PREDPOVEĎ PRIETOKOV POUŽITÍM MATEMATICKÝCH A ŠTATISTICKÝCH METÓD

Martin Suchár, Milan Čistý, Peter Valent
Katedra vodného hospodárstva krajiny, Slovenská technická univerzita v Bratislave

Abstract

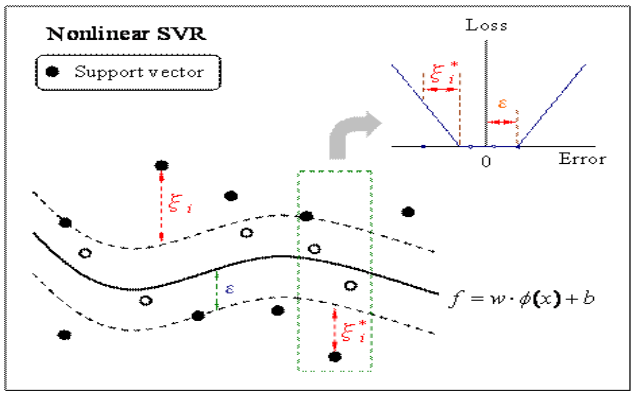
Flow forecasting is important part of water management, whether in economical terms and also in terms of security, which is currently very actual, because in last few years was Slovak republic (SR) for nearly every year affected by flooding. Flows are very important part of the water balance of the SR, assessed each year and providing information of the amount of water that flowed through our country. That's why it's necessary to understand and measure their current states, which are then used to their future forecast. This work is describing combined mathematical and statistical calculation model, the keystone of which is the simulation program created in Matlab, which involves a combination of genetic algorithm and data driven learning method of support vectors machines (SVM). This combination takes the best from these two methods.

1 Úvod

Predpoveď prietokov spolu s ďalšími hydrologickými veličinami ako sú zrážkové úhrny, evapotranspirácia, merania vlhkosti a ostatných parametrov podieľajúcich sa na celkovej vodnej bilancie sú dôležité z rôznych hľadísk. Hlavne v posledných desaťročiach je sledovanie hladín tokov čoraz dôležitejšie, či už v záujmoch bezpečnosti alebo z ekonomického hľadiska.

V predkladanej práci sa zaoberáme predpoveďou prietokov, kombináciou dvoch výpočtových algoritmov - pomocných vektorov (SVM) a genetického algoritmu (GA) v prostredí programu Matlab.

SVM sú výpočtovou metódou, založenou na umelej inteligencii a vyvinutej na základe štatistickej teórie učenia. SVM formulujú kvadratický optimalizačný problém, ktorý sa vyhýba problémom s lokálnym minimom, čo z nich robí často výkonnejší a presnejší nástroj oproti starším, iteratívnym učiacim algoritmom, ako je napr. viacvrstvový perceptrón (MLP). Model SVM regresie (Support vector regression model) s parametrami *w* a *b,* možno vyjadriť ako $f(x)=w \cdot Ø(x)+b$, kde *y* je výstup modelu a vstup modelu *x* je mapovaný do funkčného priestoru, nelineárnou funkciou Ø(x). Na obr. 1 je znázornená štruktúra takéhoto modelu. Parametre, ktoré určujú nelineárne SVM, sú konštanty C (Cost = suma), polomer rozpätia ε a parameter *x*. Určenie optimálnych hodnôt parametrov je heuristický proces riešený často metódou pokus – omyl, ktorý sa v tejto práci snažíme nahradiť použitím GA. Ten vychádza z princípov prirodzeného výberu a je schopný poskytnúť uspokojivé výsledky.



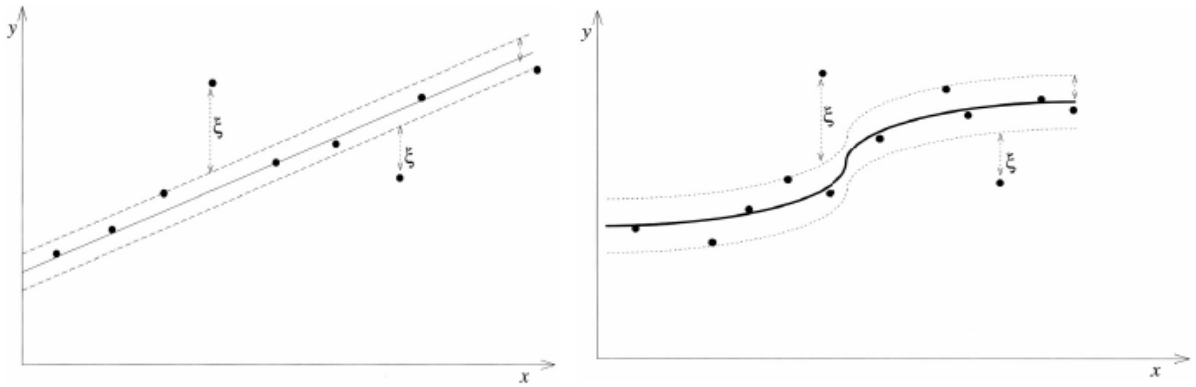
Obr. 1: Model SVM nelineárnej regresie (SVR), ξi a ξ_i^* určujú hornú a dolnú hranicu rozpätia strát (chýb), zodpovedajúcich ε-ignorujúcej stratovej funkcii. C je pozitívna konštanta, ktorá určuje stupeň penalizačnej straty, keď počas tréningu dôjde k chybe (Smola, Scholkopf, 1998)

2 Materiál a metódy

Vedný odbor, ktorý sa zaoberá prenosom ľudských inteligenčných schopností do počítačov, má názov umelá inteligencia. Jej hlavnými pod kategóriami sú expertné systémy, GA, neurónové siete (NS) spolu so SVM.

Medzi takéto systémy zaraďujeme okrem NS a SVM aj ďalšie dátovo riadené algoritmy patriace do skupiny tzv. učiacich techník, ktoré využívajú strojové učenie na modeli alebo na vzore založenom na vstupných dátach. Využívajú sa na riešenie regresných alebo klasifikačných úloh. SVM používa oproti štandardným NS založených na metóde spätného šírenia chýb, (medzi ktoré patrí MLP) princíp tzv. štrukturálnej minimalizácie namiesto iba minimalizácie chyby – (Vapnik 1995). Pri trénovaní siete MLP, je jediným cieľom minimalizovať celkovú chybu. Pri SVM sa simultánne minimalizuje chyba aj zložitosť modelu. Použitie tohto princípu vedie k umožneniu presnejších predpovedí pre dáta, ktoré neboli použité pri trénovaní SVM. Ďalším dôležitým rozdielom je spôsob určenia architektúry modelu a jeho váh. Architektúru MLP siete a jej váhy sa určujú na základe postupu pokus – omyl, čiže iteratívneho procesu a metódy spätného šírenia chýb, čo je časovo náročné. Vápnik (1995) upustil od časovo náročných trénovacích procesov a vyjadril určenie architektúry a váh ako kvadratický optimalizačný problém, ktorý možno riešiť štandardnými deterministickými algoritmami. A čo viac, pri nelineárnych úlohách SVM mapuje vstupný priestor do viacrozmerného funkčného priestoru (zobrazením vstupu do dostatočne veľkého funkčného priestoru sa problém stane lineárne separovateľný) a následne sa použije jadrová (kernel) funkcia namiesto viacrozmerného skalárneho súčinu. To má za následok, že výsledné riešenie je jednoduchšie, unikátne a optimálne bez problémov s lokálnym minimom.

Pri predpovedaní prietokov, ide o regresný problém a teda o snahu naučiť sa nejakú nelineárnu funkciu pomocou lineárnej funkcie v priestore veľkej dimenzie, ktorý je definovaný použitím tzv. kernelu. Podrobný popis uvádza Vápnik (1997). Na selekciu funkčného priestoru musíme nájsť a optimalizovať nejaké hranice alebo okraje. Celý postup je založený na vhodnej stratovej funkcii, ktorá ignoruje chyby v určitej vzdialenosti od skutočnej hodnoty. Tento typ funkcie sa nazýva ε-intensive loss function.



Obr. 2: Nelineárna regresná funkcia Lineárna regresia funkcie

Uplatnenie SVM metódy v úlohách regresnej analýzy teda spočíva vo vyhľadávaní optimálneho pásma k regresnej funkcii a na definovaní stratovej funkcie. Prostredníctvom optimálneho pásma sa reguluje veľkosť chyby aproximácie. Preto sa toto pásmo nazýva chyba aproximácie pozorovaných dát prostredníctvom regresnej funkcie a môžeme ho chápať ako mieru chýb regresie. Zmyslom zavedenia stratovej funkcie je stanovenie miery presnosti aproximácie, ktorou sa určuje priebeh zanedbávania malých chýb v rámci vymedzeného pásma. Stratová funkcia ignoruje chyby, ktoré sa nachádzajú v stanovenom pásme pozorovaných (skutočných) hodnôt a tie body, ktoré sa nachádzajú mimo pásma sú penalizované prostredníctvom ε-ignorujúcej stratovej funkcie (ε-insensitive loss function) podľa Vapnika.

Správne určenie regresnej funkcie a nastavenie SVM, závisí od nájdenia optimálnych parametrov ako už bolo spomenuté v úvode. Proces hľadania vhodných parametrov je preto veľmi dôležitý. Kvalita výsledkov je na ňom priamo závislá a tak hľadanie pomocou metódy pokus-omyl, ktoré je často zdĺhavé a nevedie k uspokojivým výsledkom, sa snažíme nahradiť použitím GA, ktorý vyhľadá vhodné parametre pre SVM.

GA je všeobecne adaptívna optimalizačná metóda, založená na priamej analógii Darwinovej teórie prirodzeného vývoja a genetiky v biologických systémoch. Tento heuristický postup sa snaží nájsť pomocou princípov evolučnej biológie riešenie zložitých problémov, pre ktoré neexistuje použiteľný exaktní algoritmus. GA, resp. všetky postupy patriace medzi tzv. evolučné algoritmy používajúce techniky napodobňujúce evolučné procesy z biológie - dedičnosť, mutácia, prirodzený výber a kríženie na „šľachtenie“ riešení zadanej úlohy. Každé riešenie úlohy sa nazýva chromozóm a je tvorené binárnym reťazcom o danej dĺžke, ktorá je rovnaká pre všetky chromozómy danej populácie. Populácia je konečná množina chromozómov. Základná populácia resp. nultá generácia populácie je začiatočný stav riešenia. Vývoj k optimálnemu riešeniu prebieha prirodzeným vývojom populácií. Nultá generácia je vygenerovaná náhodne, vygenerované chromozómy, musia byť riešením problému.

Proces reprodukcie:

- Výber chromozómov na kríženie či mutáciu (pseudonáhodný výber podľa pravdepodobnosti úmernej jeho fitness)
- Kríženie chromozómov (výmena podreťazcov, kde môže prebiehať kríženie jednobodovo, či viacbodovo)
- Mutácia, náhodne zmutujú niektoré gény, mutuje sa s malou pravdepodobnosťou, aby zostala zachovaná genetická informácia

Každá populácia sa pomocou reprodukcie zdokonaľuje a to na základe ohodnotenia chromozómov pomocou funkcie f(x) nazývanej fitness, ktorá vyjadruje kvalitu riešenia reprezentovaného týmto jedincom. Podľa tejto kvality sú stochasticky vybraní jedinci, ktorí sú modifikovaní (pomocou mutácie a kríženia), čím vznikne nová populácie. Tento postup sa iteratívne opakuje, čím sa kvalita riešenia v populácii postupne vylepšuje. V procese riešenia pomocou genetických algoritmov ide o hľadanie globálneho maxima funkcie fitness, čiže ide o to nájsť najlepšie ohodnotené riešenie problému v stavovom priestore.

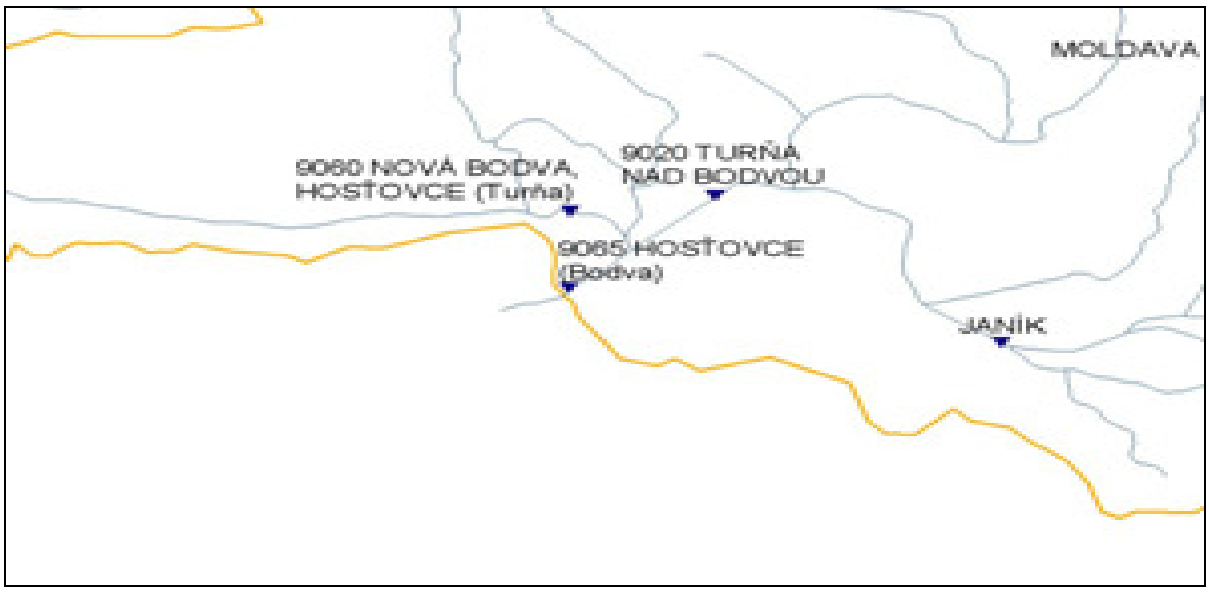
Pre každú novú generáciu platí:

- počet chromozómov je rovnaký
- krížením a mutáciou vznikajú nové chromozómy
- chromozómy s nízkym fitness ohodnotením sú nahradené chromozómami s vyšším ohodnotením

Niekedy sa najlepšie chromozómy z predchádzajúcej generácie zachovávajú. Riešenie sa zastaví buď po dosiahnutí zadanej cieľovej hodnoty, alebo po dopredu stanovenom počte generácií.

3 Záujmové územie a dáta

Na predpoveď prietokov, boli ako vstupné dáta použité denné prietoky z meracích zariadení nachádzajúce sa v záujmovom území. Na obr. č 3 je situácia rieky Bodva, do ktorej sa vlievajú tri rieky. V obci Turňa nad Bodvou (9020) a Nová Bodva Hosťovce (Turňa, 9050) sú meracie stanice a na tretej v Janíkoch nie. Ďalšie meracie zariadenie v Hosťovciach (9065) je za miestom sútoku všetkých troch riek do Bodvy. Namerané dáta sú v rozpätí rokov 1998-2002 a 2004-2006. Tieto údaje bolo nutné rozdeliť na trénovacie, testovacie a dáta kríženej validácie. Testovací súbor dát bol pevne stanovený v období od 3.11.2004 do 30.12.2006 a tvorí ho matica dát 789 x 7 (obr. č. 4). Údaje z trénovacieho a validačného dátového súboru sa budú meniť použitím funkcie v Matlabe, v pomere 80% pre trénovacie dáta a 20% pre validačné z celkového počtu dát (matice) veľkosti 1453 x 7 záznamov. Týmto spôsobom je zaručená stála obmena dátových súborov pre každý výpočtový cyklus. Výhodou oproti stabilnému rozdeleniu dát je, že výpočtový algoritmus sa pri stálej obmene nenaučí určitú postupnosť rozdelenia údajov, ktorú potom pri ďalšom výpočtovom cykle iba zopakuje, ale je nútený vytvárať stále nové kombinácie, čím je zaručená unikátnosť každého výpočtového cyklu.



Obr. 3: Situácia rieky Bodva

den	den pre matlab	HT-3	TB-3	HT-2	TB-2	HT-1	TB-1	Bodva
3.11.2004	38294	0.15	1.16	0.135	1.132	0.135	1.095	1.31
4.11.2004	38295	0.135	1.132	0.135	1.095	0.175	1.057	1.232
5.11.2004	38296	0.135	1.095	0.175	1.057	0.167	1.033	1.284
6.11.2004	38297	0.175	1.057	0.167	1.033	0.186	0.994	1.14
7.11.2004	38298	0.167	1.033	0.186	0.994	0.157	0.949	1.14
8.11.2004	38299	0.186	0.994	0.157	0.949	0.147	0.904	1.158
9.11.2004	38300	0.157	0.949	0.147	0.904	0.136	0.901	2.051
10.11.2004	38301	0.147	0.904	0.136	0.901	0.229	1.987	2.133
11.11.2004	38302	0.136	0.901	0.229	1.987	0.327	2.193	1.784
12.11.2004	38303	0.229	1.987	0.327	2.193	0.5	1.938	1.805
13.11.2004	38304	0.327	2.193	0.5	1.938	0.453	1.779	3.561
14.11.2004	38305	0.5	1.938	0.453	1.779	0.449	2.876	5.587
15.11.2004	38306	0.453	1.779	0.449	2.876	0.446	4.91	3.9
16.11.2004	38307	0.449	2.876	0.446	4.91	0.404	3.238	3.758
17.11.2004	38308	0.446	4.91	0.404	3.238	0.366	2.7	3.558
18.11.2004	38309	0.404	3.238	0.366	2.7	0.303	2.573	3.55
19.11.2004	38310	0.366	2.7	0.303	2.573	0.313	2.498	2.889

Obr. 4: Testovací súbor dát, HT-3, HT-2 a HT-1 sú dáta z meracej stanice Hosťovce o tri, dva a jeden deň dozadu. TB-3, TB-2, TB-1 sú dáta z meracej stanice Turňa nad Bodvou a posledný stĺpec dát je výsledný prietok .

4 Postup výpočtu

Na vytvorenie výpočtového modelu sme použili, programový balík Matlab, ktorý umožňuje kombinovať viacero výpočtových metód, v našom prípade ide o spomínanú kombináciu SVM a GA. Matlab v sebe obsahuje štandardný nástroj na vytvorenie GA, ale v našej práci sme použili nadstavbovú aplikáciu s názvom GA toolbox od Illinois Genetic Algorithms Laboratory, ktorej inštalácia je voľne dostupná. Tak isto boli využité súbory algoritmu SVM (bez nutnosti inštalácie), aby bolo možné natrénovanie dát a následná predikcia touto metódou. Názov tejto nadstavby je LIBSVM (A Library for Support Vector Machines), je tiež voľne dostupná a jej autormi sú Chih-Chung Chang a Chih-Jen Lin.

Po tom ako sme importovali obidve výpočtové metódy, s ktorými budeme pracovať, môžeme vykonať prvú časť úlohy - vložiť dva pripravené dátové súbory s priemernými dennými prietokmi do Matlabu. Prví, teda testovací súbor zo zvoleného obdobia sa meniť nebude, čiže sa iba naimportuje. Druhý súbor so vstupnými dátami, sa ale po importovaní bude deliť na trénovacie a validačné dátové súbory v určitom pomere, pomocou funkcie čo bolo spomenuté v predchádzajúcej kapitole. Na rozdelenie vstupnej matice o velkosti 1453 x 7 sme použili funkciu (obr. 5) s príkazom:

```
[Tren,Cross]=rozdelMatice(data,round(1453*0.8),1);
```

Jednotlivé dáta sa neopakujú (nenachádzajú) v oboch výstupných maticiach (Tren a Cross. Použitím funkcie na rozdelenie vstupnej matice je možné meniť trenovací a validačný súbor pred každým spustením výpočtového cyklu a tým zaručiť jeho unikátnosť. Na validačnom súbore sa po úprave váh overuje kvalita predikcie na vzorkách, ktoré neboli použité v optimalizačnom algoritme. Týmto spôsobom sa snažíme zabrániť preučeniu SVM.

```
function [Bcell,Ccell]=rozdelMatice(A,pocetr1,opakovania)
% vstupne udaje:     A - matica
%                    pocetr1 - pocet riadkov v matici B
%                    opakovania - pocet matic B a C

% predalokovanie bunkovych poli
Bcell=cell(opakovania,1);
Ccell=cell(opakovania,1);


parfor i=1:opakovania
    % urcenie vektoru v1 vyjadrujuceho indexy riadkov matice A pouzitych
    % pre maticu B
    x=1:size(A,1);
    v1=randsample(x,pocetr1);

    % zistenie pozicii nepouzitych riadkov v matici B
    tf=x;
    tf(v1)=0;
    tf=logical(tf);

    % indexy riadkov matice A pouzitych pre maticu C
    v2=x(tf);

    % urcenie matic B a C
    B=A(v1,:);
    C=A(v2,:);

    % priradenie matic B a C do bunkovych poli
    Bcell{i}=B;
    Ccell{i}=C;
end
```

Obr. 5: Funkcia, ktorou rozdeľujeme vstupnú maticu údajov na trénovacie a validačné dáta

Ďalším krokom je stanovenie vhodných parametrov SVM. Prví parameter označený – *s* určuje typ riešenej úlohy, v našom prípade regresie a druhý –*t* typ jadrovej funkcie. Oba sú stabilné, teda sa nemenia. Ostatné parametre –*c* (Cost), –*g* (gamma), –*e* (epsylon) hľadáme použitím GA a ich presnosť je vyjadrená pomocou korelačného koeficientu (obr. 6).

```
-t 2 -s 3 -c 0.1 -g 0.005 -p 0.05 -q 0.031447 0.85155
-t 2 -s 3 -c 0.1 -g 0.01 -p 0.05 -q 0.020427 0.85316
-t 2 -s 3 -c 0.1 -g 0.015 -p 0.05 -q 0.019576 0.85575
-t 2 -s 3 -c 0.1 -g 0.02 -p 0.05 -q 0.019569 0.85844
-t 2 -s 3 -c 0.1 -g 0.06 -p 0.05 -q 0.019801 0.87061
-t 2 -s 3 -c 0.1 -g 0.11 -p 0.05 -q 0.019714 0.87373
-t 2 -s 3 -c 0.1 -g 0.115 -p 0.05 -q 0.019687 0.87414
-t 2 -s 3 -c 0.1 -g 0.12 -p 0.05 -q 0.019647 0.87414
```

Obr. 6: Parametre SVM zapísané v textovom súbore, posledný stĺpec je korelačný koeficient, ktorý určuje ich presnosť

Potom ako sme použitím GA našli parametre pre SVM (obr. 5), uložia sa do textového súboru, z ktorého si ich SVM bude načítavať, počas trénovacieho procesu. Nastavenia GA sa realizujú cez súbor input_sga_maxSpec, ktorý je možné nastavovať podľa potrieb riešenej úlohy a takto upravený ho spúšťame v Matlabe. Nastavenia sa týkajú počtu premenných (v našom prípade 3), s ktorými počítame. Ich hornej a dolnej hranice (veľkosti). Ďalej typ cieľu, minimalizovanie alebo maximalizovanie chyby. Obmedzenia, pre veľkosť chyby a v neposlednom rade základné nastavenia o veľkosti populácie, počte generácií a percentuálny počet vrátených hodnôt.

Po určení parametrov SVM, prichádza k spusteniu samotného trénovacieho procesu príkazom (vzorový príklad uvádzaný v nápovede):

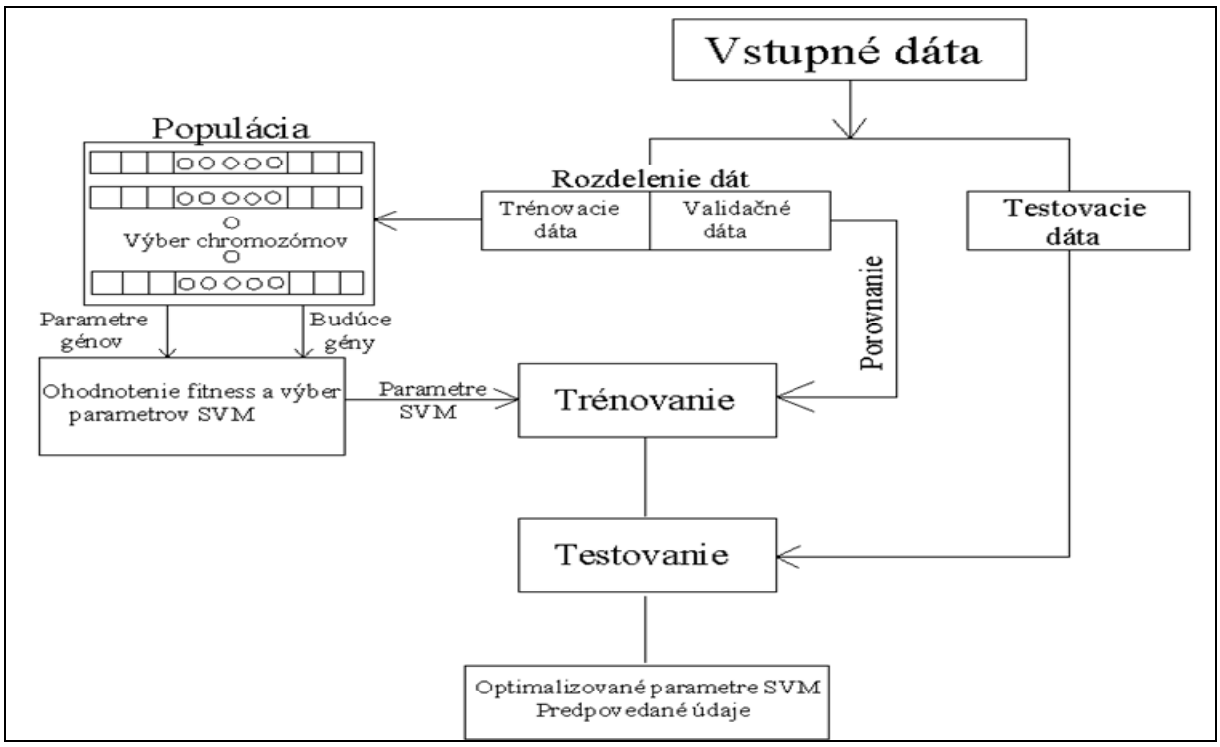
model=svmtrain(training_label_vector,training_instance_matrix [, 'libsvm_options']);
kde:

training_label_vector:
Vektor m x 1 tréningových dát (typ premennej musí byť double)
training_instance_matrix:
Matica m x n s m tréovacími dátami a s n funciami. Môže byť dense alebo sparse (typ premennej musí byť double)
libsvm_options:
Reťazec parametrov pre tréning, v rovnakom formáte ako v LIBSVM

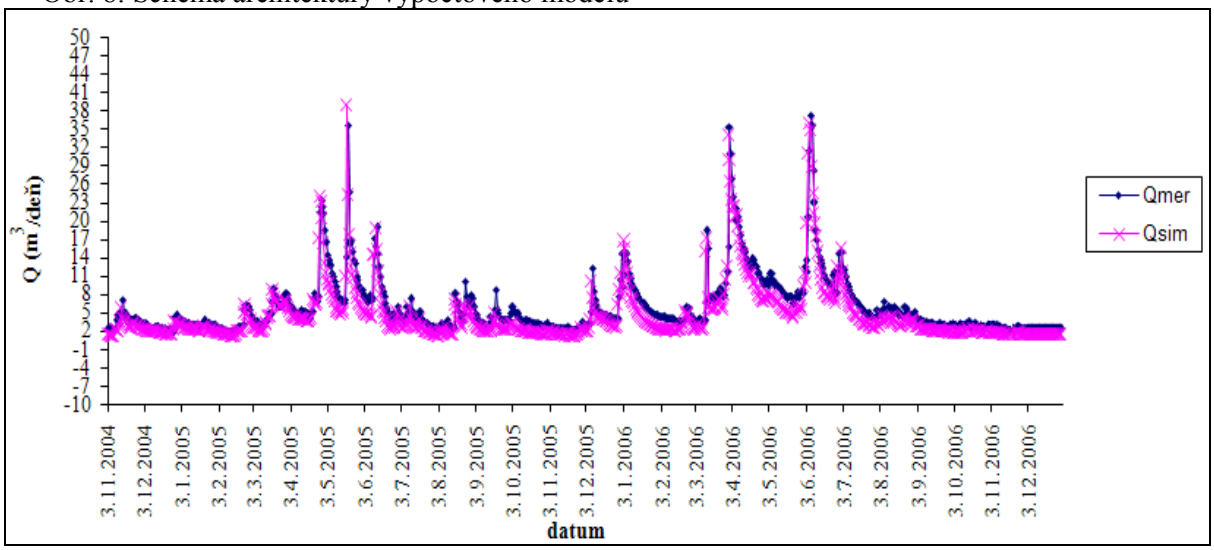
```
function objConst = sgaFitnessFunction(decVars)
% global label_tren;
% global label_test;
% global instance_tren;
% global instance_test;
% global max2;

% delete('m2.txt');
fid = fopen('m2.txt', 'a');
x = decVars;
        c = x(1);
        g = x(2);
        p = x(3);

        options = ['-t 2 ' '-s 3 ' '-c ' num2str(c) ' -g ' num2str(g) ' -p ' num2str(p) ' -q'];

        model = svmtrain(label_tren, full(instance_tren), options);
        [predict_label, accuracy, dec_values] = svmpredict(label_test, instance_test, model);

            eee = corrcoef(label_test,predict_label);

            if eee(1,2) > max2;
                max2 = eee(1,2);
                vysledky = predict_label;
                modelBest = model;  %konecny vypocet
                vystup = [options ' ' num2str(eee(1,2))];
                fprintf(fid, vystup);
                fprintf(fid,'\r\n');
            end
            objConst (1) = eee(1,2);
      fclose(fid);
end
```

Obr. 7: Ukážka kódu, ktorým sme kombinovali výpočtové metódy SVM, GA a realizovali samotnú predpoveď prietokov

Po ukončení trénovacieho procesu sa výsledky predikcie prietokov porovnajú s testovacím súborom a ich presnosť sa vyjadrí pomocou korelačného koeficientu. Parametre SVM spolu s výsledkami sa zapíšu do textového súboru ako výstup z modelu.



Obr. 8: Schéma architektúry výpočtového modelu



Obr. 9: Grafické porovnanie vypočítaných (Qsim) a testovacích (Qmer) údajov

5 Záver

Vývoj počítačovej techniky, najmä v posledných rokoch viedol k rozvoju matematických modelov. Tie sa začínajú využívať aj vo vodnom hospodárstve a ich pomocou sa dajú získať charakteristiky záujmového územia rýchlejšie a s nižšími nákladmi. Presnosť týchto modelov závisí od viacerých faktorov, ale hlavne od kvantifikácie vstupných údajov. Za účelom dosiahnutia spoľahlivých výsledkov je nutné rozvíjať nové metódy modelovania, ktoré prekonajú nedostatky ich predchodcov.

Cieľom článku bolo posúdiť a zhodnotiť využitie hybridného modelu vo vodnom hospodárstve a pri vodohospodárskych výpočtoch.

Na získanie vstupných veličín zadaných boli použité vodomerné zariadenia nachádzajúce sa v záujmovom území. Výpočty boli realizované v prostredí programu Matlab, do ktorého boli importované metódy SVM a GA. Ako vstupné dáta pre NS sme použili údaje o prítoku z dvoch riek vlievajúcich sa do rieky Bodva a údaje o výslednom prietoku. Po rozdelení dát sme na hľadanie optimálnych parametrov SVM, použili GA namiesto metódy pokus-omyl čo viedlo k nájdeniu vhodných parametrov a po skončení trénovania modelu a porovnaní s testovacím súborom dát aj k

uspokojivým výsledkom, čoho dôkazom je grafické porovnanie (obr. 9). Preto je možné použiť túto výpočtovú metódu ako jednu z možných alternatív, ku štandardným výpočtovým metódam.

6 Poďakovanie

Tento článok vznikol vďaka podpore v rámci OP Výskum a vývoj pre projekt: Centrum excelentnosti integrovanej protipovodňovej ochrany územia, ITMS: 2624012000 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja. Práca bola podporovaná tiež Agentúrou na podporu výskumu a vývoja na základe zmluvy č. APVV-0443-07 a LPP-0319-09.

7 Literatúra

[1] Cortes, C., Vapnik N. V. *Support-vector networks. Machine Learning*, 20 (3), 273–297, 1995.
[2] Chang C.C., Lin, C.J. *LIBSVM: A library for support vector machines*. http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2001.
[3] Kvasnička, V. a kol. *Úvod do teórie neurónových sietí. Iris, Bratislava*, ISBN 80-88778-30-1, 1997.
[4] Kohonen, T. *Self-Organizing Maps*. Springer-Verlag, Berlin, 2001.
[5] Smola, J. A, Scholkopf, B. *A tutorial on Support Vector Regresion*, In: NeoroCOLT2 Technical Report Series, NC2-TR-27150, 1998.
[6] Vapnik, N. V. *The Nature of Statistical Learning Theory*, In: Springer-Verlag New York, Inc, 1995.
[7] Vesanto J. and Alhoniemi E. *Clustering of the Self-Organizing Map*. In IEEE Transactions on Neural Networks, Volume 11, Number 3, 586-600, 2000.

Ing. Martin Suchár
martin.suchar@stuba.sk
doc. Ing. Milan Čistý PhD.
milan.cisty@stuba.sk
Katedra vodného hospodárstva krajiny
Stavebná fakulta Slovenskej technickej univerzity v Bratislave
Radlinského 11
813 68 Bratislava