PERTEMUAN 18

SOURCE CODE FMADM

A. Tujuan Pembelajaran

Setelah mempelajari materi dalam pertemuan 18, mahasiswa mampu menjelaskan penggunaan metode FMADM dengan menggunakan metode SAW, metode WP dan metode TOPSIS dalam pengambilan keputusan..

B. Uraian Materi

1. Fuzzy Neural Network menggunakan kode dan animasi

Selalu ada ketidakpastian dalam eksperimen dunia nyata. Dan dengan demikian saat memodelkan sistem kami, kami perlu memperhitungkan ketidakpastian ini. Anda sudah familiar dengan salah satu bentuk ketidakpastian yang menjadi dasar dari Teori Probabilitas. Analog dengan teori probabilitas, cara berbeda untuk bekerja dengan ketidakpastian dikembangkan oleh Zadeh yang dikenal sebagai Fuzzy Set.

a. Set Fuzzy vs Set Renyah

Sebelum kita membahas tentang himpunan fuzzy dan operasinya, lebih baik untuk memahami himpunan fuzzy dibandingkan dengan himpunan tajam yang sudah Anda kenal.

Kami mendefinisikan himpunan sebagai kumpulan elemen - misalnya A = {x | x ∈ Rand x> 0} di mana sebuah semesta umum dari wacana ditetapkan (dalam contoh kita, R atau himpunan bilangan Riil) dan setiap titik berbeda di alam semesta itu merupakan bagian dari set kami atau tidak.

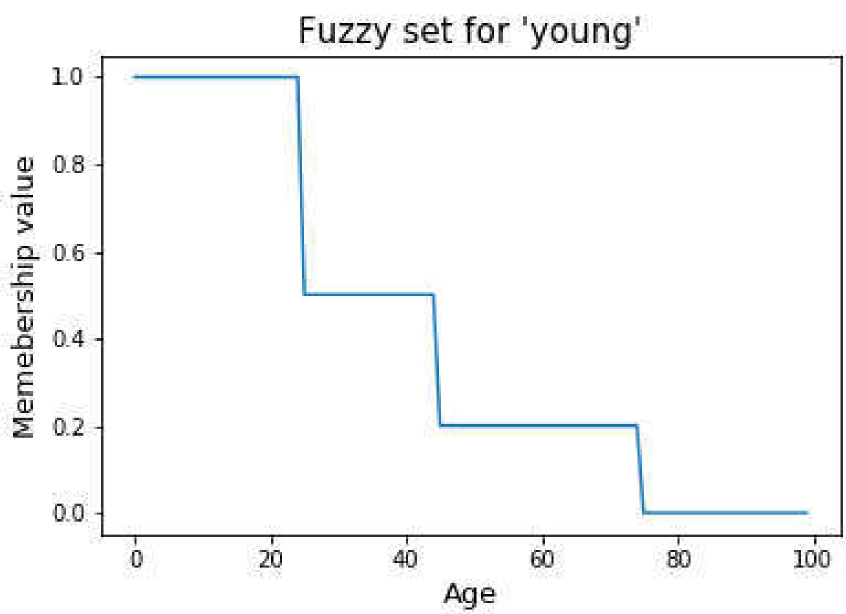
Apa yang kami jelaskan di atas adalah teori himpunan yang sudah Anda kenal. Himpunan fuzzy juga mirip dengan himpunan yang tajam tetapi lebih umum. Jadi himpunan crisp dapat dikatakan sebagai jenis himpunan fuzzy tertentu.

Dalam himpunan tajam, elemen yang termasuk dalam semesta wacana kita adalah bagian dari himpunan atau tidak, yaitu 0 atau 1, tetapi bagaimana dengan nilai-nilai di antaranya? Bisakah elemen hanya sebagian menjadi bagian dari himpunan?

Mari kita definisikan nilai 0 dan 1 sebagai nilai keanggotaan (mA (x)) untuk set crisp. Ini adalah nilai biner yaitu hanya bisa 0 atau 1 untuk set yang tajam.

Untuk himpunan fuzzy, nilai keanggotaan ini adalah nilai kontinu antara [0, 1]. Dengan cara ini himpunan fuzzy berbeda dari himpunan yang tajam. Dalam himpunan fuzzy, partisipasi elemen dalam himpunan tidak didefinisikan secara ketat tetapi dapat berupa nilai nyata antara [0, 1]. Tapi apa artinya ini secara intuitif?

Himpunan fuzzy memungkinkan kita untuk secara matematis mendefinisikan istilah yang ambigu dan subjektif seperti 'muda', 'banyak', 'sedikit'. Kita dapat mendefinisikan nilai tertentu di alam semesta wacana kita sebagai bagian dari rangkaian itu dengan partisipasi parsial. Mari kita ambil contoh untuk lebih memperjelas - Misalkan kita mendefinisikan himpunan fuzzy sebagai himpunan semua orang 'muda'. Kemudian untuk semua orang usia [0, 100] kita dapat mendefinisikan orang usia x untuk memiliki nilai keanggotaan mA (x) yang menunjukkan partisipasi mereka dalam himpunan.



Gambar 22. Grafik Himpunan Fuzzy

Tentu saja fungsi keanggotaan mA (x) akan berbeda untuk orang lain. Grafik di atas menunjukkan interpretasi saya atas kata 'muda' dan artinya. Jadi himpunan fuzzy didefinisikan sebagai pasangan terurut - A = {x, mA (x)}. Saya juga akan menyertakan definisi standar lengkap untuk ukuran yang baik.

Misalkan X adalah ruang titik (objek), dengan elemen umum X dilambangkan dengan x. [X sering disebut sebagai alam semesta wacana.] Himpunan fuzzy (kelas) A dalam X ditandai dengan keanggotaan (karakteristik) fungsi m ~ (x) yang mengasosiasikan dengan setiap titik di X bilangan real dalam interval [0, 1], dengan nilai m ~ (x) mewakili “tingkat keanggotaan” dari x dalam A. Dengan demikian, semakin dekat nilai ~ A (z) dengan persatuan, semakin tinggi tingkat keanggotaan z dalam SEBUAH.

b. Operasi himpunan fuzzy

Dengan asumsi X sebagai semesta wacana, A $\in$ X dan B $\in$ X, operasi himpunan fuzzy didefinisikan sebagai berikut:

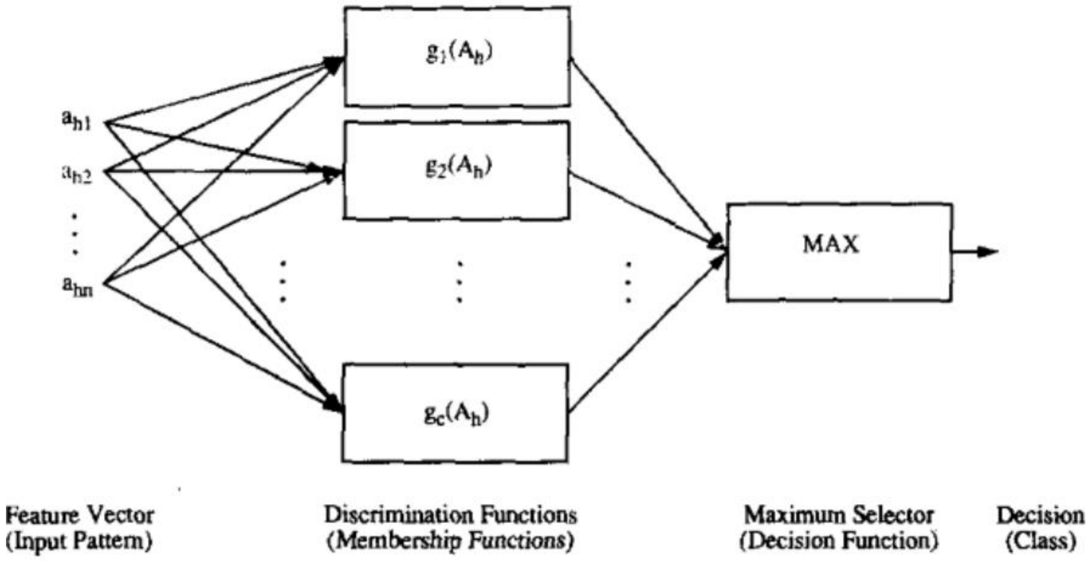
1) Perbandingan (Apakah A = B?): Iff mA (x) = mB (x) untuk semua x di X
2) Containment (Is A $\subset$ B?): Iff mA (x) <mB (x) untuk semua x dalam X
3) Union: m (A $\vee$B) (x) = max (mA (x), mB (x)) untuk semua x dalam X
4) Persimpangan: m (A $\wedge$ B) (x) = min (mA (x), mB (x)) untuk semua x di X

Operasi dasar ini cukup jelas dari definisinya. Jadi kita akan beralih ke inti postingan – pengklasifikasi.

2. Pengklasifikasi Fuzzy Maksimum

Kita akan melihat pengklasifikasi ini dengan dua cara berbeda. Cara pengklasifikasi ini digunakan untuk menyimpulkan kelas dari pola pengujian dan cara jaringan saraf pengklasifikasi ini dilatih, yaitu inferensi dan algoritme pembelajaran.

Inferensi:

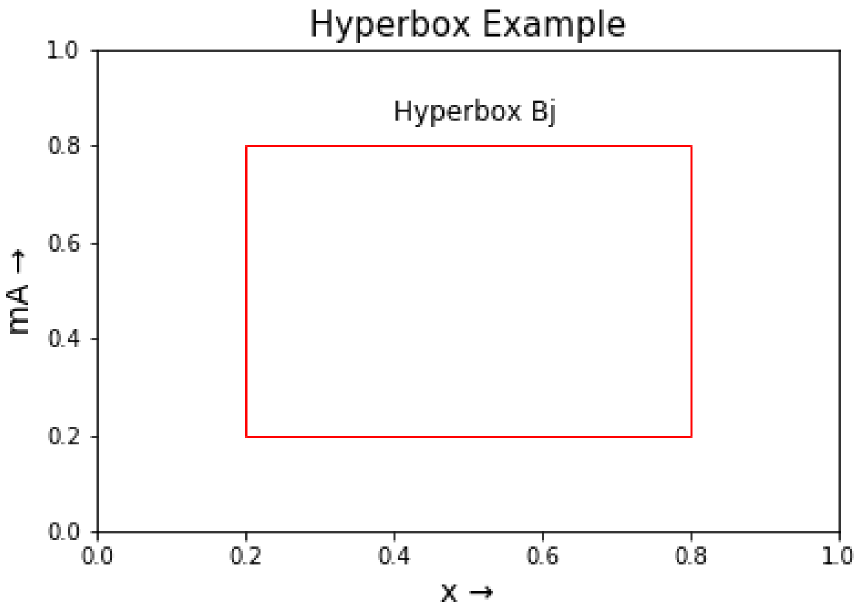


Pertimbangkan bahwa kita memiliki pola dimensi-n Ah. Dan kami memiliki sejumlah K fungsi diskriminan di mana setiap fungsi diskriminan memberitahu kelas pola dengan skor kepercayaan antara [0, 1]. Kami percaya diskriminan paling percaya diri dalam inferensi kami yaitu fungsi memberikan nilai maksimum, kami akan mempertimbangkan pola milik kelas itu.

Sekarang ganti saja fungsi diskriminan dengan fungsi keanggotaan dan Anda akan segera memahami bagaimana himpunan fuzzy dapat digunakan untuk inferensi kelas. Biarkan fungsi diskriminan K menjadi K fuzzy set masing-masing mendefinisikan partisipasi pola dalam kelas tertentu. Dengan demikian kita akan percaya bahwa pola tersebut termasuk dalam kelas yang ditentukan oleh himpunan fuzzy dimana pola tersebut memberikan nilai keanggotaan maksimum.

Hyperbox:

Sekarang sebelum kita memasukkan kerangka kerja inferensi ini ke dalam jaringan saraf, kita perlu memahami satu representasi alternatif untuk himpunan fuzzy. Pertimbangkan semesta wacana 2D [0, 1]. Kami mendefinisikan persegi panjang sebagai himpunan fuzzy sehingga semua titik di dalam kotak itu memiliki nilai keanggotaan 1.



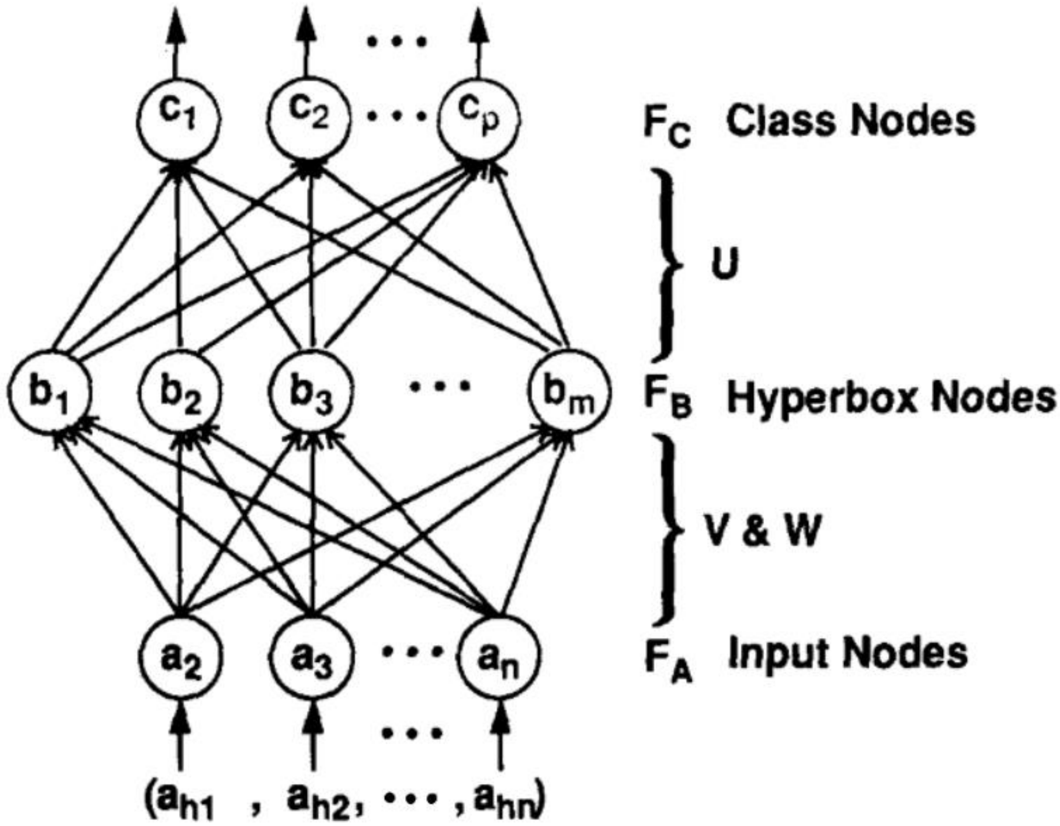
Gambar 23. Hyperbox Example

Sebuah kotak ditentukan oleh titik maksimum dan titik minimumnya. Kotak yang ditunjukkan pada grafik di atas ditentukan oleh min-pt V = [0.2, 0.2] dan max-pt W = [0.8, 0.8]. Kotak ini juga merupakan set fuzzy. Dan nilai keanggotaan himpunan fuzzy ini didefinisikan sedemikian rupa sehingga semua titik di dalam kotak tersebut memiliki mA (x) = 1.

Tapi bagaimana dengan nilai keanggotaan untuk poin di luar kotak? Nilai keanggotaan setiap titik di luar kotak berkurang (<1) seiring bertambahnya jarak antara kotak itu dan titik tersebut. Jarak ini dirata-ratakan untuk semua dimensi (dalam hal ini hanya 2). Nilai keanggotaan untuk definisi himpunan fuzzy kotak seperti itu diberikan oleh-

$$b_j(A_h) = \frac{1}{2n}\sum_{i=1}^{n}[\max(0, 1 - \max(0, \gamma \min(1, a_{hi} - w_{ji}))) + \max(0, 1 - \max(0, \gamma \min(1, v_{ji} - a_{hi})))], \quad (1)$$

Di sini Ah adalah pola dimensi-n kita (dalam contoh kita 2D) dan kotak Bj juga ditentukan dalam dimensi-n (yaitu 2 dalam kasus kita). Persamaan di atas cukup jelas kecuali untuk γ yang merupakan hyperparameter yang dikenal sebagai sensitivitas atau laju penurunan yang dapat digunakan untuk mengontrol nilai keanggotaan



Gambar 24. Hyperbox

Hyperbox:

Dalam arsitektur jaringan neural yang ditunjukkan di atas, node input adalah placeholder untuk pola input berdimensi-n. Ada n node masukan untuk ruang masukan n-dimensi. Lapisan berikutnya berisi node Hyperbox. Setiap node hyperbox ditentukan oleh min-pt dan max-pt dalam ruang n-dimensi. Dan setiap hyperbox milik beberapa kelas.

Lapisan terakhir berisi node kelas. Jumlah node di lapisan ini sama dengan jumlah kelas yang memungkinkan. Pada bagian inferensi dari pengklasifikasi kita menggunakan keluaran dari simpul kelas ini sebagai fungsi diskriminan dan simpul kelas yang memberikan skor maksimum diasumsikan untuk memprediksi nilai kelas dari pola tersebut.

Salah satu detail penting yang perlu dipahami adalah interaksi antara layer node kelas dan layer hyperbox. Saya mengatakan bahwa setiap hyperbox (node) milik beberapa kelas. Saat mempertimbangkan keluaran node kelas, kami mempertimbangkan nilai keanggotaan maksimum di antara hyperbox milik kelas itu. Nilai keanggotaan yang diberikan oleh hyperbox milik kelas lain tidak dipertimbangkan.

E. g Misalkan hyperbox Bj dan Bk milik kelas 1 dan hyperbox Bl milik kelas 2 (kami hanya memiliki tiga hyperbox dalam pengklasifikasi kami). Dan kami memiliki dua node kelas C1 dan C2. Untuk pola Ah hyperbox memberikan nilai keanggotaan masing-masing sebagai [0.4, 0.8, 0.7].

Untuk menghitung output dari node kelas C1 - C1 = max ([0.4, 0.8, 0.0]) = 0.8. Kami menganggap bahwa hubungan antara node Bl dan C1 mewakili 0 dan link antara Bj dan Bk mewakili 1. Oleh karena itu kontribusinya dalam mendefinisikan nilai node kelas.

Demikian pula untuk node kelas C2 - C2 = max ([0.0, 0.0, 0.7]) = 0.7. Sekarang kita memiliki [C1, C2] = [0.8, 0.7]. Untuk memprediksi kelas, kita ambil kelas dengan skor maksimal. Oleh karena itu pola Ah termasuk dalam kelas 1.

3. Algoritma Pembelajaran

Sebuah pertanyaan yang sangat wajar mungkin muncul di benak Anda setelah melalui bagian di atas. Bagaimana cara kami mendapatkan hyperbox ini? Setelah kita memiliki hyperbox, kita dapat menggunakan metode yang dijelaskan di atas untuk memprediksi kelas pola. Tetapi pertama-tama kita memerlukan algoritme untuk mempelajari hyperbox ini berdasarkan data yang kami miliki.

Algoritma pembelajaran juga cukup mudah. Saya akan merekomendasikan agar Anda juga mempelajari algoritme pembelajaran yang dijelaskan dalam makalah asli untuk lebih memperkuat pemahaman Anda.

Ingatlah bahwa hyperbox didefinisikan menggunakan dua poin - min-pt dan max-pt (dengan notasi Vj dan Wj untuk hyperbox ke-j). Ketika kami memulai pelatihan kami, kami tidak memiliki hyperbox yang dipelajari, yaitu 0 node hyperbox di jaringan kami. Saat kami mengamati pola dalam set pelatihan satu per satu, kami akan mulai membuat node ini. Algoritme pembelajaran dapat dikategorikan menjadi dua fase berbeda - fase Ekspansi dan fase Kontraksi.

Sebuah. Fase Ekspansi:

Misalkan Xh adalah pola pelatihan n dimensi milik kelas Y. Hal pertama yang kita lakukan adalah menemukan hyperbox yang paling cocok untuk ekspansi. Kami mengambil semua hyperbox milik kelas Y dan menghitung nilai keanggotaannya. Hyperbox dengan nilai keanggotaan maksimum adalah kandidat paling cocok untuk perluasan.

Misalkan Bj menjadi hyperbox yang paling cocok untuk ekspansi. Sebelum kami berkembang, kami harus memenuhi kriteria perluasan yang merupakan kondisi yang ditentukan oleh

$$n\theta \geq \sum_{i=1}^{n} (\max(w_{ji}, x_{hi}) - \min(v_{ji}, x_{hi})).$$

di mana θ adalah hyperparameter yang dikenal sebagai batas ekspansi. Hyperparameter ini mengontrol perluasan maksimum yang diizinkan untuk hyperbox. Jika hyperbox kami memenuhi kondisi di atas maka hyperbox tersebut akan diperluas sebagai berikut-

$$v_{ji}^{\mathrm{new}} = \min(v_{ji}^{\mathrm{old}}, x_{hi}) \qquad \forall\, i = 1, 2, \cdots, n,$$

$$w_{ji}^{\mathrm{new}} = \max(w_{ji}^{\mathrm{old}}, x_{hi}) \qquad \forall\, i = 1, 2, \cdots, n.$$

di mana Vj dan Wj adalah min-pt dan max-pt baru dari hyperbox Bj. Secara intuitif, hyperbox mengembang sedemikian rupa sehingga titik pelatihan Xh disertakan dalam regionnya.

Tetapi apa yang terjadi jika hyperbox Bj tidak memenuhi kriteria perluasan? Dalam hal ini, kami pindah ke hyperbox paling sesuai berikutnya yang ditentukan oleh urutan menurun dari nilai keanggotaan dan mencentang kriteria perluasan untuk kotak itu. Dan kami melakukannya sampai kami menemukan kotak yang cocok atau kami kehabisan kotak. Dalam kasus terakhir di mana kami tidak dapat menemukan hyperbox yang cocok untuk ekspansi, kami membuat hyperbox baru untuk kelas Y dengan Vj (min-pt) = Wj (max-pt) = Xh.

Tapi kita belum selesai, bagaimana jika dua hyperbox saling tumpang tindih? Jika keduanya termasuk dalam kelas yang sama maka pengklasifikasi akan memprediksi kelas dengan benar, tetapi jika dua hyperbox dari kelas yang berbeda tumpang tindih, maka pengklasifikasi akan memprediksi bahwa sebuah pola dimiliki oleh kedua kelas tersebut. Kami pasti ingin menghindari situasi ini. Mari kita lihat apa yang bisa dilakukan untuk menghindari tumpang tindih semacam ini.

Sebuah Fase Kontraksi:

Pertama-tama saya akan menjelaskan secara singkat fase tersebut dan kemudian memberikan kondisi sebenarnya yang menentukan operasi yang perlu kami lakukan di hyperbox kami.

Misalkan dari tahap terakhir kami menemukan hyperbox yang sesuai dengan kriteria perluasan dan kami memperluas kotak Bj. Kita perlu memastikan bahwa kotak yang diperluas ini tidak tumpang tindih dengan hyperbox lain milik kelas yang berbeda. Tumpang tindih antara hyperbox dari kelas yang sama diperbolehkan, jadi kami hanya akan memeriksa tumpang tindih antara kotak yang diperluas dan hyperbox dari kelas yang berbeda.

Saat memeriksa tumpang tindih antara hyperbox yang diperluas (Vj, Wj) dan kotak uji (Vk, Wk), kami mengukur tumpang tindih di antara mereka di semua dimensi-n yaitu kami memeriksa tumpang tindih di setiap dimensi dan pada saat yang sama kami juga catat nilai tumpang tindih itu (berapa banyak tumpang tindih yang ada?). Dengan demikian kita dapat menemukan dimensi di antara dimensi-n di mana tumpang tindih minimum. Kami mencatat dimensi ini dan nilai yang tumpang tindih ini. Jika kami menemukan bahwa kotak tidak tumpang tindih dalam dimensi apa pun, maka kami mengabaikan kotak ini dan pindah ke kotak pengujian berikutnya.

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$,

$$\delta^{\text{new}} = \min(w_{ji} - v_{ki}, \delta^{\text{old}}).$$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$,

$$\delta^{\text{new}} = \min(w_{ki} - v_{ji}, \delta^{\text{old}}).$$

Case 3: $v_{ji} < v_{ki} < w_{ki} < w_{ji}$,

$$\delta^{\text{new}} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{\text{old}}).$$

Case 4: $v_{ki} < v_{ji} < w_{ji} < w_{ki}$,

$$\delta^{\text{new}} = \min(\min(w_{ji} - v_{ki}, w_{ki} - v_{ji}), \delta^{\text{old}}).$$

Kami menghitung δnilai baru untuk setiap dimensi sesuai dengan kasus di atas (yang mencakup semua kemungkinan tumpang tindih). Dari nilai tumpang tindih ini, kami menemukan dimensi Δ di mana tumpang tindih ini (δ baru) adalah minimum. Dalam makalah aslinya, mekanisme ini dijelaskan dalam konteks penerapannya dalam suatu program. Tetapi selama Anda memahami apa yang terjadi, Anda dapat menerapkan logika tersebut secara berbeda untuk mendapatkan hasil yang sama.

Sekarang kita memiliki dimensi di mana terdapat tumpang tindih minimum (Δ) antara dua kotak. Tujuan kami adalah untuk mengontrak kotak Bj yang diperluas tetapi kami ingin kontraksi sekecil mungkin sambil tetap menghilangkan tumpang tindih. Jadi kita akan menggunakan dimensi dengan tumpang tindih minimum (Δ) dan mengontrak kotak kita hanya dalam dimensi itu.

Case 1: $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$,

$$w_{j\Delta}^{\text{new}} = v_{k\Delta}^{\text{new}} = \frac{w_{j\Delta}^{\text{old}} + v_{k\Delta}^{\text{old}}}{2}.$$

Case 2: $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$,

$$w_{k\Delta}^{\text{new}} = v_{j\Delta}^{\text{new}} = \frac{w_{k\Delta}^{\text{old}} + v_{j\Delta}^{\text{old}}}{2}.$$

Case 3a: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$,

$$v_{j\Delta}^{\text{new}} = w_{k\Delta}^{\text{old}}.$$

Case 3b: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$,

$$w_{j\Delta}^{\text{new}} = v_{k\Delta}^{\text{old}}.$$

Case 4a: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$,

$$w_{k\Delta}^{\text{new}} = v_{j\Delta}^{\text{old}}.$$

Case 4b: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$,

$$v_{k\Delta}^{\text{new}} = w_{j\Delta}^{\text{old}}.$$

Berdasarkan jenis tumpang tindih dalam dimensi ke-, kami mengontrak kotak Bj dalam dimensi tersebut menggunakan kasus-kasus yang diberikan di atas. Jadi kami telah memperluas hyperbox dan mengujinya untuk tumpang tindih dengan hyperbox kelas yang berbeda. Dan jika terjadi tumpang tindih, kami telah mencari cara untuk mengecilkan kotak tersebut secara minimal dan menghilangkan tumpang tindih tersebut.

4. Source Code Hyperbox

Kami akan melihat contoh mainan untuk memastikan bahwa kami memahami dengan jelas apa yang terjadi.

Misalkan kita memiliki 4 pola dalam set pelatihan kita sebagai berikut -pola:

A1 = [0,2, 0,2]

A2 = [0,6, 0,6]

A3 = [0,5, 0,5]

A4 = [0,4, 0,4]

kelas:

d1 = 1

d2 = 2

d3 = 1

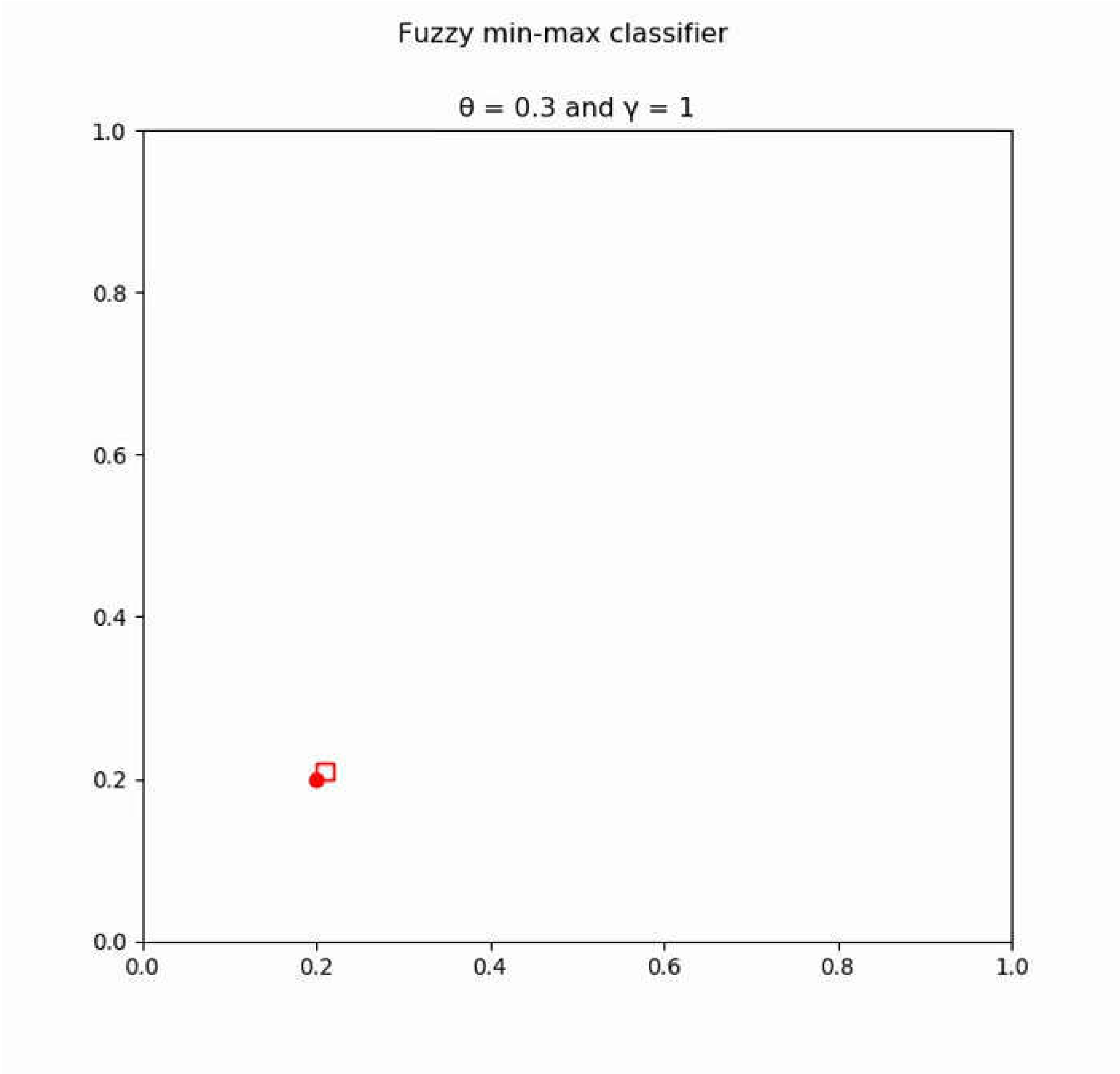
d4 = 1

Kami akan mengambil pola satu per satu dan memeriksa hasil dari algoritme pembelajaran secara visual.

Pola: [0.2, 0.2] Kelas: 1

Tidak ditemukan hyperbox yang cocok untuk perluasan.

Tambahkan hyperbox baru kelas 1 dengan Vj = Wj = [0,2, 0,2]

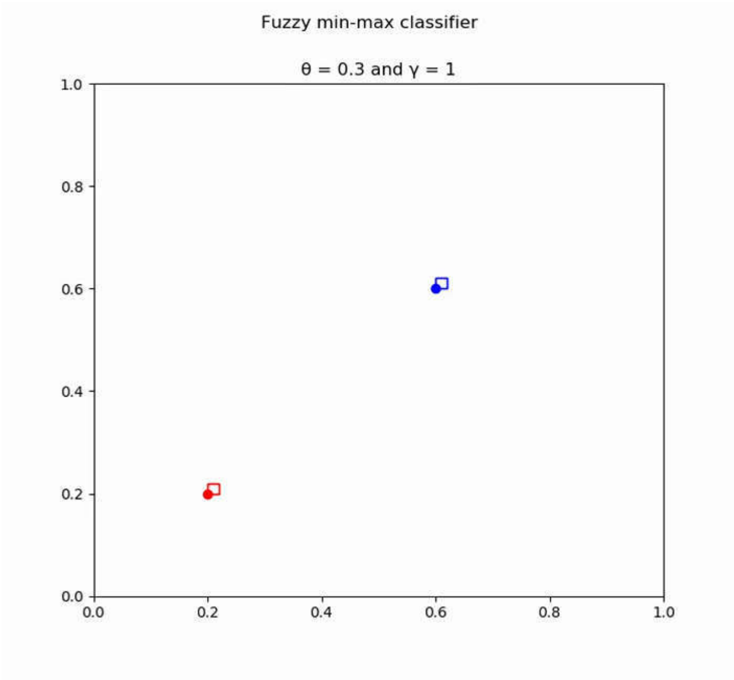


Gambar 25. Fuzzy Classifier

Pola: [0.6, 0.6] Kelas: 2

Tidak ditemukan hyperbox yang cocok untuk perluasan.

Tambahkan hyperbox baru kelas 2 dengan Vj = Wj = [0,6, 0,6]



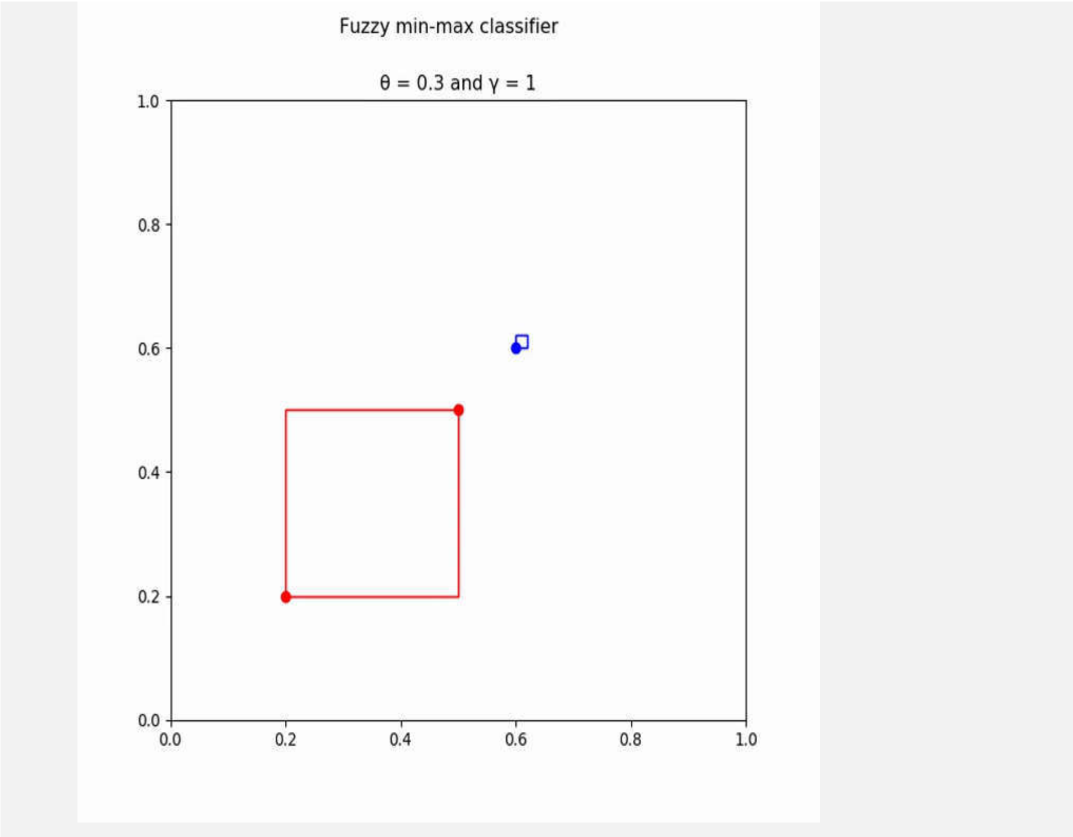
Pola: [0,5, 0,5] Kelas: 1

Hyperbox paling cocok: Vj = [0,2, 0,2] Wj = [0,2, 0,2]

Kriteria perluasan terpenuhi.

Hyperbox diperluas. Hyperbox baru: Vj = [0,2, 0,2] Wj = [0,5, 0,5]

Tidak ditemukan tumpang tindih.



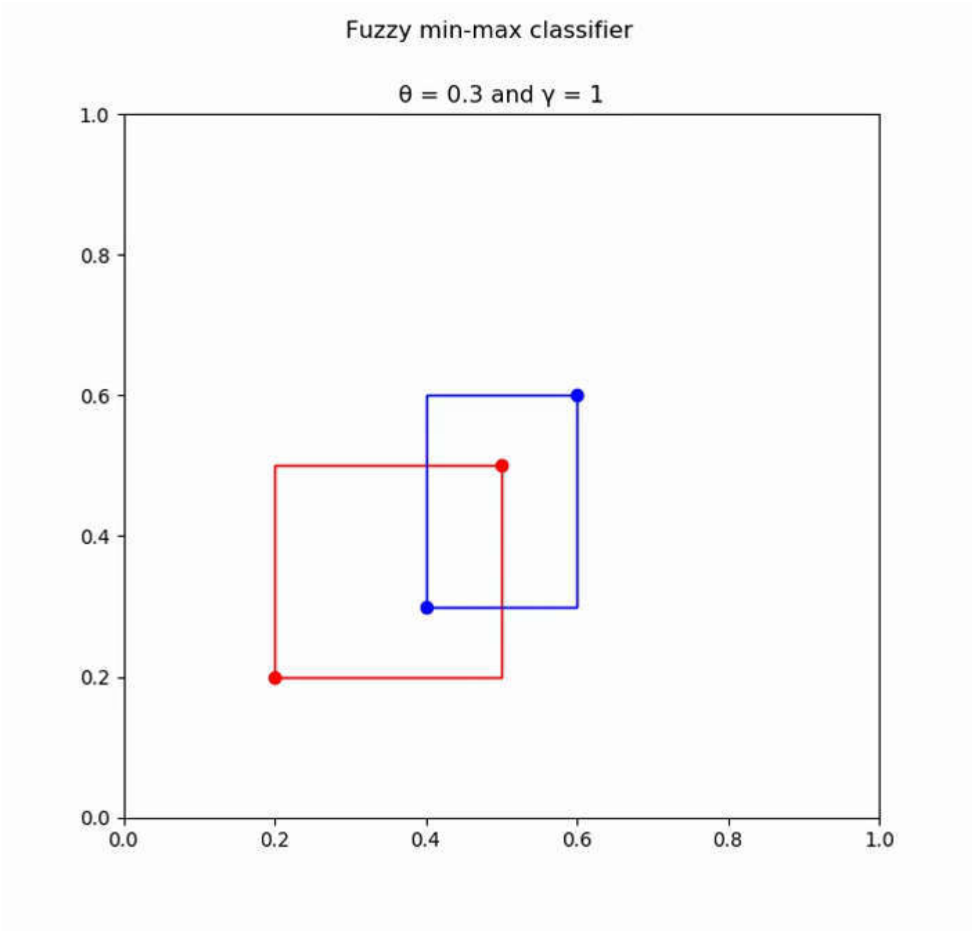
Pola: [0.4, 0.3] Kelas: 2

Hyperbox paling cocok: Vj = [0.6, 0.6] Wj = [0.6, 0.6]

Kriteria perluasan terpenuhi.

Hyperbox diperluas. Hyperbox baru: Vj = [0,4, 0,3] Wj = [0,6, 0,6]

Tumpang tindih dengan hyperbox kelas 1 ditemukan.



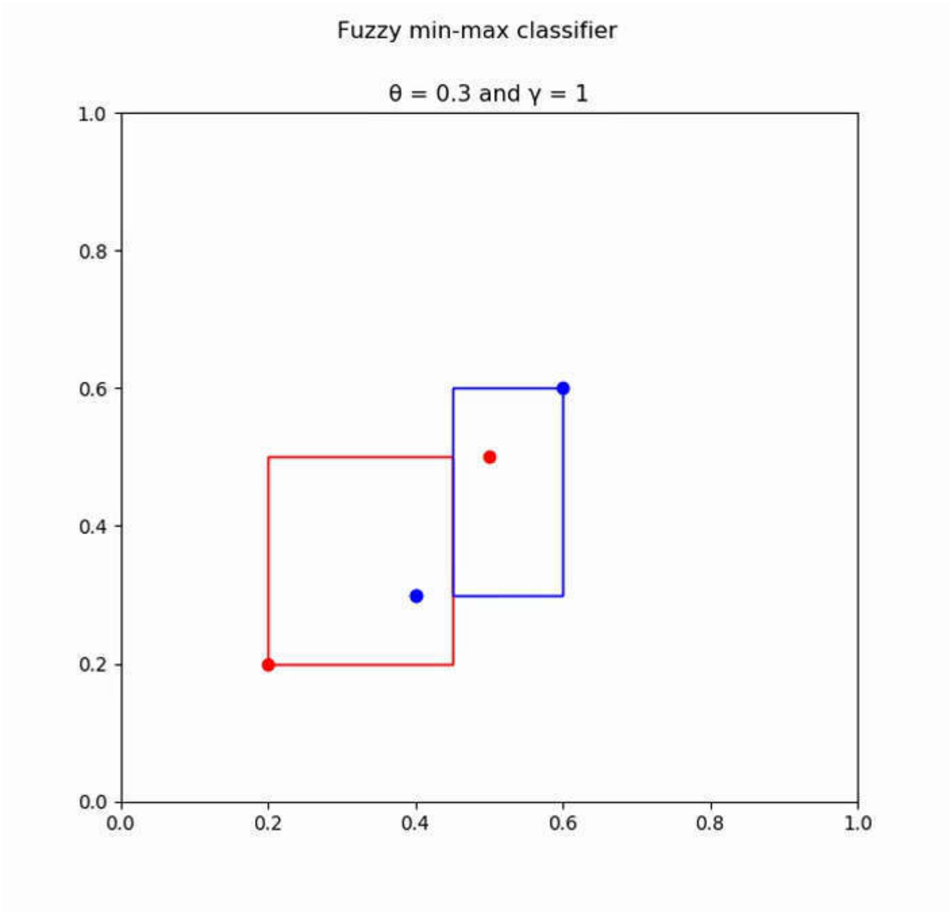
Kotak yang diperluas: Vj = [0.4, 0.3] Wj = [0.6, 0.6]

Kotak uji: Vk = [0,2, 0,2] Wk = [0,5, 0,5]

Tumpang tindih minimum ditemukan dalam dimensi: Δ = 0 (yaitu dimensi horizontal)

Kedua hyperbox disesuaikan dalam dimensi itu untuk menghilangkan tumpang tindih.

Tumpang tindih dihapus.



Jadi kami menerapkan algoritma pembelajaran dan mendapatkan hyperbox berdasarkan data pelatihan.

Catatan: Pengamat di antara Anda akan memperhatikan bahwa jika kita menggunakan hyperbox yang dipelajari pada contoh di atas dan menggunakannya untuk memprediksi poin dalam set pelatihan, maka pengklasifikasi akan memberikan prediksi yang salah. Benar, pengklasifikasi di atas tidak memiliki akurasi 100% (yang ini hanya memiliki 50%). Apa yang dapat dilakukan untuk meningkatkan kinerja?

Kami memiliki hyperparameter θ yang dapat digunakan untuk bermain-main untuk meningkatkan hasil. Saat Anda menggunakan nilai tinggi untuk θ maka ukuran maksimum yang diizinkan dari hyperbox meningkat dan ini meningkatkan kemungkinan pola label yang salah jatuh di wilayah itu.

Tetapi jika kita mengambil nilai terlalu kecil untuk θ maka jumlah hyperbox akan meningkat dan melebihi poin yang baru ditambahkan. Dengan cara ini pengklasifikasi kami kehilangan kemampuan untuk menggeneralisasi dengan baik dan memberikan performa buruk atas poin di luar set pelatihannya.

Anda perlu bereksperimen dengan nilai θ yang berbeda untuk menemukan nilai yang memberikan kinerja terbaik. Saya akan menambahkan beberapa contoh dengan nilai θ yang berbeda dan Anda dapat mengamati sendiri bagaimana θ memengaruhi pembuatan hyperbox.

C. Soal Latihan/ Tugas

1. Apa yang dimaksud logika Fuzzy dengan metode SAW?
2. Apa yang dimaksud logika Fuzzy dengan metode WP?
3. Apa yang dimaksud dengan Fuzzy Multiple Attribute Decision Making (**FMADM**)?
4. Sebutkan algoritma penyelesaian pada metode F-SAW berserta dengan rumusnya?

D. Daftar Pustaka

Andreas Meier, Edy Portmann, Kilian Stoffel, Luis Terán (2017). Penerapan Fuzzy Logic untuk Proses Pengambilan Keputusan Manajerial Penelitian dan Studi Kasus Terkini. New York: Springer.

R. Venkata Rao (2007). Pengambilan Keputusan di Lingkungan Manufaktur Menggunakan Teori Grafik dan Metode Pengambilan Keputusan Beberapa Atribut Fuzzy.

Cengiz Kahraman (2008). Pengambilan Keputusan Multi-Kriteria Fuzzy: Teori dan Aplikasi dengan Perkembangan Terbaru. New York: Springer.

Ding, T., Liang, L., Yang, M., & Wu, H. (2016). Pengambilan Keputusan Beberapa Atribut Berdasarkan Evaluasi Silang dengan Parameter Keputusan Yang Tidak Pasti. Masalah Matematika di Teknik, 1-10.

Zimmermann HJ (1991) Teori himpunan fuzzy dan aplikasinya. Kluwer Academic, Boston

Triantaphyllou E (2000) Metode pengambilan keputusan multi kriteria: studi banding. Springer, London

Saaty TL, Tran LT (2007) Tentang ketidakabsahan penilaian numerik fuzzifying dalam proses hierarki analitik. Model Hitung Matematika 46: 962–975

GLOSARIUM

BI Checking adalah informasi Debitur Individual (IDI) Historis yang mencatat lancer atau macetnya pembayaran kredit (Kolektibilitas).

Fuzzy adalah suatu logika yang memiliki nilai kekaburan atau kesamaran antara benar atau salah.

Normalisasi Database adalah proses pengelempokan atribut data yang membentuk entitas sederhana, fleksibel, dan mudah beradaptasi.

Perangkingan adalah jumlah total yang diurutkan dari yang terbesar sampai yang terkecil.