. The final weight vector from the fuzzy perceptron with pocket on the first data set (a) first, (b) second, and (c) third (bias) dimensions

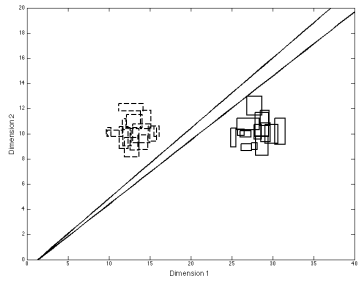


Fig. 4. Supports of fuzzy input vectors from 2 classes and fuzzy hyperplane for the first data set

TABLE II. CONFUSION MATRIX OF FUZZY PERCEPTRON WITH POCKET ON THE SECOND DATA SET

		Desired Output	
		Class 1	Class 2
Program Output	Class 1	12	1
	Class 2	1	12

TABLE III. CONFUSION MATRIX OF FUZZY PERCEPTRON WITHOUT POCKET ON THE SECOND DATA SET

		Desired Output	
		Class 1	Class 2
Program Output	Class 1	11	1
	Class 2	2	12

TABLE IV. CONFUSION MATRIX OF REGULAR PERCEPTRON WITH POCKET ON THE SECOND DATA SET

		Desired Output	
		Class 1	Class 2
Program Output	Class 1	13	1
	Class 2	0	12

For the second data set, we initialize weight vector to [0.3848, 0.4388, 0.8923]. Tables 2, 3 and 4 show the confusion matrix of the fuzzy perceptron with pocket, fuzzy perceptron without pocket, and regular perceptron with pocket respectively. We can see that fuzzy perceptron without pocket perform worse than that with pocket as expected. The fuzzy perceptron with pocket yields 92.31% correct classification rate while the one without pocket provides 88.46%. However, the regular perceptron with pocket performs better than the fuzzy perceptron with and without pocket. The correct classification from the regular one is 96.15%. The final weight vector from the fuzzy perceptron with pocket shown in figure 5 is the weight vector at iteration 384. Again, the fuzzy hyperplane is a fuzzy line shown in figure 6. The fuzzy perceptron without pocket is stopped by the maximum iteration. The final weight vector ([–0.0365, –0.0429, 0.8594]) from the regular one comes from iteration 451.

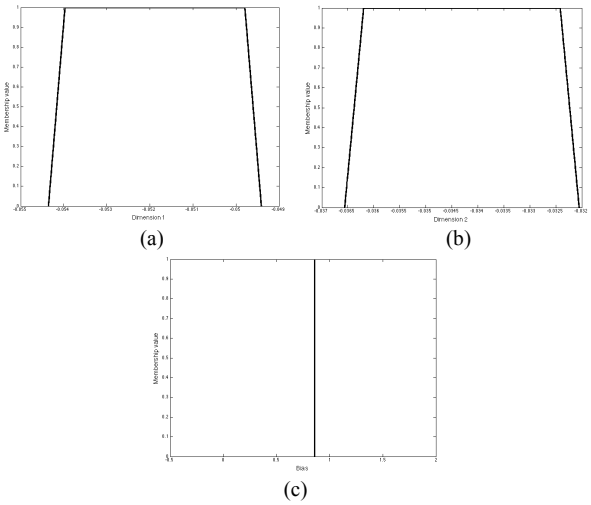


Fig. 5. The final weight vector from the fuzzy perceptron with pocket on the second data set (a) first, (b) second, and (c) third (bias) dimensions

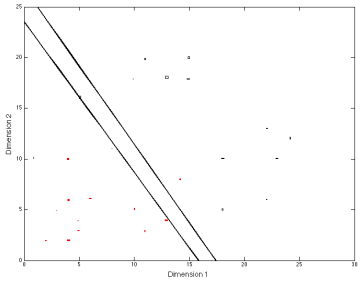


Fig. 6. Supports of fuzzy input vectors from 2 classes and fuzzy hyperplane for the second data set

B. Postoperative Patient Data Set

The postoperative patient data set [35] was collected by Sharon Summers of the University of Kansas and Linda Woolery of the University of Missouri-Columbia. It was made available to public by Jerzy W. Grzymala-Busse of the University of Kansas. Based on the hypothermia condition, the classification task in this data set is to group patients in a postoperative recovery into Intensive Care Unit (I class), general hospital floor (A class) or go home (S class). In this data set, there are 2, 64, and 24 patients in I, A, and S classes, respectively. Since, there are not enough samples for the training the fuzzy perceptron with pocket algorithm in the I class, we disregard this class from the experiments.

There are eight linguistic attributes as follows:

- Patient's internal temperature in degrees Celsius (L-CORE):
  high (>37), mid (>=36 and <37), low (<36).
- Patient's surface temperature in degrees Celsius (L-SURF):
  high (>36.5), mid (>=36.5 and <35), low (<35)
- Patient's oxygen saturation in percentile (L-O2):
  excellent (>=98), good(>=90 and <98), fair(>=80 and <90), poor(<80).
- Patient's last measurement of blood pressure (L-BP):
  high (<130), mid (<=130 and >= 90), low (<90).

- Stability of patient's surface temperature (SURF-STBL):
  stable, mod-stable, unstable.
- Stability of patient's core temperature (CORE-STBL):
  stable, mod-stable, unstable.
- Stability of patient's blood pressure (BP-STBL):
  stable, mod-stable, unstable
- patient's comfort at discharge (COMFORT):
  integer number between 0 – 20.

Although there are 8 features, some of patients in this data set have a missing last attribute. Hence, we disregard the last attribute in our experiments. Therefore, there are 88 patients with seven attributes. Figure 7 shows a fuzzy value (based on expert's opinion) of each attribute.

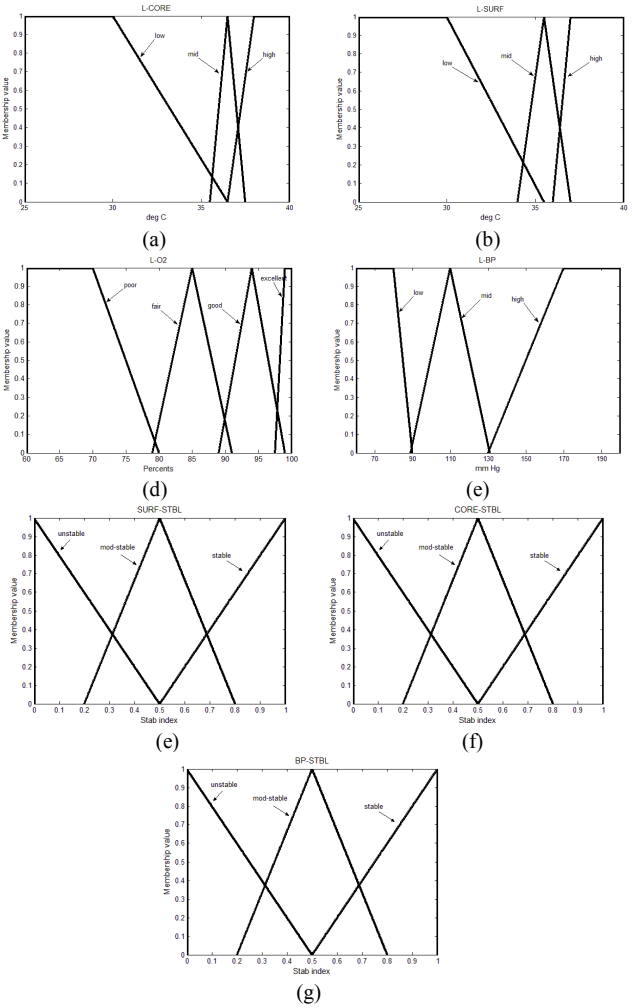


Fig. 7. Fuzzy values of attributes (a) L-CORE, (b) L-SURF, (c) L-O2, (d) L-BP, (e) SURF-STBL, (f) CORE-STBL and (g) BP-STBL.

We apply the fuzzy perceptron with pocket algorithm on this data set followed by a regular perceptron with pocket algorithm on the centroid of each fuzzy attribute value and on the numeric value (mapped from each fuzzy attribute value, shown in table 5). Again to make a comparison, we implement the fuzzy perceptron without pocket on this data set, as well.

In all experiments, we randomly initialize weight to be a singleton fuzzy vector. For all experiments, we use the same initialized weight vector set ([0.6929, 0.1528, 0.5337, 0.1604, 0.0336, 0.9554, 0.1163, 0.5232]). The learning rate is set to 0.0003 for all experiments. The stopping criterion is set to a maximum iteration count of 3000 or $\varepsilon=10^{-4}$.

TABLE V. FUZZY ATTRIBUTE VALUE TO NUMERIC NUMBER MAPPING

Attribute	fuzzy label	Numeric number
L-CORE	low	1
	mid	2
	high	3
L-SURF	low	1
	mid	2
	high	3
L-O2	poor	1
	fair	2
	good	3
	excellent	4
L-BP	low	1
	mid	2
	high	3
SURF-STBL	unstable	1
	mod-stable	2
	stable	3
CORE-STBL	unstable	1
	mod-stable	2
	stable	3
BP-STBL	unstable	1
	mod-stable	2
	stable	3

The final weight vector of the fuzzy perceptron with pocket algorithm is from iteration 76. While the fuzzy perceptron without pocket is converged at iteration 3000. The confusion matrix of the fuzzy perceptron with and without pocket algorithms are shown in table 6 and 7, respectively. From the confusion matrix, we can see that the correct classification from the fuzzy perceptron with pocket is 53.41%, while that from the one without pocket is 52.27%. The final fuzzy weight vector from the fuzzy perceptron with pocket is also shown in figure 8.

The regular perceptron with pocket algorithm of centroid and numeric value are converged at iterations 2474 and 2980, respectively. The confusion matrix of the regular perceptron with pocket algorithm on centroid and numeric value are shown in tables 8 and 9, respectively. From table 8, we can see that the correct classification from the centroid is 65.91%, whereas that from numeric value is 68.18%. The final weight vector from centroid is [0.2613, 0.0398, –0.1541, 0.0271, 0.0303, 0.9660, 0.0916, 0.5142], whereas that from numeric value is [0.2396, –0.1514, –0.0591, –0.1378, –0.1524, 0.3260, –0.0763, –0.0201]

From all the experiments, we can see that the correct classification is around 53% – 68% from both fuzzy data set and regular data set. This might be because there are overlapping between these two classes. For the indirectly comparison, in the report from [36, 37], they used the LERS algorithm (LEM2) that achieved 48% correct classification for three classes. Therefore, all four experiments show better results.

TABLE VI. CONFUSION MATRIX OF FUZZY PERCEPTRON WITH POCKET

		Desired Output	
		A class	S class
Program Output	A class	41	18
	S class	23	6

TABLE VII. CONFUSION MATRIX OF FUZZY PERCEPTRON WITHOUT POCKET

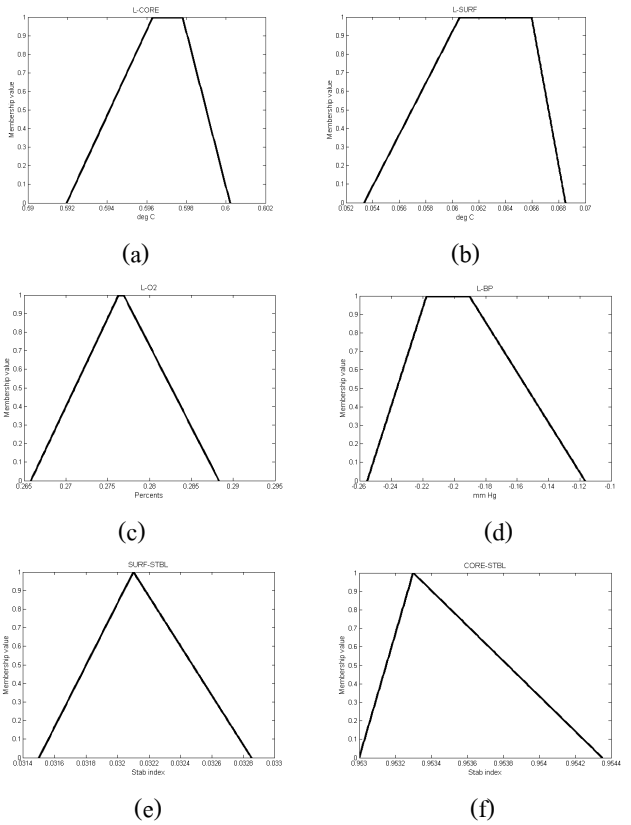
		Desired Output	
		A class	S class
Program Output	A class	36	14
	S class	28	12

TABLE VIII. CONFUSION MATRIX OF REGULAR PERCEPTRON WITH POCKET ON CENTROID

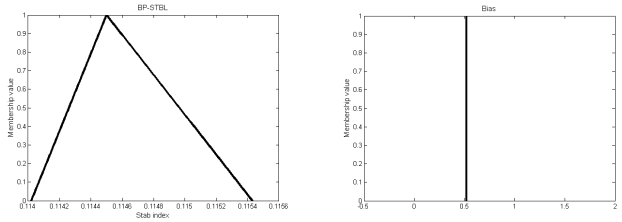
		Desired Output	
		A class	S class
Program Output	A class	48	14
	S class	16	10

TABLE IX. CONFUSION MATRIX OF REGULAR PERCEPTRON WITH POCKET ON NUMERIC VALUE

		Desired Output	
		A class	S class
Program Output	A class	53	17
	S class	11	7



(a) (b) (c) (d) (e) (f)



(g) (h)

Fig. 8. Final weight fuzzy vectors (a) L-CORE, (b) L-SURF, (c) L-O2, (d) L-BP, (e) SURF-STBL, (f) CORE-STBL, (g) BP-STBL, and (h) bias.

Although, the correct classification from the fuzzy perceptron with pocket algorithm is not as good as the regular perceptron with pocket on centroid or numeric value, it still has an advantage over the regular one. This is because if the mapping in table 5 or the defuzzification method of the attribute is changed, the classification result will be different definitely. We also lose some information when we map the fuzzy attribute into number in the first place. Hence, the resulting hyperplane will not reflect the real hyperplane. In addition, from the comparison between the fuzzy perceptron with and without pocket experiments, the one with pocket always performs better than the one without pocket as expected.

IV. CONCLUSION

In this paper, we develop a variant version of our fuzzy perceptron, i.e., a fuzzy perceptron with pocket algorithm. We implement this algorithm on both synthetic data set and a real-world fuzzy data set, i.e., the postoperative patient data set. To compare the result with the conventional method, we implement the regular perceptron with pocket on the peaks in synthetic data sets and on the defuzzified fuzzy attribute in the postoperative patient data set. In a real world data set, we defuzzify each fuzzy attribute using centroid and we also map each fuzzy attribute to numeric value. We found out that the correct classification of the fuzzy perceptron with pocket is comparable with their regular counterpart. The fuzzy perceptron do not perform as well as the one with pocket.

ACKNOWLEDGMENT

The authors would like to thank Miss Saichon Preunglampoo for providing the knowledge on hypothermia. This research is supported by the Ministry of Education and the Thailand Research Fund under the contract No. MRG4780053. We appreciate Chiang Mai University for financial support

REFERENCES

[1] M. Gupta and J. Qi, "On Fuzzy Neuron Models", *Proceedings of Inter. Joint Conference on Neural Networks* (*IJCNN*91), Seattle, 1991, 431-436.

[2] Y. Hayashi, J. J. Buckley, and E. Czogala, "Fuzzy Neural Network with Fuzzy Signals and Weights", *International Journal of Intelligent Systems*, vol. 8, 1993, 527-537.

[3] H. Ishibuchi, R. Fujioka, and H. Tanaka, "An Architecture of Neural Networks for Input Vectors of Fuzzy Numbers", *Proceedings, FUZZ-IEEE*, San Diego, 1992, 1293-1300.

[4] H. Ishibuchi, R. Fujioka, and H. Tanaka, "Neural Networks that Learn from If-Then Rules", *IEEE Trans. on Fuzzy Systems*, vol. 1, no. 2, 1993, 85-97.

[5] H. Ishibuchi, K. Kwon, and H. Tanaka, "A Learning Algorithm of Fuzzy Neural Networks with Triangular Fuzzy Weights", *Fuzzy Set and Systems*, vol. 71, 1995, 277-293.

[6] J. L. Chen and J. Y. Chang, "Fuzzy Perceptron Learning and Its Application to Classifiers with Numerical Data and Linguistic Knowledge", *Proc. IEEE Int. Conf. Neural Networks*, Perth, Australia, 1995, 3129-3133.

[7] J. L. Chen and J. Y. Chang, "Fuzzy Perceptron Neural Networks for Classifiers with Numerical Data and Linguistic Rules as Inputs", *IEEE Trans. On Fuzzy Systems*, vol. 8, no. 6, 2000, 730-745.

[8] H. M. Lee and W. T. Wang, "Fuzzy Pocket Algorithm: A Generalized Pocket Algorithm for Classification of Fuzzy Inputs", *Proc. IEEE Int. Conf. Neural Networks*, 1993, 2873-2876.

[9] H. M. Lee and W. T. Wang, "A Neural Network Architecture for Classification of Fuzzy Inputs", *Fuzzy Sets and Systems*, vol. 63, 1994, 159-173.

[10] S. Galichet, and L. Foulloy, "Fuzzy Control with Non-Precise Inputs", *5th Inter. Conf. On Information Processing and Management in Knowledge-Based Systems*, Paris, France, July 1994, pp. 84-89.

[11] R. Palm, and D. Driankov, "Fuzzy Inputs", *Fuzzy Sets and Systems*, vol. 70, 1995, pp.315-335.

[12] L. Ughetto, D. Dubois, and H. Parde, "Efficient Inference Procedures with Fuzzy Inputs", *Proceedings, FUZZ-IEEE*, Barcelona, Spain, July, 1997, pp. 567-572.

[13] R. R. Yager, and D. P. Filev, "Modeling Fuzzy Logic Controllers having Noisy Inputs, *Proceedings, FUZZ-IEEE*, Orlando, June, 1994, pp.1702-1707.

[14] S. Auephanwiriyakul, J. M. Keller, and P. D. Gader, "Generalized Choquet Fuzzy Integral Fusion", *Information Fusion*, vol. 3, 2002, 69-85.

[15] J. M. Keller, S. Auephanwiriyakul, and A. Adrian, "Linguistic Classifiers with Application to Management Questionnaires", *Proceedings, FUZZ-IEEE*, San Antonio, May 2000, pp 387-392.

[16] J.M. Keller, S. Auephanwiriyakul, and P. D. Gader, "New Fuzzy Set Tools to Aid in Predictive Sensor Fusion" *Proceedings SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Target V*, Orlando Florida, April 2000, pp.1497-1507.

[17] S. Auephanwiriyakul, J. M. Keller, "Analysis and Efficient Implementation of a Linguistic Fuzzy C-Means", *IEEE Trans on Fuzzy Systems*, vol. 10, no. 5, 2002, 563-583.

[18] S. Auephanwiriyakul, "A Preliminary Investigation of a Linguistic Perceptron", *Lecture Notes in Artificial Intelligence*, vol. 3809, Springer, December 2005, pp. 1180-1186.

[19] S. Auephanwiriyakul and S. Dhompongsa, "An Investigation of Linguistic Perceptron in a Nonlinear Decision Boundary Problem", *2006 IEEE International Conference on Fuzzy Systems*, Vancouver, BC., Canada, July 2006, pp. 6590-6596.

[20] G. Klir, and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Application*, New Jersey, Prentice Hall, 1995.

[21] M. Mizumoto and K. Tanaka, "Some Properties of Fuzzy Numbers in Advances in Fuzzy Sets Theory and Applications", In: Gupta, M. M. eds., *Advances in Fuzzy Set Theory and Applications*, Amsterdam, North-Holland, 1979, 153-164.

[22] O. Kaleva and S. Seikkala, "On Fuzzy Metric Spaces", *Fuzzy Sets and Systems*, vol. 12, 1984, 215-229.

[23] L. Zadeh, "Outline of New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. Sys. Man Cyb.*, vol. 3, no. 1, 1973.

[24] W. Dong, H. Shah, and F. Wong, "Fuzzy Computations in Risk and Decision Analysis", *Civ. Eng Syst.*, vol. 2, 1985, 201-208.

[25] W. Dong, and F. Wong, "Fuzzy Weighted Averages and Implementation of the Extension Principle", *Fuzzy Sets and Systems*, vol. 21, 1987, 183-199.

[26] R. Moore, *Interval Analysis*, New Jersey, Prentice-Hall, 1966.

[27] M. Mares, *Computation over Fuzzy Quantities*, CRC Press, 1994.

[28] I. Kramosil, and J. Michalek, "Fuzzy Metric and Statistical Metric Spaces", *Kybernatika*, vol. 11, 1975, 336-344.

[29] F. Rosenblatt, *The Perceptron: A Perceiving and Recognizing Automaton*, Cornell University, Ithaca, NY, Project PARA, Cornell Aeronaut Laboratory, Rep., 85-460.1, 1957.

[30] T. Tou and R. C. Gonzalez, *Pattern Recognition Principles,* Reading, MA: Addison-Wesley, 1974.

[31] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New Jersey, Prentice Hall 1999.

[32] J. N. Mordeson and P. S. Nair, *Fuzzy Mathematics: An Introduction for Engineers and Scientists*, New York, Physica-Verlag, 2001.

[33] P. H. A. Sneath, "The Application of Computers in Taxonomy", *J. General Microbiology*, vol. 17, 1957, 201-226.

[34] S. I. Gallant, "Perceptron-Based Learning Algorithms", *IEEE Trans. on Neural Networks*, vol. 1, no. 2, 1990, 179-191.

[35] The UCI Repository of Machine Learning Databases and Domain Theories. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[36] A. Budihardjo, J. Grzymala-Busse, L. Woolery, "Program LERS_LB 2.5 as a Tool for Knowledge Acquisition in Nursing", *Proceedings of the 4th Int. Conf. On Industrial & Engineering Applications of AI & Expert Systems*, 1991, pp. 735-740.

[37] L. Woolery, J. Grzymala-Busse, S. Summers, A. Budihardjo, "The Use of Machine Learning Program LERS_LB 2.5 in Knowledge Acquisition for Expert System Development in Nursing", *Computers in Nursing*, vol. 9 1991, 227-234.