

## Список літератури

1. Berument H. The effects of different inflation risk premium on interest rate spreads / Hakan Berument, Zubeyir Kilinc, Umit Ozlale // *Physica A*. – 2004. – Vol. 333. – P. 317–324.
2. Chornei R. K. Control of Spatially Structured Random Processes and Random Fields with Applications / Ruslan K. Chornei, Hans Daduna, Pavel S. Knopov. – New York : Springer Science + Business Media, Inc., 2006. – 261 p.
3. González-Hernández J. Markov control processes with randomized discounted cost / Juan González-Hernández, Raquel R. López-Martínez, J. Rubén Pérez-Hernández // *Math. Meth. Oper. Res.* – 2007. – Vol. 65, № 1. – P. 27–44.
4. Gil-Alana L. A. Modelling the U.S. interest rate in terms of I(d) statistical models / Luis Alberiko Gil-Alana // *Quarterly Review of Economics & Finance*. – 2004. – Vol. 44, № 4. – P. 475–486.
5. Hernández-Lerma O. Discrete-time Markov control processes: basic optimality criteria / O. Hernández-Lerma, J. B. Lasserre. – Berlin Heidelberg New York : Springer, 1996. – 216 p.
6. Newell R. G. Discounting the distant future: how much do uncertain rates increase valuations? / Richard G. Newell, William A. Pizer // *Journal of Environmental Economics and Management*. – 2003. – Vol. 46, № 1 – P. 52–71.

*N. Kovalchuk., R. Chornei*

## SEMI-MARKOV DECISION-MAKING PROCESSES WITH RANDOMIZED DISCOUNT

*In this paper we consider semi-Markov decision processes with randomized discounted factors. We find sufficient conditions for the existence and uniqueness of optimal stationary nonrandomized strategies for control these processes on finite and infinite horizon.*

**Keywords:** Decision-making processes, semi-Markov processes, randomized discount, optimality equation.

*Матеріал надійшов 07.08.2013*

УДК 519.7

*Галкін О. А.*

## АВТОМАТИЧНИЙ ВИБІР ПАРАМЕТРІВ ЯДРА ОПОРНО-ВЕКТОРНИХ МАШИН

*Досліджено методологію автоматичного вибору параметрів ядра опорно-векторних машин. Використано метод градієнтного спуску для мінімізації оцінок тестової похибки. Отримано гладку апроксимацію тестової помилки з використанням оцінки апостеріорних ймовірностей. Запропоновано та реалізовано алгоритм, де градієнтним кроком є напрямок градієнта в просторі параметрів. Проведено експериментальні дослідження для оцінки ефективності запропонованого методу.*

**Ключові слова:** ядро, метод градієнтного спуску, згладжування оцінок, гладка апроксимація тестової помилки, множина перевірки.

### Вступ

Алгоритм опорно-векторних машин (ОВМ) залежить від кількох параметрів. Один із них, що позначається через  $C$ , контролює відповідність між максимізацією поля та мінімізацією помилки. Інші параметри знаходяться в нелінійному відображенні у простір характеристик. Вони називаються *параметрами ядра*.

Зазвичай, встановлення параметрів здійснюється з використанням мінімаксного підходу, який полягає в максимізації поля по коефіцієнтах гіперплощини та зведенні до мінімуму оцінки помилки узагальнення по множині параметрів ядра. Однак, коли є множина параметрів, класична стратегія, що ґрунтується на методі перебору в просторі параметрів, стає нерозв'язною,

оскільки вона буде відповідати виконанню алгоритму на кожному можливому значенні вектора параметрів (до певної дискретизації). Замість цього ми пропонуємо виконувати такий перебір, використовуючи метод градієнтного спуску.

### Квадратичні втрати

Ми будемо досліджувати слабке поле ОВМ з квадратичною пеналізацією помилок, що можна розглядати як окремий випадок версії жорсткого поля з модифікованим ядром [3].

$$K \leftarrow K + \frac{1}{C}I, \quad (1)$$

де  $I$  є одиничною матрицею, а  $C$  є постійною величиною, що пеналізує навчальні помилки. Надалі ми зосередимось на жорсткому полі ОВМ і використаємо (1) під час аналізу випадку нероздільних даних. Таким чином,  $C$  буде розглядатися як ще один параметр функції ядра.

Такий підхід можна інтуїтивно зрозуміти так [5]:  $n$  додаткових віртуальних компонентів додаються у простір характеристик, а модифіковане відображення визначається так:

$$\tilde{\Phi}(x_i)^T = (\Phi(x_i) \quad 0 \quad \dots \quad y_i / \sqrt{C} \quad \dots \quad 0).$$

1 і n позиція

Вводячи:

$$\tilde{w}^T = (w \quad \xi_1 \sqrt{C} \quad \dots \quad \xi_n \sqrt{C}),$$

ми отримуємо:

$$\tilde{w}^2 = w^2 + C \sum_{i=1}^n \xi_i^2,$$

$$y_i(\tilde{w} \cdot \tilde{\Phi}(x_i) + b) \geq 1 \Leftrightarrow y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i.$$

Важливо підкреслити, що навчання жорсткого поля ОВМ в модифікованому просторі характеристик є еквівалентним навчанням слабкого поля ОВМ з квадратичною пеналізацією слабких змінних. Нове ядро має вигляд:

$$\tilde{K}_y = \tilde{\Phi}(x_i) \cdot \tilde{\Phi}(x_j) = K_y + \delta_{ij} / C.$$

Відзначимо також, що з  $\tilde{w} = \sum \alpha_i^0 y_i \tilde{\Phi}(x_i)$  слідує:

$$\xi_i C = \alpha_i^0.$$

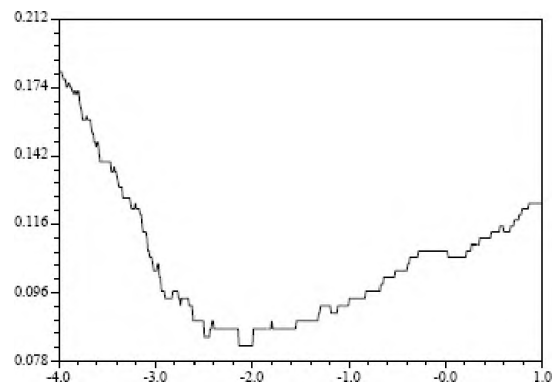
### Згладжування оцінок тестової похибки

Оцінка продуктивності ОВМ з використанням помилки перевірки ( $T = \frac{1}{p} \sum_{i=1}^p \Psi(-y'_i f(x'_i))$ ), або ж помилки пропуску

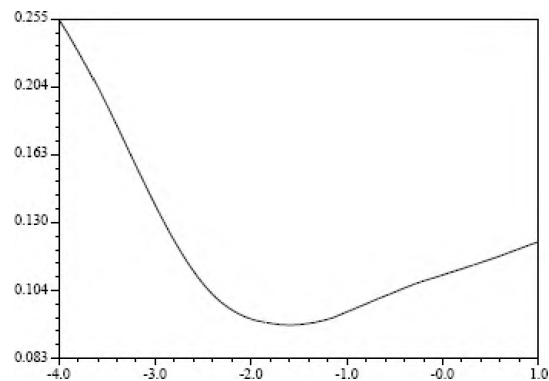
$$(L(x_1, y_1, \dots, x_n, y_n) = \sum_{p=1}^n \Psi(-y_p f^p(x_p))) \quad \text{вимагає}$$

використання функції кроку  $\Psi$ . Однак ми будемо використовувати *метод градієнтного спуску* для мінімізації оцінок тестової похибки. На жаль, функція кроку не є диференційованою, однак існує можливість обмежити  $\Psi(x-1)$  по  $x$  для  $x \geq 0$ . Наслідком є отримання обмеження  $R^2 / M^2$  з помилки пропуску. Незважаючи на можливість обмеження  $\Psi(x-1)$  по  $x$  ( $x \geq 0$ ), великі помилки трапляються частіше одного разу, тому кориснішим є використання функції узгодження такого вигляду (рис. 1):

$$\Psi(x) = (1 + \exp(-Ax + B))^{-1}.$$



1



2

Рис. 1. Помилка перевірки для різних значень ширини (в логарифмічній шкалі) радіальної функції ядра: 1 – функція кроку  $\Psi(x) = 1_{x>0}$ ; 2 – сигмоїдна функція

$$\Psi(x) = (1 + \exp(-5x))^{-1}$$

Однак вибір постійних величин  $A$  та  $B$  є непростю задачею. Якщо значення  $A$  є занадто малим, тоді оцінка не є точною, якщо ж значення  $A$  є занадто великим, отримана оцінка не є гладкою.

Замість того, щоб підбирати відповідні постійні величини  $A$  та  $B$ , можна спробувати безпо-

середньо отримати гладку апроксимацію тестової помилки, оцінюючи апостеріорні ймовірності. Нещодавно запропоновано таку оцінку апостеріорного розподілу  $P(Y=1|X=x)$  виходу ОВМ  $f(x)$  [6]:

$$\tilde{P}_{A,B}(x) = \tilde{P}(Y=1|X=x) = \frac{1}{1 + \exp(Af(x) + B)},$$

де  $f(x)$  є виходом ОВМ. Постійні величини  $A$  та  $B$  знаходяться при мінімізації відстані Кульбака – Лейблера між  $\tilde{P}$  та емпіричною оцінкою  $P$ , що будується з множини перевірки  $(x'_i, y'_i)_{i \leq n}$ :

$$(A^*, B^*) = \arg \max_{A,B} \sum_{i=1}^{n_x} \left( \frac{1+y'_i}{2} \log(\tilde{P}_{A,B}(x'_i)) + \frac{1-y'_i}{2} \log(1 - \tilde{P}_{A,B}(x'_i)) \right). \quad (2)$$

Ця оптимізація здійснюється за допомогою алгоритму градієнтного спуску другого порядку [6].

Відповідно до цієї оцінки, найкращий поріг для нашого класифікатора ОВМ  $f$  є такий, що:

$$f(x) = \text{sgn}(\tilde{P}_{A^*,B^*}(x) - 0.5).$$

Варто зауважити, якщо  $B^* \neq 0$ , ми отримуємо певну корекцію, порівняно зі звичайним порогом ОВМ. Відповідно до визначення, помилкою узагальнення нашого класифікатора є:

$$P(Y \neq f(X)) = \int_{x, f(x)=1} P(Y=1|x) d\mu(x) + \int_{x, f(x)=-1} (Y=-1|x) d\mu(x).$$

Ця помилка може бути емпірично оцінена таким чином:

$$P(Y \neq f(X)) \approx \sum_{i, \tilde{P}(x'_i) < 0.5} \tilde{P}(x'_i) + \sum_{i, \tilde{P}(x'_i) > 0.5} 1 - \tilde{P}(x'_i) = \sum_{i=1}^{n_x} \min(\tilde{P}(x'_i), 1 - \tilde{P}(x'_i)).$$

Варто звернути увагу на те, що мітки на множині перевірки не використовуються на останньому кроці безпосередньо, а використовуються опосередковано через оцінки постійних величин  $A$  та  $B$ , що входять у параметричну форму  $\tilde{P}_{A^*,B^*}$ . Для кращого розуміння цієї оцінки розглянемо граничний випадок, коли помилки на множині перевірки немає. Алгоритм максимальної ймовірності призведе до того, що  $A = -\infty$ , а  $\tilde{P}_{A^*,B^*}(x)$  буде приймати лише двійкові значення. Як наслідок, оцінка ймовірнісної помилки дорівнюватиме нулю.

Якщо множина перевірки є недоступною, для знаходження значень  $A^*$  та  $B^*$  можна викорис-

товувати навчальну множину, мінімізуючи функціонал (2), однак це рівнозначно оцінці згладженої навчальної помилки. Прийнятним способом оцінки згладженої помилки узагальнення є використання процедури пропуску: знайти  $A^*$  та  $B^*$ , які мінімізують (2) з  $x'_p = x_p, y'_p = y_p$  та модифіковану версію апостеріорної ймовірності:

$$\tilde{P}_{A,B}(x_p) = \frac{1}{1 + \exp(Af^p(x_p) + B)},$$

де  $f^p$  є функцією навчання за умови, якщо точка  $x_p$  була видалена з навчальної множини.  $f^p(x_p)$  може бути точно оцінена, використовуючи рівняння:

$$t_n = \frac{1}{n} L(x_1, y_1, \dots, x_n, y_n) = \frac{1}{n} \text{Card}\{p : \alpha_p^0 S_p^2 \geq y_p f^0(x_p)\}, \text{ як:}$$

$$f^p(x_p) \approx f^0(x_p) - y_p \alpha_p^0 S_p^2.$$

Іншим методом оцінки апостеріорних ймовірностей є використання оцінювача щільності вікна Парцена ( $dP_{est}(x, y) = \frac{1}{n} \sum_{i=1}^n dP_{x_i}(x) \delta_{y_i}(y)$ ), що

використовується для обчислення локального ризику. Даний метод призводить до мінімізації наступного інтеграла:

$$\int \frac{1+y}{2} \log(\tilde{P}_{A,B}(x)) + \frac{1-y}{2} \log(1 - \tilde{P}_{A,B}(x)) dP_{est}(x, y).$$

Недоліком такого методу є те, що вибір ширини оцінювача вікна Парцена має вирішальне значення, а її знаходження має здійснюватись за допомогою перевірки або процедури пропуску.

### Оптимізація параметрів ядра

Беручи за основу алгоритм ОВМ, ми пропускаємо, що ядро  $K$  залежить від одного чи декількох параметрів, закодованих у вектор  $\theta = (\theta_1, \dots, \theta_n)$ . Отже, розглянемо клас функцій рішення, що параметризовані  $\alpha, b$  та  $\theta$ :

$$f_{\alpha,b,\theta}(x) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i K_{\theta}(x, x_i) + b\right).$$

Нашою метою є вибір таких значень параметрів  $\alpha$  та  $\theta$ , при яких значення  $W$

$$(W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j)$$

є максимальним (алгоритм максимального поля), а значення  $T$  (критерій вибору моделі) зводиться до мінімуму (найкращі параметри ядра). Тобто для фіксованого  $\theta$ , ми хочемо отримати  $\alpha^0 = \arg \max W(\alpha)$

та вибрати таке  $\theta^0$ , що:

$$\theta^0 = \arg \min_{\theta} T(\alpha^0, \theta).$$

У випадку, коли  $\theta$  є одновимірним параметром, зазвичай, мають справу з кінцевим числом значень, з яких вибирають те значення, яке дає найменше значення критерію  $T$ . Однак, слід зауважити, що коли  $T$  та рішення ОВМ є неперервними відносно  $\theta$ , існує ефективніший підхід [4]: використовуючи алгоритм інкрементної оптимізації, можна навчати ОВМ з невеликими зусиллями, коли  $\theta$  змінюється на невелику величину. Однак, як тільки  $\theta$  буде мати більше одного компонента, обчислення  $T(\alpha, \theta)$  для кожного можливого значення  $\theta$  стає нерозв'язним, тому важливо знайти спосіб оптимізації  $T$  вздовж траєкторії в просторі параметрів ядра.

Використання градієнта як критерію вибору моделі для оптимізації параметрів моделі було запропоновано в [1] та продемонстровано для випадку лінійної регресії та прогнозування часових рядів. Крім того, дане використання було запропоновано для оптимізації параметрів регуляризації нейронної мережі.

Далі ми пропонуємо алгоритм, який є альтернативою до оптимізації ОВМ, де градієнтним кроком є напрямок градієнта  $T$  у просторі параметрів. Це може бути досягнуто з використанням такої ітераційної процедури:

1. Ініціалізація  $\theta$  до деякого значення.
2. Використання стандартного алгоритму ОВМ для знаходження максимуму квадратичної форми  $W$  :

$$\alpha^0(\theta) = \arg \max_{\alpha} W(\alpha, \theta).$$

3. Обновлення параметрів  $\theta$  таким чином, щоб  $T$  було зведено до мінімуму. Це, зазвичай, досягається за допомогою градієнтного кроку.
4. Повернення до кроку 2 або зупинка, якщо досягнуто мінімуму  $T$ .

Рішення кроку 3 вимагає оцінки того, як  $T$  варіюється з  $\theta$ . Таким чином, ми обмежимося випадком, коли величина  $K_{\theta}$  може бути диференційована відносно  $\theta$ . Крім того, ми будемо розглядати тільки випадки, коли може бути обчислений (або наближений) градієнт  $T$  відносно  $\theta$ .

Зауважимо, що  $\alpha^0$  неявно залежить від  $\theta$ , оскільки  $\alpha^0$  визначається як максимум  $W$ . Якщо ми маємо  $k$  параметрів ядра  $(\theta_1, \dots, \theta_k)$ , повною

похідною від  $T^0(\cdot) \equiv T(\alpha^0(\cdot), \cdot)$  відносно  $\theta_p$  є:

$$\frac{\partial T^0}{\partial \theta_p} = \frac{\partial T^0}{\partial \theta_p} \Big|_{\alpha^0 \text{ фікс.}} + \frac{\partial T^0}{\partial \alpha^0} \cdot \frac{\partial \alpha^0}{\partial \theta_p}.$$

Обчисливши градієнт  $\nabla_{\theta} T^0$ , способом виконання кроку 3 є застосування *градієнтного кроку*:

$$\delta \theta = -\varepsilon \nabla_{\theta} T^0,$$

для деяких малих та в кінцевому підсумку зменшуваних  $\varepsilon$ . Збіжність може бути покращено з використанням похідних другого порядку (метод Ньютона):

$$\delta \theta = -(\Delta_{\theta} T^0)^{-1} \nabla_{\theta} T^0,$$

де оператор Лапласа  $\Delta$  визначається таким чином:

$$(\Delta_{\theta} T^0)_{i,j} = \frac{\partial^2 T^0}{\partial \theta_i \partial \theta_j}.$$

У такому формулюванні додаткові обмеження можуть бути накладені через проєкцію градієнта.

### Експериментальні дослідження й аналіз результатів

Експериментальні дослідження було проведено для оцінки ефективності та придатності нашого методу. Експеримент полягав в автоматичному знаходженні оптимального значення двох параметрів: ширини радіального ядра та постійної величини  $C$  в рівнянні (1). В основі методології, яка безпосередньо використовувалась в експерименті, був алгоритм градієнтного спуску, реалізацію якого здійснено з використанням оптимізаційного інструментарію пакету Matlab. Даний інструментарій включає оновлення другого порядку для покращення швидкості збіжності. Оскільки ми не були зацікавлені в точному значенні параметрів мінімізації функціонала, був використаний вільний критерій зупинки алгоритму.

У даному експерименті ми намагались автоматично вибрати ширину  $\sigma$  радіального ядра

$$K(x, y) = \exp\left(-\sum_{i=1}^n \frac{(x_i - y_i)^2}{2n\sigma^2}\right)$$

вздовж постійної величини  $C$ , яка пеналізує помилку навчання, що входить у рівняння (1).

Для того, щоб уникнути додавання позитивних обмежень у задачі оптимізації (для постійної величини  $C$  та ширини  $\sigma$  радіального ядра), ми використали параметризацію  $\theta = (\log C, \log \sigma)$ ,

що забезпечує стабільнішу оптимізацію. Початковими значеннями були  $C=1$  та  $\log \sigma = -2$ . Кожен компонент нормалізувався своїм стандартним відхиленням, що відповідає досить малому значенню для  $\sigma$ .

Ми використовували тестові бази даних, представлені в [7], а також слідували тій експериментальній процедурі, що наведена в [7]. На кожній з перших 5 навчальних множин параметри ядра оцінювались з використанням 5-кратної перехресної перевірки, мінімізації  $R^2 / M^2$  та інтервального обмеження. Врешті параметри ядра обчислювались як середнє значення з 5 оцінок. Результати представлено в таблиці 1.

У другому стовпці представлено результати з [7]. У третьому та четвертому стовпці, параметри знайдено шляхом мінімізації  $R^2 / M^2$  та з використанням алгоритму градієнтного спуску відповідно.

Зауважимо, що мінімізація  $R^2 / M^2$  та інтервальні оцінки забезпечують отримання приблизно однакових характеристик, що є так званим підбиранням параметрів, які мінімізують помилку перехресної перевірки. Цей результат є передбачуваним, оскільки перехресна перевірка є точним методом вибору гіперпараметрів будь-якого алгоритму навчання.

Більш цікавим порівнянням є обчислювальні витрати цих методів. Таблиця 2 демонструє, скільки навчань ОВМ необхідно в середньому для вибору параметрів ядра на кожному розриві. Результати для перехресної перевірки представлені в [7], де було застосовано 10 різних значень

для  $C$  і  $\sigma$ , та проведено 5-кратну перехресну перевірку. На кожній з 5 навчальних множин, кількість навчань ОВМ, необхідних для цього методу, є  $10 \times 10 \times 5 = 500$ .

Отриманий результат оцінки складності є вражаючим: у середньому в 100 разів менше ітерацій навчання ОВМ необхідно для знаходження параметрів ядра. Основною причиною цього результату є те, що в оптимізації брали участь два параметри. Через обчислювальні складнощі метод перебору перехресної перевірки не забезпечує вибір більш ніж 2 параметрів, тоді як запропонований нами метод забезпечує такий вибір.

### Висновки

Зауважимо, що враховуючи результати з [2],  $R^2 / M^2$  може здатися грубою верхньою межею інтервального обмеження, що є точною оцінкою тестової помилки. Однак у процесі вибору параметрів ядра важливим є те, щоб отримати обмеження, мінімум якого є близьким до оптимальних параметрів ядра. Навіть якщо значення  $R^2 / M^2$  не може бути використано для оцінки тестової помилки, попередній експеримент демонструє, що його мінімізація забезпечує непогані результати. Помилка узагальнення, отримана при мінімізації інтервального обмеження, є ненабагато кращою. Оскільки мінімізацію помилки узагальнення здійснити та контролювати (багато локальних мінімумів) важче, на практиці ефективніше проводити мінімізацію  $R^2 / M^2$ .

Таблиця 1. Тестові помилки, знайдені різними алгоритмами вибору параметрів  $C$  та  $\sigma$  ОВМ

№ прикладу	Перехресна перевірка	$R^2 / M^2$	Інтервальне обмеження
приклад 1	26,04±4,74	26,84±4,741	25,59±4,18
приклад 2	23,53±1,73	23,25±1,7	23,19±1,67
приклад 3	15,95±3,26	15,92±3,18	16,13±3,11
приклад 4	4,80±2,19	4,62±2,03	4,56±1,97
приклад 5	22,42±1,02	22,88±1,23	22,5±0,88

Таблиця 2. Середня кількість навчань ОВМ з однієї навчальної множини, необхідних для вибору параметрів  $C$  та  $\sigma$  з використанням стандартної перехресної перевірки, мінімізації  $R^2 / M^2$  та інтервального обмеження

№ прикладу	Перехресна перевірка	$R^2 / M^2$	Інтервальне обмеження
приклад 1	500	14,2	7
приклад 2	500	12,2	9,8
приклад 3	500	9	6,2
приклад 4	500	3	11,6
приклад 5	500	6,8	3,4

## Список літератури

1. Bengio Y. Gradient-based optimization of hyper-parameters / Y. Bengio // *Neural Computation*. – 2000. – Vol. 12(8).
2. Chapelle O. Model selection for support vector machines / O. Chapelle, V. Vapnik // *Advances in Neural Information Processing Systems*. – 1999.
3. Cortes C. Support vector network / C. Cortes, V. Vapnik // *Machine learning*. – 20: 1–25. – 1995.
4. Cristianini N. Dynamically adapting kernels in support vector machines / N. Cristianini, C. Campbell, J. Shawe-Taylor // *Advances in Neural Information Processing Systems*. – 1999.
5. Freund Y. Large margin classification using the perceptron algorithm / Y. Freund, R. E. Schapire // *Computational Learning Theory*. – 1998. – P. 209–217.
6. Platt J. Probabilities for support vector machines / J. Platt // *Advances in Large Margin Classifiers* / ed. A. Smola, P. Bartlett, B. Schölkopf, D. Schurmans. – Cambridge, MA : MIT Press, 2000.
7. Rätsch G. Soft margins for AdaBoost / G. Rätsch, T. Onoda, K.-R. Müller // *Machine Learning*. – 2001. – 42(3). – P. 287–320.

O. Galkin

## AUTOMATIC SELECTION OF THE PARAMETERS OF THE SUPPORT VECTOR MACHINES KERNEL

*Methodology of the automatic selection of the support vector machines kernel is investigated. Gradient descent approach is used for minimizing test error estimates. A smooth approximation of the test error is obtained by estimating posterior probabilities. An algorithm is proposed and implemented where a gradient step is the direction of the gradient in the parameters space. Experiments have been carried out to assess the performance and feasibility of the proposed method.*

**Keywords:** kernel, gradient descent approach, smooth approximation of the test error, validation set.

Матеріал надійшов 22.10.2012

УДК: 004.4

Галковська Л. О.

## АЛГОРИТМИ РОЗВ'ЯЗАННЯ РОЗПОДІЛЕНОЇ ЗАДАЧІ ЗАДОВОЛЕННЯ ОБМЕЖЕНЬ

*У даній роботі представлено розроблену автором класифікацію алгоритмів розв'язання розподіленої задачі задоволення обмежень. Наведено приклади представників кожного з класів, а також їх порівняльна характеристика та рекомендації щодо їх застосування для розв'язання конкретних DCSP задач.*

**Ключові слова:** задача задоволення обмежень, розподілена задача задоволення обмежень, алгоритми локального пошуку, алгоритми конструктивного пошуку, гібридні алгоритми.

Багато задач оптимізації та пошуку можуть бути формалізовані як задачі задоволення обмежень (Constraint Satisfaction Problems-CSP). Прикладом такої задачі може слугувати відома задача розфарбування графа. З поширенням концепції розподілених обчислень сформувався підклас розподілених CSP задач (Distributed Constraint

Satisfaction Problem-DCSP). У багатьох випадках для їх розв'язання залучають групу незалежних виконавців (наприклад, агентів). Кожен агент оперує частиною задачі (частіше за все опікується однією змінною з задачі або однією підзадачею) і обмінюється інформацією з іншими виконавцями, з якими він пов'язаний певними умовами