SIMULACE FUZZY MATEMATIKY VE FUZZY LOGIC TOOLBOX A JEJÍ VYUŽITÍ VE SHLUKOVÉ ANALÝZE A PREDIKCI

Libor Žák

Ústav matematiky, Fakulta strojního inženýrství, VUT v Brně
Technická 2, 616 69 Brno, Česká republika

Abstrakt:
Článek pojednává o využití Matlabu a zvláště Fuzzy Logic Toolboxu při shlukování objektů s využitím fuzzy množin a pro predikci časové řady pomocí FIS. Popisuje příkazy pro fuzzy matematiku a jejich využití spolu se shlukovací procedurou Fuzzy C-means clustering pro shlukování vágních objektů. Článek dále pojednává o vhodné volbě a využití FIS (typu Sugeno) pro predikci časové řady.

1. Úvod

Oblastí mého zájmu jsou fuzzy množiny a jejich využití v různých oborech. Poslední dobou jsem se snažil využít fuzzy množiny ve dvou oblastech. První z nich je shluková analýza a další pak predikce časových řad. Pro výpočet konkrétních příkladů jsem hledal vhodný prostředek. Jako nejvhodnější se mi jevil systém **Matlab**. Velkou výhodou tohoto systému je vložený balík programů zabývající se fuzzy problematikou – **Fuzzy Logic Toolbox**. Tento balík obsahuje jak Fuzzy Interface Systém (FIS) a k němu přidané některé výpočty s fuzzy množinami, tak i některé shlukovací metody – **Fuzzy C-means clustering**. Proto jsem tento programový produkt zvolil pro řešení svých příkladů.

V tomto článku bych chtěl popsat některé funkce, které jsem vytvořil pro práci s fuzzy množinami a jejich využití (spolu s **Fuzzy Logic Toolbox**) na řešení některých problémů z oblasti shlukování a predikce časových řad.

2. Fuzzy množiny

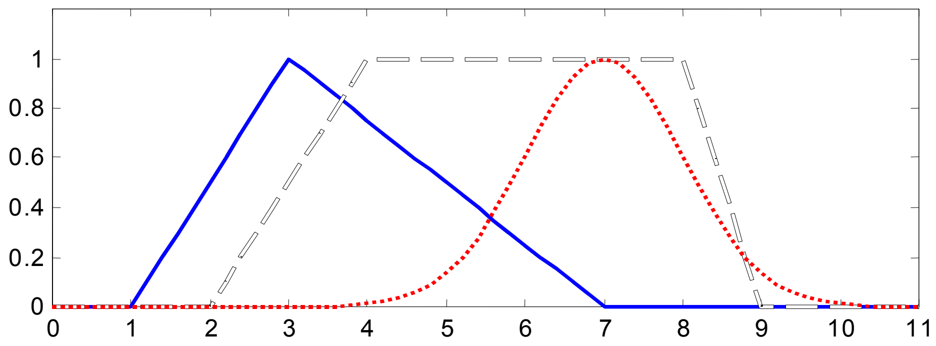
Fuzzy množinou A se rozumí dvojice (U, μ_A), kde U je univerzum a μ_A: U $\rightarrow \langle 0,1 \rangle$ je funkce popisující příslušnost prvků z U do fuzzy množiny A. Tuto příslušnost označíme $\mu_A(x)$. Fuzzy množina je zobecnění „klasické“ množiny, neboť pro příslušnost u „klasické“ množiny A platí μ_A: U $\rightarrow \{0, 1\}$ a $x \in A \Leftrightarrow \mu_A(x) = 1$ a $x \notin A \Leftrightarrow \mu_A(x) = 0$.

Podle typu funkce příslušnosti μ_A lze fuzzy množiny rozdělit na
- trojúhelníkové
- lichobežníkové
- s Gaussovou křivkou příslušnosti, atd.

Fuzzy množiny v Matlabu uchovávám ve tvaru matice *n* x 2, kde první sloupec popisuje *x*-složku a druhý sloupec popisuje *y*- složku. Tyto základní typy fuzzy množin se dají vytvořit pomocí předchystaných funkcí.

Např.:

```
x=(0:0.2:10)';
y1=trimf(x,[1 3 7]); ........značeno plnou čárou
y2=trapmf(x,[2 4 8 9]); ...značeno přerušovanou čárou
y3=gaussmf(x,[1 7]); ........značeno tečkovanou čárou
plot(x, [y1, y2, y3]);
axis([0 11 0 1.2])
```



Obr. 1 Fuzzy množiny

Základní operace potřebné k počítání s fuzzy množinami jsem již vytvořil v podobě m-souborů. Jedná se zejména o tyto operace: sjednocení, průnik a doplněk fuzzy množin. Tyto operace jsou většinou definované ve tvaru:

Sjednocení fuzzy množin A a B:

$A \cup B = (U, \mu_{A\cup B})$, kde $\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}\ \forall x \in U$

Průnik fuzzy množin A a B:

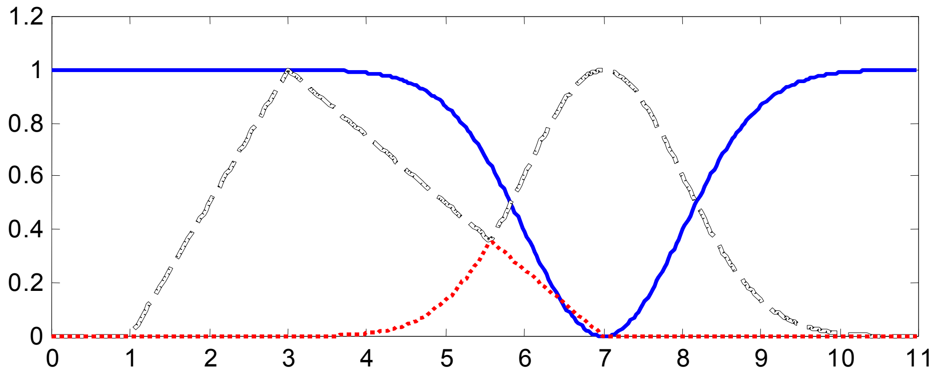
$A \cap B = (U, \mu_{A\cap B})$, kde $\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}\ \forall x \in U$

Doplněk fuzzy množiny A:

$\mathrm{com}(A) = (U, \mu_{\mathrm{com}(A)})$, kde $\mu_{\mathrm{com}(A)}(x) = 1 - \mu_A(x)\ \forall x \in U$

A příslušné funkce:

```
function [ak]=f_komp(a);
% pocita fuzzy komplement a=[x;ay]

function [pr]=f_prun(a,b);
% pocita fuzzy prunik  a=[x;ay]
%                      b=[x;by]

function [sj]=f_sjed(a,b);
% pocita fuzzy sjadnoceni  a=[x;ay]
%                          b=[x;by]

x=(0:0.03:11)';
y1=trimf(x,[1 3 7]);
y2=trapmf(x,[2 4 8 9]);
y3=gaussmf(x,[1 7]);
a=[x';y1'];
b=[x';y3'];
[bk]=f_komp(b); ........značeno plnou čárou
[sj]=f_sjed(a,b); ]); .....značeno přerušovanou čárou
[pr]=f_prun(a,b); .......značeno tečkovanou čárou
plot(pr(1,:),[bk(2,:); sj(2,:); pr(2,:)])
axis([0 11 0 1.2])
```



Obr. 2 Operace s fuzzy množinami

Samozřejmě lze tyto operace definovat obecněji a to například přes S-normy a T-normy.

V praxi používané fuzzy množiny by měly být normální a konvexní. Fuzzy množina A se nazývá **normální**, jestliže Ker(A) ≠ ∅, kde Ker(A) je jádro fuzzy množiny definované: Ker(A) = $\{x \in U;\ \mu_A(x) = \mathbf{1}\}$. Fuzzy množina A=(U, μ_A) se nazývá **konvexní**, jestliže U je lineární prostor a pro každé dva prvky $x, y \in U$ a pro každé $\lambda \in \langle 0, 1\rangle$ platí: $\mu_A(\lambda x+(1-\lambda)y) \geq \mu_A(x) \wedge \mu_A(y)$.

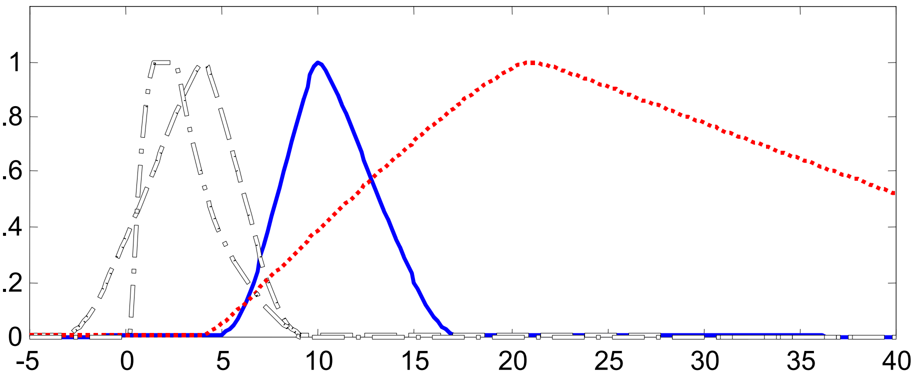
Fuzzy množiny definované pomocí příkazů trimf, trapmf, gaussmf, adt. jsou normální a konvexní. Dále splňují požadavek částečné spojitosti fumkce příslušnosti (μ_A). Takto definované fuzzy množiny se nazývají fuzzy čísla. Pomocí Zadehova principu rozšíření můžeme s fuzzy čísly pracovat způsobem obvyklým pro čísla. Zadehův princip rozšíření lze definovat ve tvaru:
Mějme funkci $f : U_1 \times U_2 \times \ldots \times U_n \rightarrow V$ a nechť A_i je fuzzy množina na U_i : $A_i = (U_i, \mu_{A_i})$,

pak fuzzy množina $f(A_1, A_2, \ldots, A_n)$ nad univerzem V má funkci příslušnosti:

$$\mu_{f(A_1,A_2,\ldots,A_n)}(y) = \sup_{\substack{(x_1,x_2,\ldots,x_n)\in U_1\times U_2\times\ldots\times U_n;\\ y=f(x_1,x_2,\ldots,x_n)}} \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \ldots, \mu_{A_n}(x_n)\}$$

Fuzzy Logic Toolbox umožňuje pomocí funkce `fuzarith` s fuzzy čísly provádět operace: sčítat, odčítat, násobit, dělit.

```
x=(-5:0.2:40)';
y1=trimf(x,[1 3 7]);
y2=trapmf(x,[2 4 8 9]);
y3=gaussmf(x,[1 7]);
z1=fuzarith(x, y1, y3, 'sum'); …..značeno plnou čárou
z2=fuzarith(x, y3, y1, 'sub'); …..značeno přerušovanou čárou
z3=fuzarith(x, y1, y3, 'prod'); …značeno tečkovanou čárou
z4=fuzarith(x, y2, y1, 'div');  …..značeno čerchovanou čárou
plot(x,[z1, z2, z3, z4]);
axis([0 11 0 1.2]);
```



Obr. 3 Výsledné fuzzy čísla

Zadehův princip rozšíření se prakticky provádí tak, že fuzzy množiny pomocí α-řezů „nařežeme“ na intervaly. S využitím intervalové aritmetiky vypočteme α-řezy výsledné fuzzy množiny a pomocí interpolace z nich sestrojíme výslednou fuzzy množinu (**α-řez** fuzzy množiny A, kde $\alpha \in L$, je klasická množina $A_\alpha = \{x;\ \mu_A(x) \geq \alpha\}$).

K těmto operacím (a samozřejmě i k dalším) jsem vytvořil řadu m-souborů. Namátkou vybírám např.:

```
function v=f_vyska(f);
%Funkce spocita vysku fm

function [int]=fm_rez(a,alfa);
%Funkce spocita jeden alfa_rez na fuzzy mnozine

function [int]=fm_rezy(a,rezy);
%Funkce spocita alfa_rezy na fuzzy mnozine

function int=int_souc(a,b);
% soucet dvou intervalu

function int=int_nas(a,b);
% nasobeni dvou intervalu

function int=int_2odm(a);
% druha odmocnina intervalu
```

Fuzzy Logic Toolbox byl vytvořen pro nejčastější aplikaci fuzzy množin a to pro návrh Fuzzy Interface Systemu. Dříve se pro tento systém fuzzy pravidel používal název **fuzzy regulátor**. Ve Fuzzy Logic Toolboxu je tento systém velmi zdařile naprogramován včetně velmi příjemného prostředí pro odladění FSI.

3. Využití fuzzy množin ve shlukování

Jednou z oblastí, kterou se zabývám, je shluková analýza a využití fuzzy množin ve shlukové analýze. V tomto směru mi poskytl Matlab a Fuzzy Logic Toolbox prostředí a prostředky pro vytvoření řady algoritmů zabývající se shlukovou analýzou. V Matlabu jsem vytvořil řadu m-souborů, pomocí nichž jsem prováděl výpočty klasického shlukování. Jedná se zvláště o soubory, které počítají nepodobnost (podobnost) objektů. Tato nepodobnost se

nejčastěji definuje jako vzdálenost objektů v Euclidově prostoru. Může být definovaná i jiným způsobem, ale musí vyhovovat požadavků kladených na nepodobnost:
Nepodobnost objektů je zobrazení $\mathbf{d} : \boldsymbol{O} \times \boldsymbol{O} \rightarrow \mathbf{R}^{+}{}_{0}$, pro které platí:

$$\mathbf{d}(O_h, O_s) = 0 \Leftrightarrow O_h = O_s,$$
$$\mathbf{d}(O_h, O_s) \geq 0,$$
$$\mathbf{d}(O_h, O_s) = \mathbf{d}(O_s, O_h).$$

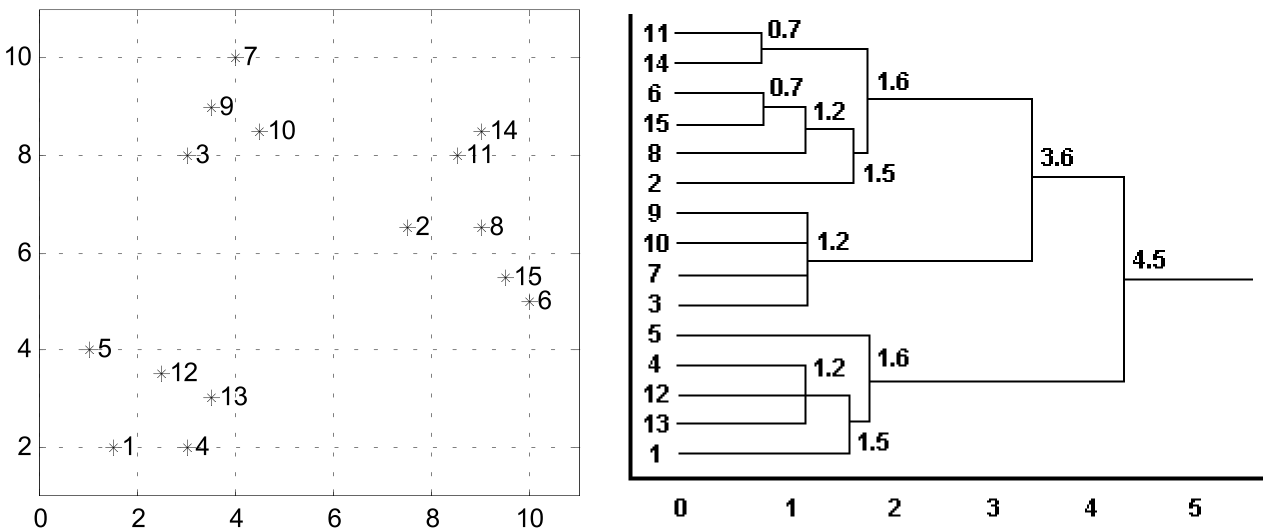
Pomocí nepodobnosti objektů lze definovat nepodobnost shluků (více v [1, 3]).
Namátkou vybírám funkce:

```
function nep=b_nep(b1,b2,vaha,G);
% funkce spocita nepodobnost (vzdalenost) bodu b1 b2
%G - matice metriky, vaha - vaha pro jednotlive dimenze

function snep=shl_nep(bbn,sh1,sh2,v_b);
% funkce najde nepodobnost dvou shluku na zaklade nepoddobnosti
bodu

function poc=po_pr_sh(sh);
% funkce udava pocet prvku ve shluku

function [p_s,p_ob]=sh_par(sh);
% funkce da pocet shluku a pocet objektu
```

S využitím předcházejících pojmů lze definovat hierarchické a nehierarchické shlukováni.

```
function hl=sh_hiea(bb,typ,prn);
% funkce simuluje hiearchicke shlukovani

function sh_nhie(bb,sh0,typ,po_1b,prn);
% funkce simuluje nehiearchicke shlukovani
```

Například hierarchické shlukování dává výsledek ve tvaru podobnostního stromu.



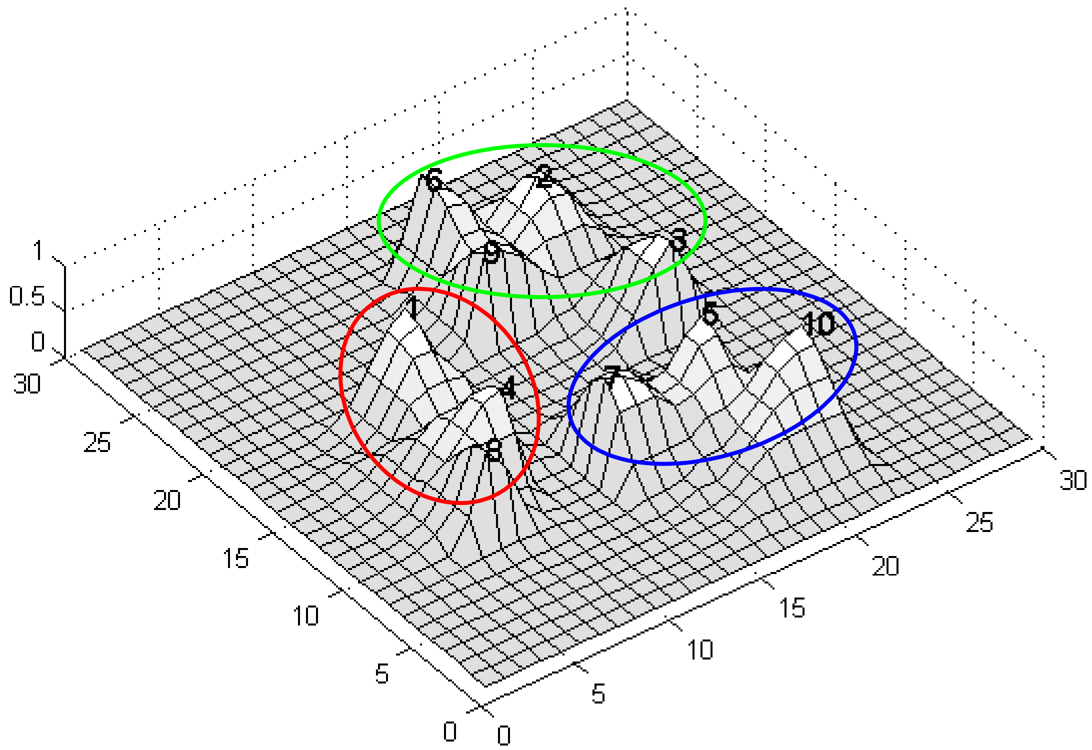
Obr. 4. Objekty a jejich výsledný podobnostní strom hierarchického shlukování

Pro klasické objekty lze využít fuzzy shlukování definované J.C. Bezdekem. Toto shlukování je naprogramováno v Fuzzy Logic Toolboxu pod funkcí *fcm* (Fuzzy C-mean Clustering).

Mým dalším zájmem je shlukování objektů, které nejsou přesně definovány. Pro jejich vágnější popis se využívají fuzzy množiny. Pro tyto fuzzy objekty jsem vytvořil obdobné funkce jako pro klasické objekty. Například:

```
function hl=sk_hiear(fbb,typ,prn,typ_def);
% funkce pocita hiearchicke shlukovani fuzzy objektu

function sh=sk_nhie(fbb,sh0,typ,prn);
% funkce pocita nehiearchicke shlukovani fuzzy objektu
```

Příklad výsledných shluků nehierarchického shlukování fuzzy objektů:

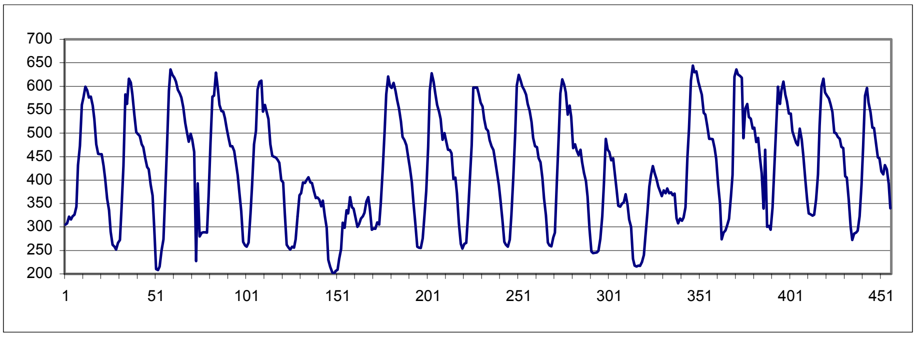


Obr. 5. Výsledné shluky nehierarchického shlukování fuzzy objektů

Více informací o fuzzy objektech a jejich shlukování lze nalézt v [6, 7, 8, 9].

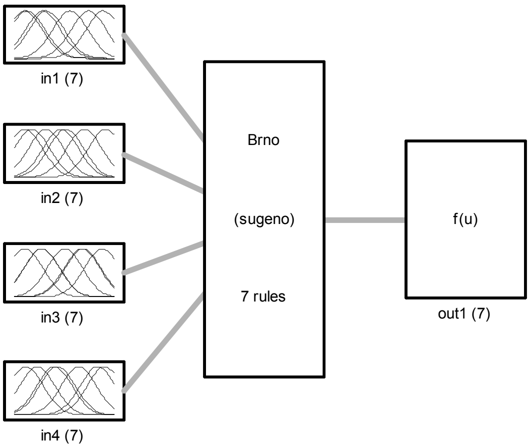
4. Využití fuzzy množin v predikci časových řad

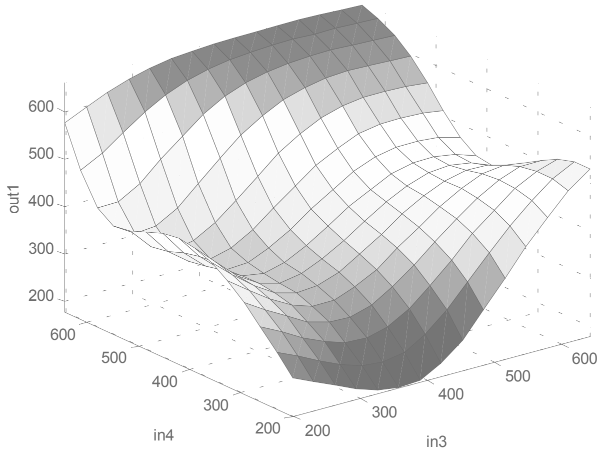
Další oblastí, ve které využívám Matlab a zvláště FIS obsažený v Fuzzy Logic Toolboxu, je predikce časových řad. Využily jsme FIS typu Sugeno pro predikci časové řady odběru tepla města Brna. FIS byl navržen tak, aby na základě předcházejících hodnot předpovídal jednu následující hodnotu. Požadavek byl na předpověď dalších 48 hodnot. Šlo by samozřejmě definovat FIS, který předpovídá v jednom kroku 48 hodnot zaráz. Vyzkoušeli jsme ale řešení, kdy jsme měli FIS pro předpověď jedné další hodnoty. Pro předpověď dalších hodnot jsme uvažovali předpovězenou hodnotu jako reálnou a s její pomocí jsme předpověděli stejným FIS další hodnotu. Tento postup jsme opakovali 48 krát. Tento postup jsme použili na časovou řadu spotřeby tepla města Brna v MW, naměřené od 1:00 6.2.1998 do 1:00 25.2.98. Graf řady je ukázán na obr.6.



Obr. 6. Město Brno – spotřeba tepla

Časové řadě spotřeby tepla města Brna vyhovoval FIS typu Sugeno se 4 vstupními proměnnými (tedy další hodnota byla předpovězena na základě 4 předcházejících hodnot) a 7 vstupními hodnotami.

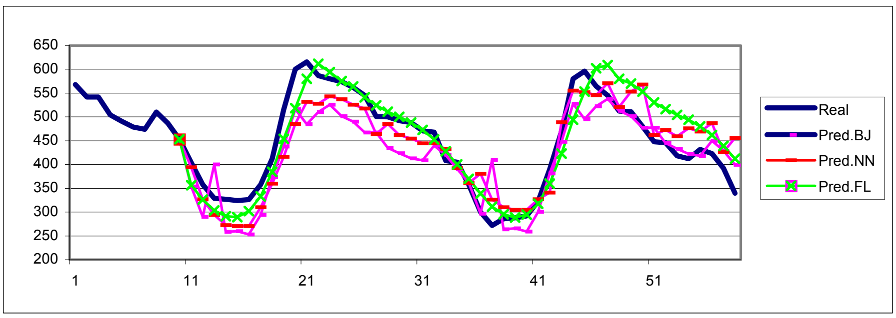




Obr. 7. Regulátor Sugeno pro Brno

Obr. 8. Část regulační plochy

I když jsme další předpovězené hodnoty předpovídali již na základě hodnot předpovězených, FIS se dokázal velmi úspěšně „naladit" na časovou řadu. Tato úspěšnost je zřejmá z porovnání s jinými druhy predikce. Nejlepší predikce bylo dosaženo pomocí fuzzy logiky, kdy MAPE = 0.082 (MAPE – průměrná odchylka předpovězeného průběhu od skutečného). O něco horší výsledky dalo použití neuronových sítí, kdy MAPE = 0.096 a největší chyba byla vypočtena při použití Box-Jenkinsonovy metodologie, kdy MAPE = =0.106. Viz obr.9.



Obr. 9. Město Brno – spotřeba tepla – predikce

Více informací o predikci časových řad a zkoumání časových řad pomocí FIS by měl přinést další článek.

5. Závěr

Vestavěný Fuzzy Logic Toolbox se pro mou práci velmi osvědčil. Zvláště pak FIS je velmi zdařile konstruován. Přínosem je také funkce Fuzzy C-mean clustering (fcm), která pracuje podle algoritmu J.C. Bezdeka. K výše uvedeným funkcím jsem vytvořil řadu dalších funkcí pracujících s fuzzy množinami a simulující metody shlukové analýzy. Ze zkušenosti s různými výpočty mohu říci, že systém Matlab je velmi vhodný pro tento typ výpočtů.

Literatura

[1] ANDERBERG, M. R. *Cluster Analysis for Applications*. Academic Press, New York 1973

[2] BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Alghorithms*. Plenum Press, New York 1981.

[3] LUKASOVÁ, A.- ŠARMANOVÁ, J. *Metody shlukové analýzy*, SNTL, Praha 1985.

[4] NOVÁK, V. *Fuzzy množiny a jejich aplikace*. SNTL, Praha 1986.

[5] ZADEH, L. A. *Fuzzy Sets and Their Application to Pattern Classification and Cluster Analysis.* In Classification and Clustering . Academic Press, New York 1977.

[6] ŽÁK, L. Clustering of Fuzzy Objects. Sborník, *Mendel 2000*, 6th International Conference on Soft Computing. Brno, 2000, pp 310 – 317 ISBN 80-214-1131-7.

[7] ŽÁK, L. Zobecnění fuzzy shlukování pro fuzzy objekty. Sborník *Inteligentní systémy pro praxi*, Luhačovice, 2000, pp 59 - 68, ISBN 80-238-6140-9.

[8] ŽÁK, L Generalization of Fuzzy Clustering for Vaguely Defined Objects, *9th Zittau Fuzzy Colloquium*, Zittau, 2001, pp 268-277, ISBN 3-9808089-0-4.

[9] ŽÁK, L. Shlukování vágně definovaných objektů, *PhD Thesis*, Vysoké učení technické, Brno, 2002, ISSN 1213-4198.

Acknowledgement.

Tento článek je součástí řešení výzkumného záměru CEZ: J22/98:261100009 "Netradiční metody studia komplexních a neurčitých systémů".

Kontakt:

tel.: +420-05-41142550

E-mail: zak@um.fme.vutbr.cz