

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a umelej inteligencie

Evolúcia bázy pravidiel fuzzy klasifikátora určeného na klasifikáciu
multispektrálnych satelitných obrazových dát
(Zadanie predmetu Fuzzy rozhodovanie)

1. ročník Ing. štúdia
Umelá inteligencia
Zimný semester 2011/2012

Michal Puheim

Obsah

Obsah.....	1
1 Úvod.....	2
2 Princípy evolúcie bázy pravidiel fuzzy klasifikátora.....	3
2.1 Popis pravidlového systému.....	3
2.2 Inferenčný mechanizmus fuzzy klasifikátora.....	3
2.3 Popis iteratívneho učenia pravidiel.....	4
2.4 Algoritmus tvorby nových pravidiel.....	5
3 Multispektrálne satelitné obrazové dáta.....	7
4 Návrh a implementácia systému.....	8
4.1 Popis ovládacích prvkov programu.....	8
4.2 Parametre inferenčného systému fuzzy klasifikátora.....	9
4.3 Úpravy pôvodného návrhu plynúce z potreby zlepšenia klasifikácie.....	10
5 Experimenty.....	12
5.1 Experiment č. 1.....	12
5.2 Experiment č. 2.....	13
5.3 Experiment č. 3.....	14
5.4 Porovnanie s klasifikátorom založeným na neurónovej sieti.....	15
6 Záver.....	16
7 Literatúra.....	17

1 Úvod

Obsahom zadania je evolúcia bázy pravidiel fuzzy klasifikátora určeného na klasifikáciu multispektrálnych satelitných obrazových dát. Inšpiráciou pri vypracovaní zadania bol článok [1], ktorý popisuje príznaky vyberajúci lingvistický klasifikátor (Feature Selective Linguistic Classifier – FeSLIC), čo je vlastne genetický fuzzy-pravidlovo založený klasifikačný systém (Genetic Fuzzy Rule-Based Classification System – GFRBCS) vytvorený pomocou prístupu iteratívneho učenia pravidiel (Iterative Rule Learning – IRL).

Model klasifikačného systému predstaveného v článku [1] je zostavený z troch procesných štádií. V prvom štádiu sa vytvára počiatočná báza pravidiel podľa princípov IRL, kedy sa postupne jeden za druhým generujú pravidlá pomocou genetického algoritmu. Vytvorená báza pravidiel prechádza v ďalšej fáze procesom zjednodušenia, ktorý má za úlohu znížiť jej komplexnosť. Zjednodušenie je zabezpečené odstránením redundantných pravidiel alebo odfiltrovaním nadbytočných klauzúl v predpokladových častiach pravidiel. Po týchto úpravách je klasifikátor v poslednom, treťom štádiu, optimalizovaný pomocou genetického tuningu.

Po uvážení komplexnosti popísaného systému bol predmet záujmu zúžený na implementáciu prvého štádia systému, ktoré rieši evolúciu bázy pravidiel pomocou genetického algoritmu.

2 Princípy evolúcie bázy pravidiel fuzzy klasifikátora

Klasifikátor je algoritmus, ktorý priradzuje objekty do tried na základe ich popisu. Popis objektu je zvyčajne vo forme vektora príznačkov (atribútov), ktoré sú relevantné pre daný klasifikačný proces. Fuzzy klasifikátor je klasifikátor, ktorého proces klasifikácie nejakým spôsobom zahŕňa fuzzy logiku.

Obvykle sa klasifikátor učí rozpoznávať triedu objektu pomocou učiaceho algoritmu a trénovacej dátovej vzorky. V prípade, že trénovacie dáta nie sú k dispozícii, klasifikátor môže byť navrhnutý priamo na základe znalostí a skúseností architekta klasifikátora, alebo experta. Podobným spôsobom sú veľmi často navrhované bázy pravidiel pri fuzzy klasifikátoroch.

Veľmi zaujímavým spôsobom učenia fuzzy klasifikátora je využitie genetického algoritmu, ktorým sa z počiatočnej bázy pravidiel získavajú nové pravidlá procesom kríženia a mutácie. Nevýhodou oproti pravidlovým systémom vytvorených priamo človekom je zvyčajne nižšia miera čitateľnosti generovaných pravidiel a ich veľký počet. Veľkou výhodou je však zautomatizovanie a tým pádom aj urýchlenie procesu návrhu klasifikátora.

2.1 Popis pravidlového systému

Základom fuzzy-pravidlovo založeného klasifikačného systému je množina fuzzy produkčných pravidiel, ktoré zabezpečujú mapovanie z daného príznačkového priestoru $x = \{x_1, \dots, x_N\} \in F^N$ do priestoru tried $C = \{C_1, \dots, C_M\}$, kde N a M sú rozmery priestoru príznačkov, resp. tried. Systém používa pravidlá v tvare:^[1]

$$R^k: \text{IF } x^1 \text{ is } A_1^k \text{ and } \dots x^N \text{ is } A_N^k \text{ THEN } y \text{ is } C^k \text{ with } r^k, \quad (1)$$

kde $A_i^k = 1, \dots, N$ sú fuzzy množiny definované na i -tej vstupnej premennej a r^k je miera istoty klasifikácie pravidla do triedy C^k , pre vzorku patriacu do fuzzy podpriestoru definovaného predpokladovou časťou pravidla. Pre danú množinu označených vstupných vzoriek $E = \{(x_p, c_p), p = 1, \dots, Q\}$ je miera istoty pravidla vypočítaná podľa vzťahu:^[1]

$$r^k = \sum_{p|c^p=C^k} \mu^k(x^p) / \sum_{p=1}^Q \mu^k(x^p), \quad (2)$$

kde $\mu^k(x^p)$ je miera kompatibility vzorky a predpokladovej časti pravidla, tiež známa ako miera zhody alebo jednoducho odpálenie pravidla.

2.2 Inferenčný mechanizmus fuzzy klasifikátora

Vstupné premenné sú pred klasifikáciou rozdelené do N_L lingvistických termov implementovaných pomocou trojuholníkových funkcií príslušnosti. Avšak v predpokladovej časti pravidla figuruje vždy jeden lingvistický term danej premennej. Je tu zrejmy rozdiel v implementácii oproti systému popísanému v [1], kde je ich možno použiť viac oddelených OR operátormi. Okrem toho, niektoré vstupné premenné môžu byť v predpokladovej časti pravidla úplne absentujúce. Miera zhody k -teho pravidla je potom definovaná ako:^[1]

$$\mu^k(x^p) = \mu_1^{L_q}(x_1^p) \wedge \dots \wedge \mu_N^{L_q}(x_N^p), \quad (3)$$

kde $\mu_i^{L_q}(x_i^p)$ je hodnota funkcie príslušnosti každého lingvistického termu vyskytujúceho sa v predpokladovej časti pravidla k jeho fuzzy množine. Operátor \wedge naznačuje T-normu vybranú k reprezentácii logického operátora AND. AND operátor je implementovaný konvenčne pomocou operátora minima.

Pre každú vzorku x^p je inferenciou v pravidlovom systéme klasifikátora nakoniec určená trieda podľa vzťahu:^[1]

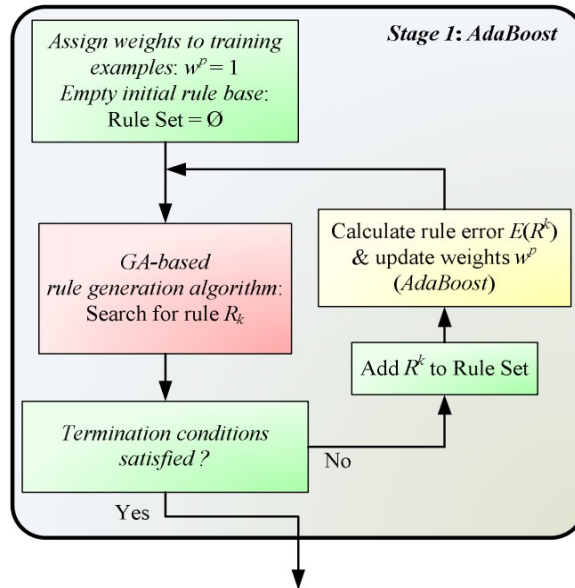
$$C^k = \arg \max_{j=1, \dots, M} \sum_{R_k | C^k = j} \mu^k(x^p) \cdot r^k, \quad (4)$$

využitím takzvanej schémy najväčšieho počtu hlasov.

2.3 Popis iteratívneho učenia pravidiel

Metóda IRL rieši problematiku tvorby bázy pravidiel postupným pridávaním jedného pravidla za druhým pomocou opakovaného spúšťania genetického algoritmu. Vždy keď je vyprodukované nové pravidlo, trénovacie vzorky, ktoré sú ním dostatočne pokryté, sú z trénovacej množiny odstránené, takže novo vytvorené pravidlá sa môžu zamerať na nepokryté časti príznakového priestoru. Táto metóda však môže viesť ku tvorbe konfliktných pravidiel, keďže pravidlá vytvorené v neskorších štádiách behu algoritmu neberú v úvahu predtým odstránené trénovacie vzorky.^[1]

Riešením tohto problému je použitie algoritmu AdaBoost [2], ktorý je vhodný pre tvorbu bázy pravidiel pomocou IRL [3]–[4]. Majme trénovaciu množinu $E_{\text{tm}} = \{e^p = (x_p, c_p), p = 1, \dots, Q\}$. AdaBoost priradí každej z Q trénovacích vzoriek váhu w^p reprezentujúcu relatívnu dôležitosť p -tej trénovacej inštancie. Vždy keď je vyprodukované nové pravidlo, váhy u tých vzoriek, ktoré boli klasifikované správne, sú znížené, čo umožňuje novým pravidlám koncentrovať sa na nesprávne klasifikované vzorky. Keďže však žiadna zo vzoriek nie je z trénovacej množiny odstránená úplne, novo vytvorené pravidlá sú si vedomé predchádzajúco odvodených, čo obmedzuje tvorbu konfliktných pravidiel.^[1]



Obrázok 1: Návrh algoritmu IRL učenia (AdaBoost). Obrázok prevzatý z [1].

Na začiatku učenia sú všetky váhy $w^p = 1$. Vždy keď je vytvorené nové pravidlo, AdaBoost vypočíta jeho chybu, pričom berie v úvahu aktuálne váhy a mieru zhody (3) vzoriek s predpokladovou časťou pravidla.^[1]

$$E(R^k) = \sum_{p | c^p \neq C^k} w^p \mu^k(x^p) / \sum_{p=1}^Q w^p \mu^k(x^p), \quad (5)$$

kde trieda C^k pravidiel je trieda dominujúca tréningovým vzorkám pokrytým predpokladovou časťou pravidla.^[1]

$$C^k = \arg \max_{j=1, \dots, M} \sum_{p|c^p=j} \mu^k(x^p). \quad (6)$$

Nesprávne klasifikované vzorky si zachovávajú pôvodné váhy, zatiaľ čo správne klasifikovaným sú váhy znížené.^[1]

$$w^p \leftarrow \begin{cases} w^p, & \text{ak } c^p = C^k \\ w^p \cdot \beta^k(x^p), & \text{ak } c^p \neq C^k, \end{cases} \quad (7)$$

kde faktor zmeny.^[1]

$$\beta^k(x^p) = (E(R^k) - 1/E(R^k))^{\mu^k(x^p)} \quad (8)$$

je úmerný aktuálnej chybe pravidla a jeho miery zhody s p -tou vzorkou. V dôsledku tejto zmeny sú váhy vzoriek klasifikovaných správne znížené, zatiaľ čo nesprávne klasifikované alebo nepokryté vzorky si zachovávajú pôvodné hodnoty váh, čo umožňuje ďalším generáciám genetického algoritmu sústrediť sa na tie tréningové vzorky, ktoré sa učia ťažšie.^[1]

2.4 Algoritmus tvorby nových pravidiel

Algoritmus tvorby nových pravidiel je vlastne genetický algoritmus, kde každý jedinec v populácii predstavuje jedno fuzzy pravidlo. Po určitom počte generácií je najlepší jedinec vybraný z populácie a vložený do finálnej pravidlovej bázy. Zahnutie váh na tréningových vzorkách vo funkcii vhodnosti (fitness) podporuje kooperáciu medzi pravidlami, čo smeruje genetický algoritmus k produkcii nových pravidiel, ktoré vhodne dopĺňajú bázu už vygenerovaných pravidiel.^[1]

Každé pravidlo v populácii je reprezentované ako chromozóm zložený z dvoch častí – časti premenných a časti lingvistických hodnôt. Za predpokladu existencie N vstupných premenných obsahuje prvá časť reťazec N bitov, kde i -tý bit rozhoduje či bude i -tá premenná aktivovaná, alebo nie. Druhá časť obsahuje informáciu o použitej lingvistickej hodnote každej vstupnej premennej (aktivovanej aj neaktivovanej).^[1]



Obrázok 2: Reprezentácia chromozómu jedinca v populácii. Zelenou farbou je zobrazená časť premenných a červenou časť lingvistických hodnôt.

Genetický algoritmus využíva elitistický model, kde je pravdepodobnosť reprodukcie určená lineárnym hodnotením podľa funkcie vhodnosti a výber jedincov je realizovaný pomocou rulety. Pri reprodukcii je použité jednobodové kríženie ako v časti premenných, tak aj v časti lingvistických hodnôt. V časti premenných je použitá uniformná mutácia, v časti hodnôt tzv, Thriftova mutácia.

Prvotná populácia chromozómov je generovaná podľa tréningovej množiny vzoriek. Pre každého jedinca v populácii je náhodne vybraná tréningová vzorka x^p s pravdepodobnosťou proporcionálnou k váhe w^p . Pre každú vstupnú premennú x_i^p vzorky x^p je vybraná lingvistická hodnota ktorá dosahuje najväčšiu hodnotu funkcie príslušnosti $\mu_i^{L^q}(x_i^p)$ k zodpovedajúcej fuzzy množine a touto lingvistickou hodnotou je následne inicializovaná príslušná časť chromozómu. Časť premenných je inicializovaná náhodne.

Funkcia vhodnosti genetického algoritmu zahŕňa koncept konzistencie a pokrytia (jednotlivých tried). Konzistencia pravidla od neho vyžaduje pokrytie vysokého počtu pozitívnych vzoriek a nízkeho počtu negatívnych vzoriek. Je definovaná ako:^[1]

$$\text{Cons} = \begin{cases} 0, & \text{ak } n^+ < n^- \\ 1 - n^-/n^+, & \text{inak,} \end{cases} \quad (9)$$

kde n^+ a n^- sú fuzzy čísla pozitívnych, resp. negatívnych vzoriek, ktoré berú v úvahu úroveň zhody vzorky a predpokladovej časti pravidla, ako aj váhu vzorky w^p .^[1]

$$n^+ = \sum_{p|c^p = C^k} w^p \mu^k(x^p), \quad (10)$$

$$n^- = \sum_{p|c^p \neq C^k} w^p \mu^k(x^p). \quad (11)$$

Trieda C^k je určená vzťahom (6). Zo vzťahu (9) vyplýva, že nulová hodnota konzistencie je priradená pravidlám, ktoré pokrývajú viac negatívnych vzoriek ako pozitívnych. V opačnom prípade hodnotu určuje lineárna funkcia s oborom hodnôt $<0,1>$.^[1]

Kritérium pokrytia určuje počet vzoriek pokrytých pravidlom v porovnaní s celkovým počtom vzoriek reálne prislúchajúcich k triede pravidla. Je definované ako:^[1]

$$\text{Cov} = n^+ / \sum_{p|c^p = C^k} w^p. \quad (12)$$

Vzťah pre výpočet funkcie vhodnosti má potom tvar:^[1]

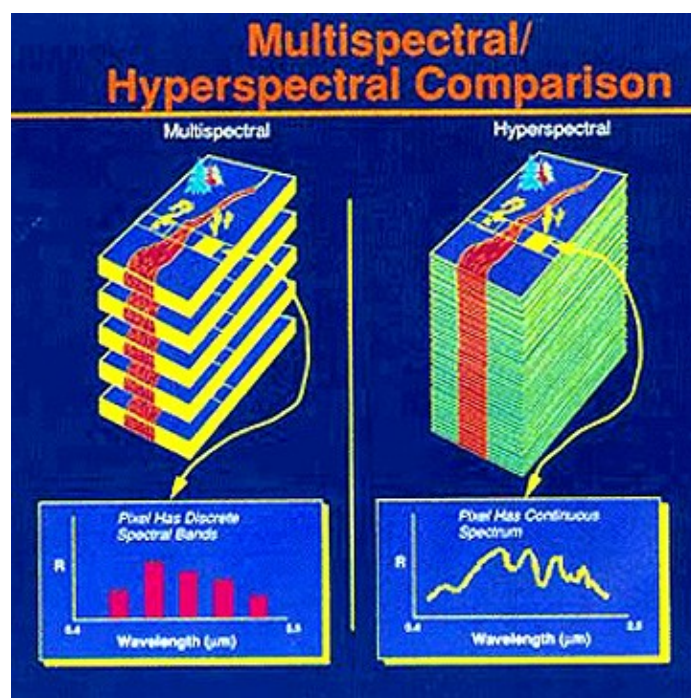
$$f_1 = \text{Cons} \cdot \text{Cov}. \quad (13)$$

Zahrnutie hodnôt váh w^p vo vzťahoch pre výpočet konzistencie a pokrytia efektívne nasmeruje genetický algoritmus k produkcii nových pravidiel, ktoré kooperujú s už odvodenými pravidlami. Algoritmus AdaBoost je tak schopný prekonať problémy nekonzistentnosti, ktoré vyvstávajú v iných klasifikačných systémoch založených na IRL.^[1]

3 Multispektrálne satelitné obrazové dáta

Multispektrálny obraz zachytáva obrazové dáta na špecifických frekvenciách naprieč elektromagnetickým spektrom. Vlnové dĺžky môžu byť oddelené filtermi, alebo použitím nástrojov, ktoré sú citlivé na špecifické vlnové dĺžky, vrátane frekvencií mimo svetelného spektra viditeľného ľudským okom. Spektrálne zobrazovanie umožňuje extrakciu dodatočných informácií, ktoré ľudské oko nedokáže zachytiť so svojimi receptormi na červené, zelené a modré svetlo. Pôvodne bolo vytvorené za účelom mapovania zemského povrchu z vesmíru. Opakom multispektrálneho obrazu je monochromatický obraz.^[5]

Hoci článok [1] využíva pri klasifikácii hyperspektrálne dáta, v rámci implementácie klasifikačného systému boli využité dáta multispektrálne, jednak z dôvodu okamžitej dostupnosti, ako aj kvôli menšiemu celkovému objemu dát.

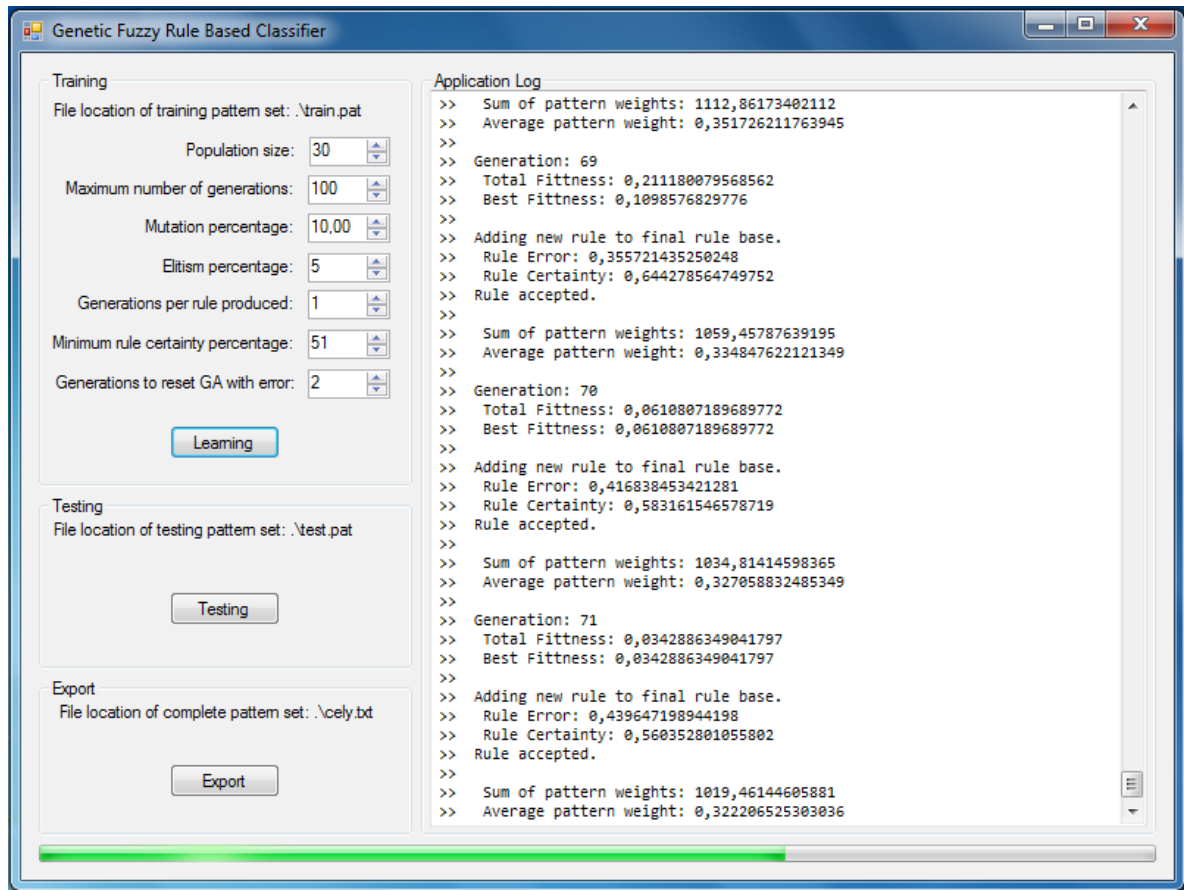


Obrázok 3: Porovnanie multispektrálneho a hyperspektrálneho zobrazovania.

Hyperspektrálne zobrazovanie je však veľmi príbuzné multispektrálnemu zobrazovaniu. Rozdiel medzi nimi je v počte a šírke pokrytých frekvenčných pásem. Multispektrálne zobrazovanie sa zaoberá niekoľkými obrazmi na diskretných a úzkych pásmach. Multispektrálny senzor môže pokrývať mnoho pásem na rôznych častiach spektra od viditeľného svetla, až po infračervené vlnenie. Multispektrálne zobrazovanie neprodukuje spektrum objektu. Hyperspektrálne zobrazovanie sa zaoberá úzkymi frekvenčnými pásmami na spojitom spektrálnom rozsahu a produkuje spektrum všetkých pixelov v danom rozsahu. Senzor môže byť hyperspektrálny ak pokrýva rozsah frekvencií od 500 do 700 nm dvadsiatimi pásmami o šírke 10 nm. Oproti tomu senzor pokrývajúci viacero oddelených pásem (napr. VIS, NIR, SWIR, MWIR, alebo LWIR) by bol považovaný za multispektrálny.^[6]

4 Návrh a implementácia systému

Systém klasifikácie multispektrálnych obrazových dát pomocou evolúcie bázy fuzzy pravidiel je implementovaný pomocou vývojového prostredia Microsoft Visual Studio v programovacom jazyku C#. Ovláda sa cez grafické rozhranie.



Obrázok 4: Ukážka chodu programu počas učenia bázy pravidiel.

4.1 Popis ovládacích prvkov programu

Účelom programu „GeneticFuzzyRuleBasedClassifier.exe” je demonštrácia možností genetického fuzzy klasifikátora. Program pracuje s tromi sadami vzorkovacích dát (trénovacou, testovacou a úplnou), ktoré sú uložené v súboroch „train.pat”, „test.pat” a „cely.txt” v rovnakom priečinku, ako vlastný program.

Hlavné ovládacie prvky programu sú rozdelené rovnako ako súbory pracovných dát. Tlačidlom „Learning” je možné spustiť vlastné učenie bázy pravidiel na trénovacích dátach, tlačidlo „Testing” umožňuje overenie klasifikačnej presnosti vytvorenej bázy pravidiel na testovacích dátach a tlačidlo „Export” zabezpečuje klasifikáciu celej množiny dát s následným uložením výslednej klasifikačnej mapy do súboru vo formáte „ppm” v umiestnení podľa voľby užívateľa. Tento súbor obsahuje obrázok zobrazujúci výslednú klasifikačnú mapu, ktorú možno zobrazit’ pomocou ľubovoľného kompatibilného prehliadača obrázkov (napr. Irfan View).

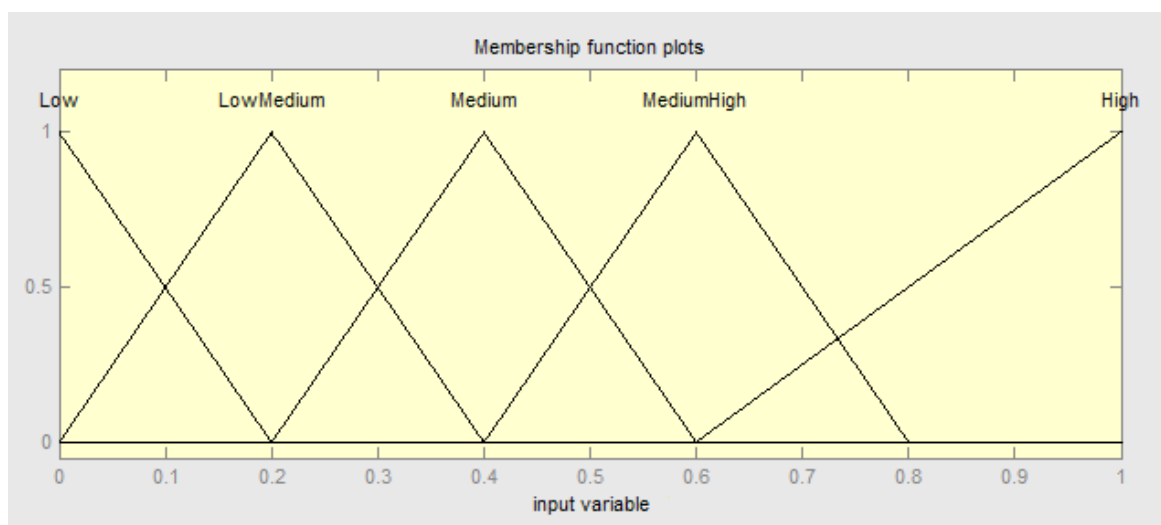
Okrem týchto základných akcií je možné nastavovať aj parametre učenia, medzi ktoré patrí:

- *veľkosť populácie* – určuje počet jedincov vystupujúcich v genetickom algoritme.
- *počet generácií* – počet cyklov genetického algoritmu.
- *percento mutácie* – úroveň mutácie jedincov pri reprodukcii v genetickom algoritme.
- *elitárstvo* – percento jedincov, ktorí sú ponechaní v novej generácii bezo zmeny.
- *počet generácií potrebných pre výber pravidla* – počet generácií genetického algoritmu po ktorých je najlepší jedinec vybraný do bázy pravidiel.
- *minimálna istota pravidla* – umožňuje užívateľovi zadať minimálnu požadovanú istotu, ktorú musí pravidlo splniť, aby bolo vybrané do konečnej bázy pravidiel.
- *počet neúspešných generácií potrebných pre reštart GA* – počet neúspešných výberov pravidla (s nedostatočnou istotou), po ktorých sa genetický algoritmus reinitializuje.

Program tiež obsahuje denník činnosti, ktorý detailne informuje užívateľa o stave a priebehu aktuálneho procesu. Pri učení podáva správy o stave populácie, informácie o pravidlách vybraných do bázy pravidiel, ako aj o stave váh na vzorkách dát z trénovacej množiny. Pri testovaní zobrazuje priebeh testovania a samozrejme jeho výsledky.

4.2 Parametre inferenčného systému fuzzy klasifikátora

Počet vstupných premenných klasifikovaných vzoriek je 6. Z toho vyplýva, že každé fuzzy pravidlo obsahuje predpokladovú časť o maximálnej dĺžke 6 klauzúl. Každá vstupná premenná je fuzzifikovaná do 5 lingvistických termov, ktoré majú trojuholníkové funkcie príslušnosti podľa obrázku 5.



Obrázok 5: Funkcie príslušnosti vstupných premenných.

Inferencia medzi pravidlami prebieha podľa popisu v kapitole 2.2. Klasifikácia vzorky je určená vzťahom (4).

4.3 Úpravy pôvodného návrhu plynúce z potreby zlepšenia klasifikácie

Oproti systému popísanému v kapitole č. 2, boli vykonané niektoré ďalšie úpravy. Vzťah (2) určujúci mieru istoty pravidiel je upravený na:

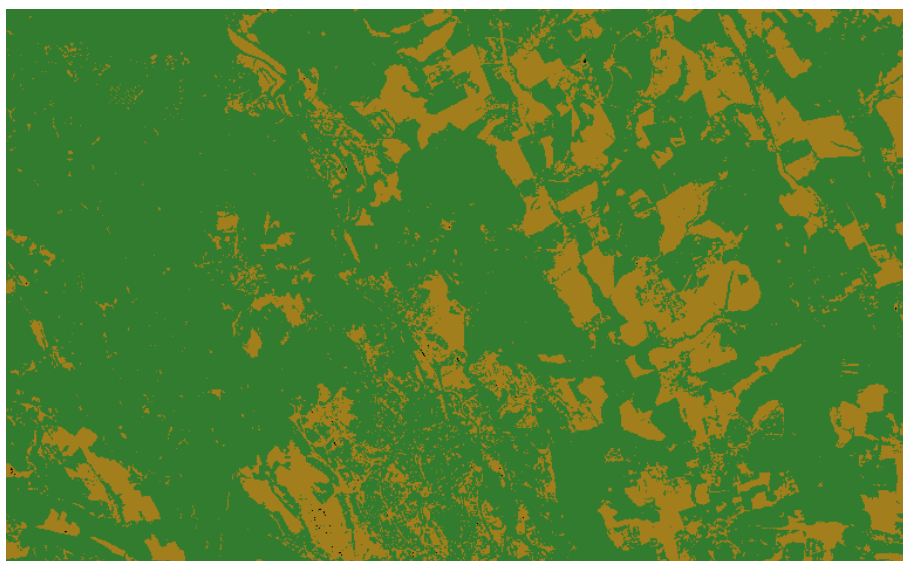
$$r^k = \sum_{p|c^p=C^k} w^p \mu^k(x^p) / \sum_{p=1}^Q w^p \mu^k(x^p), \quad (14)$$

čo spôsobuje, že nesprávne klasifikované a neklasifikované vzorky s vyššími hodnotami váh w^p ovplyvňujú mieru istoty vo väčšej miere ako predtým. Takto upravená miera istoty pravidiel bola zahrnutá aj vo funkcii vhodnosti počas hodnotenia jedincov v genetickom algoritme. Vzťah (13) určujúci výpočet funkcie vhodnosti sa po tejto úprave zmenil na:

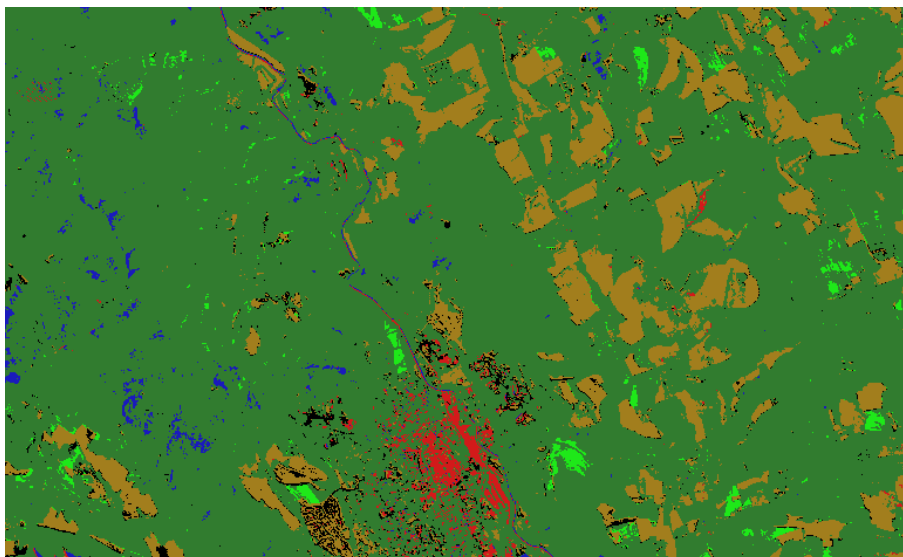
$$f_1 = \text{Cons} \cdot \text{Cov} \cdot r^k. \quad (15)$$

Tieto zmeny boli vykonané z dôvodu potreby získania pravidiel s väčšou presnosťou klasifikácie, ktorú kritériá pokrytia ani konzistencie nebrali v úvahu. Cieľom kritérií pokrytia a konzistencie je maximalizácia počtu relatívne dobre klasifikovaných vzoriek, čo malo za následok vytvorenie veľkého množstva nepresných pravidiel pre triedu vzoriek s najväčšou početnosťou (na satelitných snímkach napr. lesy a polia) a takmer úplnú ignoranciu vzoriek, ktorých početnosť v trénovacej množine bola nižšia (rieky, domy).

Hoci presnosť klasifikácie pri použití pôvodných vzťahov presahovala 70 percentnú úroveň, klasifikátor nebol schopný správne klasifikovať zvyšných 30 % vzoriek. Situáciu pred a po zmene vzťahov ilustrujú obrázky 6 a 7.



Obrázok 6: Vizualizácia klasifikácie pred zmenou funkcie vhodnosti.



Obrázok 7: Vizualizácia klasifikácie po zmene funkcie vhodnosti.

Napriek tomu, že po aplikovaní úprav sa priemerná presnosť klasifikácie výrazne nezlepšila, je zrejmé, že klasifikátor získal väčšiu citlivosť voči triedam s menšou početnosťou prvkov.

Ďalšou snahou o podporu tvorby rôznorodejších pravidiel je zavedenie možnosti reinicializácie genetického algoritmu v prípade konštantnej produkcie pravidiel s nízkou istotou, resp. pri poklese súčtu vhodnosti celej populácie na hodnotu blízku nule. Dôvodom tejto úpravy je fakt, že pravidlá počas evolúcie v genetickom algoritme konvergujú k určitému riešeniu a po vybraní tohto riešenia z populácie do bázy pravidiel, nasleduje pokles váh u vzoriek, ktoré toto riešenie zahŕňa. Populácia sa v tomto momente musí adaptovať na novú situáciu a hľadať nové riešenie. Často to však už nie je možné, keďže pri snahe nájsť predchádzajúce riešenie sa stratí značná časť rôznorodosti jedincov populácie a genetický algoritmus uviazne v lokálnom extréme. Riešením je reštart algoritmu a inicializácia novej populácie.

5 Experimenty

Pomocou vytvoreného programu bolo realizovaných niekoľko experimentov. Výsledkom niektorých z nich sa venuje táto kapitola. Každý experiment obsahuje popis nastavenia parametrov učenia genetického algoritmu, výsledky učenia a testovania, ako aj vizualizáciu klasifikačnej mapy vytvorenej po klasifikácii celej množiny dát. Na záver je pre porovnanie uvedený výsledok klasifikácie podobnej úlohy pomocou neurónovej siete.

5.1 Experiment č. 1

Parametre učenia:

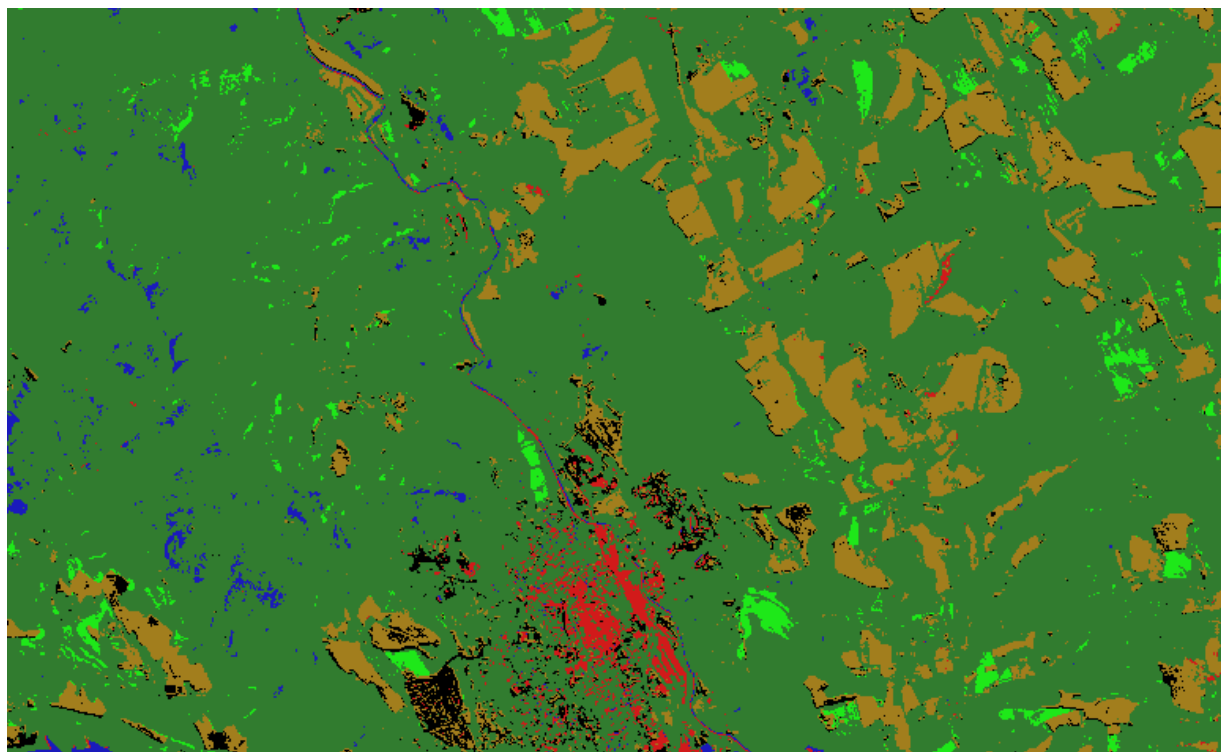
- Veľkosť populácie: 50
- Počet generácií: 10000
- Mutácia: 20 %
- Elitárstvo: 5 %
- Poč. generácií na vyprodukované pravidlo: 20
- Min. istota pravidla: 60 %
- Poč. generácií potrebných na reštart GA pri nízkej istote: 10

Výsledky učenia:

- Počet pravidiel: 24
- Počet správne klasifikovaných vzoriek: 2528 z 3164
- Presnosť klasifikácie: 79,90 %.

Výsledky testovania:

- Počet správne klasifikovaných vzoriek: 2739 z 3167
- Presnosť klasifikácie: 86,49 %



Obrázok 8: Vizualizácia klasifikácie pri experimente č. 1

5.2 Experiment č. 2

Parametre učenia:

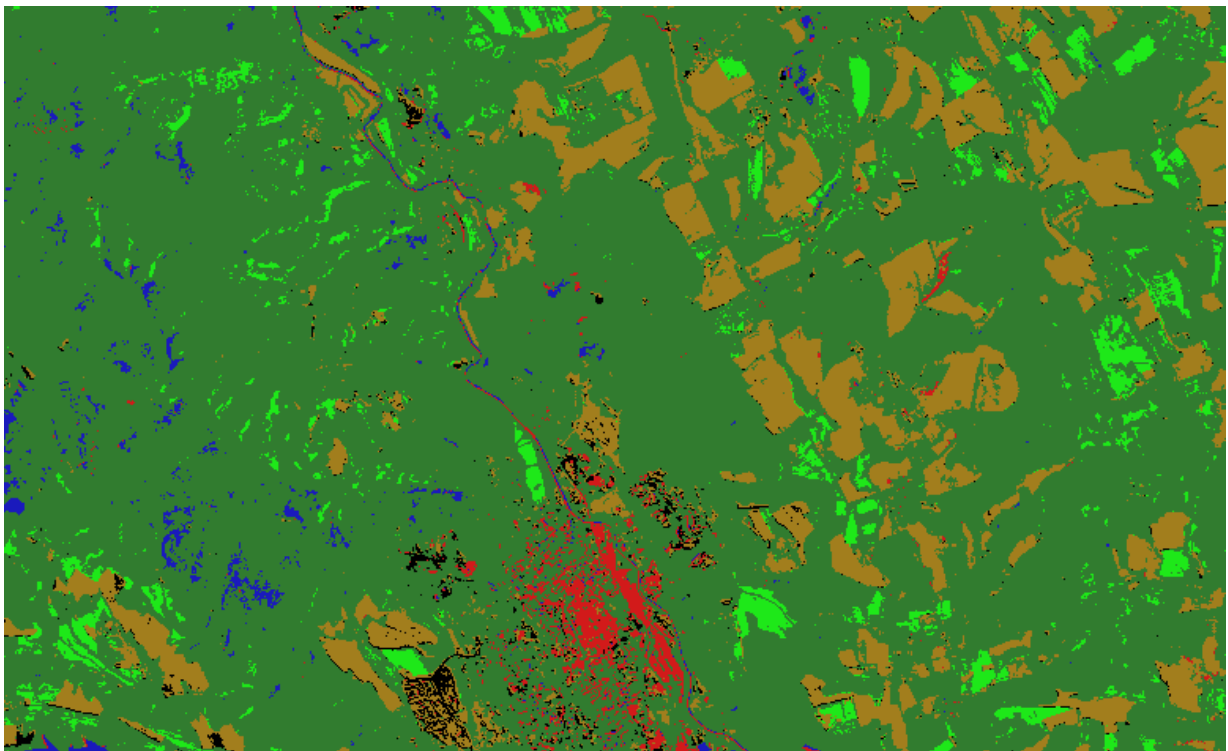
- Veľkosť populácie: 100
- Počet generácií: 1000
- Mutácia: 20 %
- Elitárstvo: 10 %
- Poč. generácií na vyprodukované pravidlo: 20
- Min. istota pravidla: 50 %
- Poč. generácií potrebných na reštart GA pri nízkej istote: 100

Výsledky učenia:

- Počet pravidiel: 50
- Počet správne klasifikovaných vzoriek: 2413 z 3164
- Presnosť klasifikácie: 76,26 %.

Výsledky testovania:

- Počet správne klasifikovaných vzoriek: 2637 z 3167
- Presnosť klasifikácie: 83,26 %



Obrázok 9: Vizualizácia klasifikácie pri experimente č. 2

5.3 Experiment č. 3

Parametre učenia:

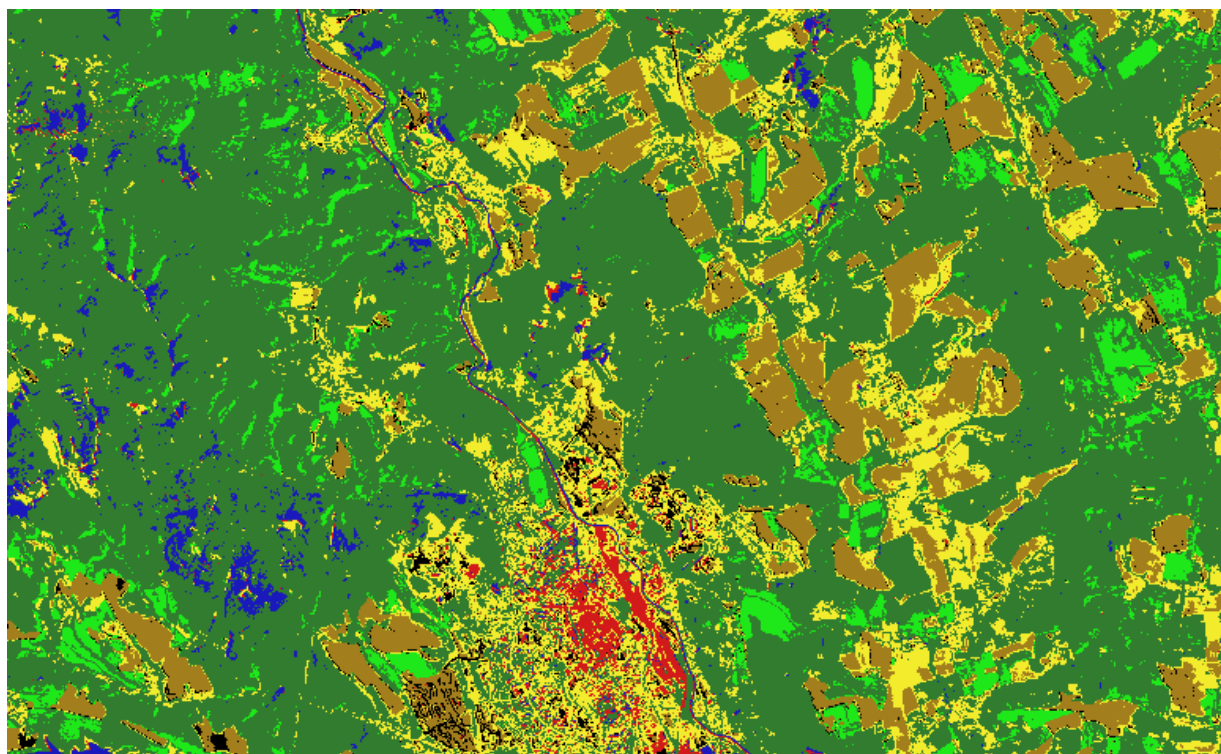
- Veľkosť populácie: 100
- Počet generácií: 10000
- Mutácia: 20 %
- Elitárstvo: 10 %
- Poč. generácií na vyprodukované pravidlo: 20
- Min. istota pravidla: 50 %
- Poč. generácií potrebných na reštart GA pri nízkej istote: 100

Výsledky učenia:

- Počet pravidiel: 500
- Počet správne klasifikovaných vzoriek: 2466 z 3164
- Presnosť klasifikácie: 77,94 %.

Výsledky testovania:

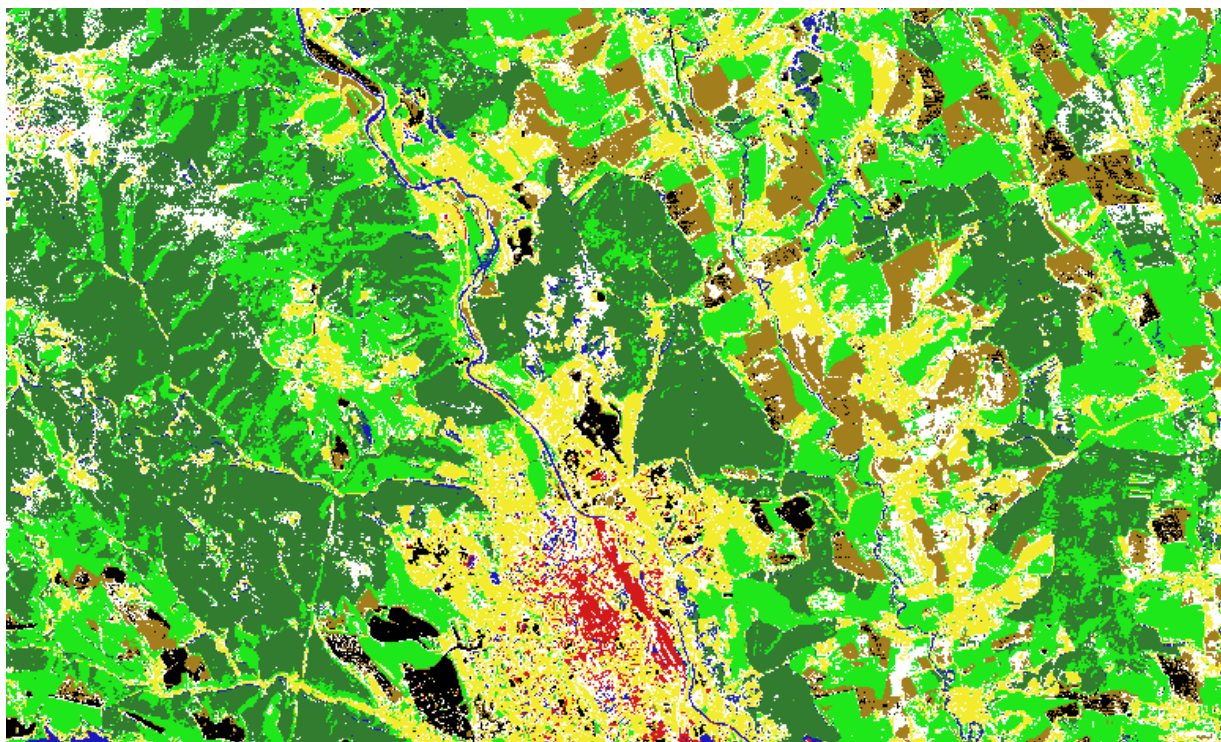
- Počet správne klasifikovaných vzoriek: 2650 z 3167
- Presnosť klasifikácie: 83,68 %



Obrázok 10: Vizualizácia klasifikácie pri experimente č. 3

5.4 Porovnanie s klasifikátorom založeným na neurónovej sieti

Pre porovnanie je na obrázku 11 uvedený výstup z ekvivalentného klasifikátora, doprednej neurónovej siete s dvoma skrytými vrstvami s rozmerom 10x8 neurónov po 10000 cykloch učenia (typu backpropagation) s presnosťou klasifikácie 83,54 %.



Obrázok 11: Vizualizácia z ekvivalentného klasifikátora implementovaného pomocou NS.

6 Záver

Klasifikácia multispektrálnych satelitných dát pomocou fuzzy klasifikátora sa javí ako zaujímavá alternatíva ku podobným klasifikátorom implementovaných pomocou neurónových sietí. Hlavnou výhodou oproti neurónovým sieťam, pri ktorých je prístup k uloženým znalostiam veľmi komplikovaný, je možnosť jednoduchšej extrakcie pravidlovej bázy do podoby zrozumiteľnej človeku.

Nevýhodou je zložitejšia implementácia a nižšia citlivosť na málo početné triedy. Aj napriek použitiu genetického algoritmu (a teda značnej automatizácii návrhu klasifikátora) je stále zložité vybrať správne parametre učenia. Problémy s citlivosťou je však teoreticky možné vyriešiť použitím ďalších úprav pravidlovej bázy, ako je spomenuté v [1].

Pri iteratívnom učení pravidiel by bolo zaujímavé použiť namiesto genetického algoritmu, ktorý má po už po nájdení prvých pár pravidiel tendenciu uviaznuť v nesprávnej časti priestoru riešení (vďaka zmenám váh na vzorkách pri použití metódy AdaBoost), nejaký iný perturbačný algoritmus, ktorý by vychádzal z princípov použitých pri inicializácii populácie (tj. inicializácia podľa neklasifikovaných vzoriek a pod.).

7 Literatúra

- [1] STAVRAKLOUDIS, D. G., GALIDAKI G. N., GITAS I. Z., THEOCHARIS J. B.: *Enhancing the Interpretability of Genetic Fuzzy Classifiers in Land Cover Classification from Hyperspectral Satellite Imagery*. WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain. 2010. Dostupný na internete: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5584855
- [2] FREUND Y., SCHAPIRE R.: *Experiments with a new boosting algorithm*. in Proc. 13th Int. Conf. Machine Learning, pp. 148–156, 1996.
- [3] HOFFMANN F.: *Combining boosting and evolutionary algorithms for learning of fuzzy classification rules*. Fuzzy Sets Syst., vol. 14, no. 1, pp. 47–58, 2004.
- [4] del JESUS M.J., HOFFMANN F., NAVASCUÉS L.J., SÁNCHEZ L.: *Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms*. IEEE Trans. Fuzzy Syst., vol. 12, no. 3, pp. 296–308, 2004.
- [5] WIKIPEDIA: *Multispectral image*. [online]. [citované 2011-12-07]. Dostupný na internete: http://en.wikipedia.org/wiki/Multispectral_imaging
- [6] WIKIPEDIA: *Hyperspectral imaging*. [online]. [citované 2011-12-07]. Dostupný na internete: http://en.wikipedia.org/wiki/Hyperspectral_imaging
- [7] HOFFMANN F.: *Boosting a Genetic Fuzzy Classifier*. R. Inst. of Technol., Stockholm, Sweeden, 2001. Dostupný na internete: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=943782