Метод опорних векторів (SVM)

О. І. ШЕРЕМЕТ,О. В. САДОВОЙ***

*Донбаська державна машинобудівна академія
**Дніпродзержинський державний технічний університет

У статті розглядається метод опорних векторів (SVM), який є машинним алгоритмом, що навчається на прикладах та використовується для класифікації об'єктів. Встановлено, що SVM може бути успішно застосований для керування складними електромеханічними системами, він може забезпечити адаптивність алгоритмів керування, виконувати функції спостерігача, ідентифікатора невідомих параметрів, деякої еталонної моделі, за його допомогою можна керувати складними нелінійними об'єктами, а також об'єктами зі стохастичними параметрами.

В статье рассматривается метод опорных векторов (SVM), который является машинным алгоритмом, обучающимся на примерах, и используется для классификации объектов. Установлено, что SVM может быть успешно применен для управления сложными электромеханическими системами, он может обеспечить адаптивность алгоритмов управления, выполнять функции наблюдателя, идентификатора неизвестных параметров, некоторой эталонной модели, с его помощью можно управлять сложными нелинейными объектами, а также объектами со стохастическими параметрами.

The article describes a method of support vectors (SVM), which is a machine algorithm trained on examples, and used to classify objects. Established that the SVM can be used successfully for the management of complex electromechanical systems, it can provide adaptive control algorithms, can function observer identifier unknown parameters of a reference model, it can be used to manage the complex non-linear objects as well as objects with stochastic parameters.

Вступ. Метод опорних векторів, відомий в англомовній літературі [1] як support vector machine (SVM), є машинним алгоритмом, котрий навчається на прикладах та використовується для класифікації об'єктів. Наприклад, SVM може розрізнити аварійний режим роботи електромеханічної системи та класифікувати його за наявності попередніх досліджень, можливих за технологічними вимогами режимів роботи. Такий підхід розкриває значні можливості для побудування адаптивних систем автоматичного керування.

В основі SVM лежить деяка математична сутність – алгоритм максимізації деякої математичної функції відносно наявного набору даних. Для розуміння того, як працює SVM, потрібно мати уявлення про чотири ключові поняття:

– відділяюча гіперплощина (the separating hyperplane);

– гіперплощина максимальної межі (the maximum-margin hyperplane);

– м'яка межа (the soft margin);

– функція ядра (the kernel function).

Відділяюча гіперплощина є математичною сутністю, що відділяє між собою класи об'єктів з однаковими ознаками. Наприклад, так як це показано на *рис. 1*, де у тривимірному просторі площина відділяє кульки світлого кольору від темних кульок.

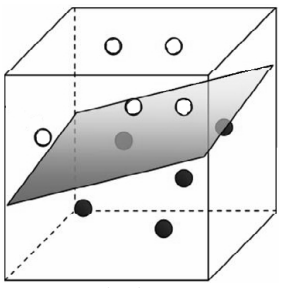

Рис. 1. Приклад відділяючої площини

Можна екстраполювати цю процедуру математично до вимірів, значно вищих за третій. Загальний термін для лінії, котра відділяє елементи різних класів, – багатовимірна гіперплощина.

Спосіб, яким можна провести відділяючу гіперплощину за методом SVM, не є унікальним. Завжди існує багато різних можливостей розташування гіперплощини (*рис. 2*).

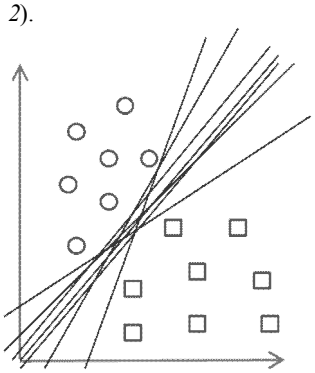

Рис. 2. Можливі варіанти розташування гіперплощини у двохвимірному просторі

Постановка задачі. SVM відрізняється від інших гіперплощинних методів класифікації тим, що він дозволяє обирати оптимальне розташування гіперплощини. Гіперплощина обирається таким чином, щоб бути розташованою на максимальній відстані від елементів кожного з класів, тобто посередині деякої зони, що відділяє між собою ці елементи (на рис. 3 граничні елементи заретушовані). В цьому полягає сутність другого ключового поняття – гіперплощина максимальної межі.

Метою даної роботи є дослідження можливостей математичного апарату, що надається методом опорних векторів, та виконання його критичного аналізу.

Результати роботи. Об'єкти, що класифікуються, не завжди можуть бути розділені гіперплощиною. У реальних системах будуть наявними похибки в даних,

внаслідок яких гіперплощина не виконає розподіл абсолютно точно (*рис. 4*).

Тому для роботи методу SVM вводять допустиму похибку класифікації, що називається м'якою межею.

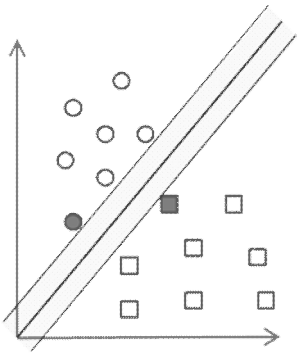

Рис. 3. Розташування гіперплощини максимальної межі

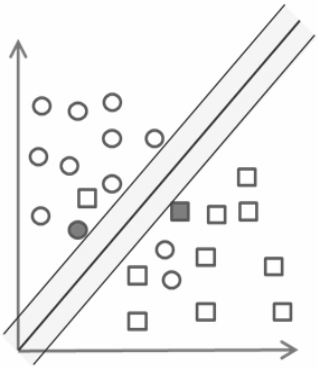

Рис. 4. Можливі похибки під час класифікації об'єктів

Звісно, що метод SVM не повинен враховувати забагато похибок класифікації об'єктів, тому потрібно вводити додатковий параметр, котрий встановлює скільки невірно класифікованих об'єктів можуть перетинати гіперплощину максимальної межі і як далеко вони можуть розташовуватись відносно неї. Таким чином, вводиться так звана м'яка межа похибки навколо гіперплощини.

Об'єкти, що класифікуються, можуть бути поділені лінійно лише в окремих випадках. Здебільшого вони не є такими, що допускають лінійне розподілення. Для вирішення проблеми лінійного розподілення використовують функції ядра, що проектують дані з низьковимірного простору у багатовимірний. При вірному виборі функції ядра об'єкти можуть бути розділені лінійно гіперплощиною у багатовимірному просторі. Таким чином, функції ядра виконують роль спрямляючого простору.

Графічний приклад переходу від задачі, що не має лінійного розподілення об'єктів, до такої, яка дозволяє побудування гіперплощини максимальної межі, наведено на *рис. 5.*

Розглянемо головні математичні залежності, на базі яких працює SVM, та поставимо типову задачу класифікації. Кожен з об'єктів класифікації розглядається як вектор у n-вимірному просторі.

Кожна координата вектора – це деяка ознака, і вона тим більша, чим більше ця ознака відображена у даного об'єкта. Чим менше ця координата, тим менше ознака відповідає об'єкту.

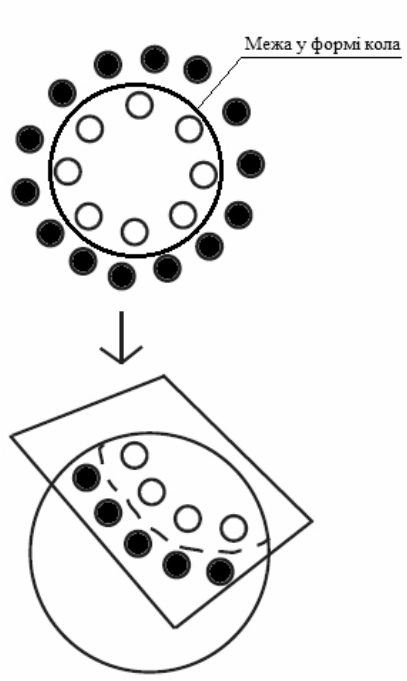

Рис. 5. Приклад спрямлення простору

Під час навчання SVM користуються учбовими колекціями, тобто множиною векторів $(x_1, x_2, \ldots x_n)$, які належать до гіперповерхні R^d та чисел $(y_1, y_2, \ldots y_n)$, значення яких належать до множини $\{-1, 1\}$. Причому число y_i дорівнює 1, якщо відповідний йому вектор x_i належить до категорії, що розглядається, та –1 – у протилежному випадку. Нехай є деяка гіперплощина, що розділяє позитивні та негативні приклади. Тоді точки x, що лежатимуть на цій площині, будуть задовольняти умові [2]

$$w \cdot x + b = 0,$$

де w – нормальний до гіперплощини вектор.

Перпендикуляр, що визначає відстань від гіперплощини до початку координат, визначиться як $|b|/\|w\|$, де вираз $\|w\|$ називають Евклідовою нормою або довжиною [3] вектора w.

Позначимо найкоротшу відстань від відділяючої гіперплощини до найближчого "позитивного" прикладу d_+, а до найближчого "негативного" – d_-. Тоді межа навколо гіперплощини матиме ширину $(d_+ + d_-)$. У тому випадку, коли задача є лінійно розділимою, SVM шукає відділяючу гіперплощину з максимальною межею (таким чином, щоб відстань між елементами, які належать до класу та тими елементами, що не належать до нього, була найбільшою).

Тоді розташування векторів у лінійно розділимій задачі можна описати наступною системою нерівностей [4]

$$\begin{cases} x_i w + b \geq +1 \ \text{ для } \ y_i = +1, \\ x_i w + b \leq -1 \ \text{ для } \ y_i = -1. \end{cases} \quad (1)$$

Систему нерівностей (1) можна спростити до вигляду

$$y_i(x_i w + b) - 1 \geq 0 \ \ \forall i . \quad (2)$$

Межі навколо гіперплощини максимальної межі також являють собою гіперплощини, а саме – гіперплощину H_1, що описується рівнянням $x_i w + b = 1$, та гіперплощину H_2 з рівнянням $x_i w + b = -1$. Ці гіперплощини будуть паралельними одна до одної та матимуть один нормальний вектор w. Відстань від площини H_1 до початку координат буде $|1 - b| / \|w\|$, а від гіперплощини H_2 до початку координат – становитиме $|-1 - b| / \|w\|$. Відповідно, значення $d_+ = d_- = 1/\|w\|$, а ширина максимальної межі визначиться як $d_+ + d_- = 2/\|w\|$. Таким чином, при умові того, що задача є лінійно розділимою, виконується пошук пари гіперплощин, котрі розташовані одна від одної на максимальній відстані, для чого мінімізують $\|w\|$. Гіперплощина максимальної межі буде розташовуватись посередині на рівній відстані від H_1 і H_2 паралельно до них.

Лінійне розділення точок за зазначеним вище принципом для двовимірної задачі наведено на рисунку 6. Точки, що лежать у гіперплощинах H_1 та H_2, та виключення яких з прикладів призвело б до зміни положення гіперплощини максимальної межі, називають опорними векторами (на *рис. 6* вони обведені колами).

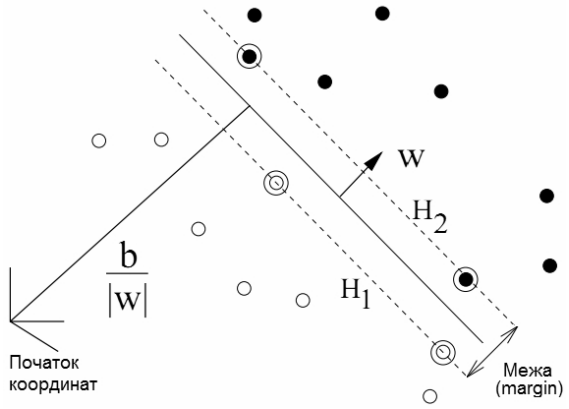

Рис. 6. Лінійне розділення точок

Відомо, що для знаходження мінімуму функції потрібно дослідити її похідну. У випадку мінімізації $\|w\|$ задача ускладнюється тим, що задані лінійні обмеження, які слід враховувати при мінімізації. Множина точок, які задовольняють обмеженням, в n-вимірному просторі являє собою багатогранник: простір ділиться гіперплощинами або напівгіперплощинами у залежності від того, стоїть в обмеженнях знак рівності чи нерівності. Пошук мінімуму відбувається в цьому обмеженому просторі.

Розв'язати задачу мінімізації в обмеженому просторі можна за допомогою метода Лагранжа [5]. Лагранж звів задачу пошуку умовного мінімуму до задачі мінімізації без обмежень, щоб потім скористатись стандартним методом пошуку мінімуму функції. Для використання метода Лагранжа потрібно змінити функцію, що підлягає мінімізації.

Для формування нової функції потрібно ввести множники Лагранжа α_i, $i = 1, 2, \ldots, n$ – один для кожного елементу нерівності (2). лагранжіан має наступний вигляд:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{n} \alpha_i . \quad (3)$$

Тепер потрібно мінімізувати лагранжіан (3) відносно w, b, одночасно вимагаючи, щоб похідні відносно всіх α_i дорівнювали нулю для $\alpha_i \geq 0$. Ці вимоги призводять до необхідності виконання наступних умов (lagrangian trick) [6]:

$$w = \sum_i \alpha_i y_i x_i , \quad (4)$$

$$\sum_i \alpha_i y_i = 0 . \quad (5)$$

Тоді, з урахуванням формул (4) та (5) вираз (3) можна представити у вигляді

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j . \quad (6)$$

У формулі (3) та (6) для позначення лагранжіана використовуються різні індекси: у формулі (3) – індекс "P" (primal – головний), а у формулі (6) – індекс "D" (dual – подвійний). Ці два вирази дещо не співпадають: вони формуються на основі тієї ж оптимізаційної функції, проте з різними обмеженнями, та розв'язок знаходиться шляхом мінімізації L_P або максимізації L_D. Якщо задача оптимізації формулюється таким чином, що $b = 0$, тобто гіперплощини проходять через початок координат, тоді обмеження (5) не з'являється.

У наведених рівняннях точки, для яких $\alpha_i > 0$ називаються опорними векторами, вони лежать на одній з гіперплощин H_1 або H_2. Для всіх інших точок $\alpha_i = 0$. Для віртуальних машинних методів класифікації, що навчаються за таким принципом, опорні вектори є критичними точками навчальної множини. Якщо інші точки будуть змінюватись або пересуватись у просторі, не зачіпаючи опорні вектори, то результат навчання (відділяюча гіперплощина) не буде змінюватись.

Під час оптимізації у просторі з обмеженнями застосовують умови Каруша-Куна-Таккера (Karush-Kuhn-Tucker) або KKT, які є узагальненням метода множників Лагранжа. В теорії оптимізації умови KKT – це необхідні умови розв'язку задач нелінійного програмування. Щоб розв'язок був оптимальним, повинні виконуватись умови (7) – (11) для лагранжіана L_P.

$$\frac{\partial}{\partial w_\nu} L_P = w_\nu - \sum_i \alpha_i y_i x_{i\nu} = 0 \quad \nu = 1, \ldots, d , \quad (7)$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0 , \quad (8)$$

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad i = 1, \ldots, n , \quad (9)$$

$$\alpha_i \geq 0 \quad \forall i , \quad (10)$$

$$\alpha_i (y_i (w \cdot x_i + b) - 1) = 0 \quad \forall i . \quad (11)$$

Таким чином, оптимізація за KKT у випадку лінійно розділимої задачі буде являти собою один з варіантів категоризації за методом SVM.

Під час виконання категоризації реальних даних можуть виникати похибки, котрі призводитимуть до неможливості лінійного розподілення. Для усунення впливу похибок вводяться так звані коефіцієнти вартості $\xi_i, i = 1, ..., n$ у нерівності обмежень, які після цього приймають наступний вигляд (при цьому $\xi_i \geq 0 \ \forall i$):

$$\begin{cases} x_i w + b \geq +1 - \xi_i & \text{для } y_i = +1, \\ x_i w + b \leq -1 + \xi_i & \text{для } y_i = -1. \end{cases}$$

Коли виникає похибка, тоді відповідний коефіцієнт вартості ξ_i змінює праву частину нерівності і вона починає відрізнятись від $+1$ або -1. Тоді $\sum_i \xi_i$ буде верхньою межею значення похибки навчання методу. Логічно змінити функцію, що підлягає мінімізації з $\frac{1}{2}\|w\|^2$, до вигляду

$$\frac{1}{2}\|w\|^2 + C\left(\sum_i \xi_i\right)^k,$$

де C – параметр, який обирається користувачем, збільшення C робить вимоги до точності більш жорсткими.

Головний лагранжіан L_P при наявності коефіцієнтів вартості ξ_i представляється наступним чином:

$$L_P = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i,$$

де μ_i – множники Лагранжа, що вводяться для підсилення позитивних значень коефіцієнтів ξ_i.

Для виконання вимог KKT та лагранжіана L_P при умові зміни значення i від 1 до кількості точок n, що приймають участь у навчанні метода, а значення ν – від 1 до розмірності простору даних d. Тоді одержимо такі умови KKT при категоризації з похибкою:

$$\frac{\partial L_P}{\partial w_\nu} = w_\nu - \sum_i \alpha_i y_i x_{i\nu} = 0, \qquad (12)$$

$$\frac{\partial L_P}{\partial b} = -\sum_i \alpha_i y_i = 0,$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0,$$

$$y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0,$$

$$\xi_i \geq 0,$$

$$\alpha_i \geq 0,$$

$$\mu_i \geq 0,$$

$$\alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} = 0, \qquad (13)$$

$$\mu_i \xi_i = 0. \qquad (14)$$

Як і в попередніх визначеннях, значення b можна розрахувати з рівнянь (13) та (14). Комбінуючи рівняння (12) та (14), можна встановити, що $\xi_i = 0$, якщо $\alpha_i < C$. Лінійне розділення точок за наявності похибки наведено на рис. 7.

Все, що розглядалося вище, стосується лінійно розділимих задач. На практиці такий вид категоризації зустрічається, проте він є лише винятком, здебільшого категоризаційні задачі не допускають лінійного розподілення об'єктів.

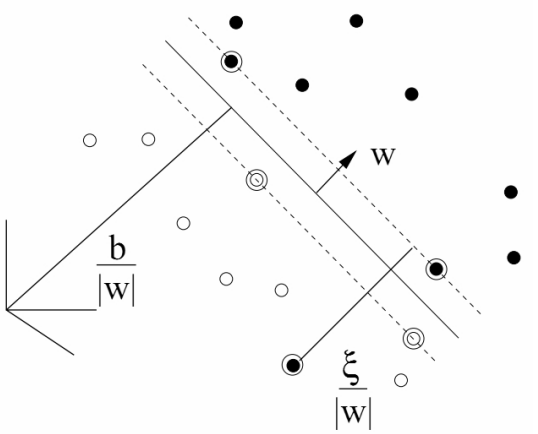

Рис. 7. Лінійне розділення точок за наявності похибки

Для того, щоб задача знову стала лінійно розділимою і можна було застосовувати розглянуті вище лагранжіани, треба розмістити дані у просторі більш високого порядку, де можливе лінійне розділення об'єктів гіперплощиною. Цю операцію називають спрямленням простору або мапінгом даних (рис. 8).

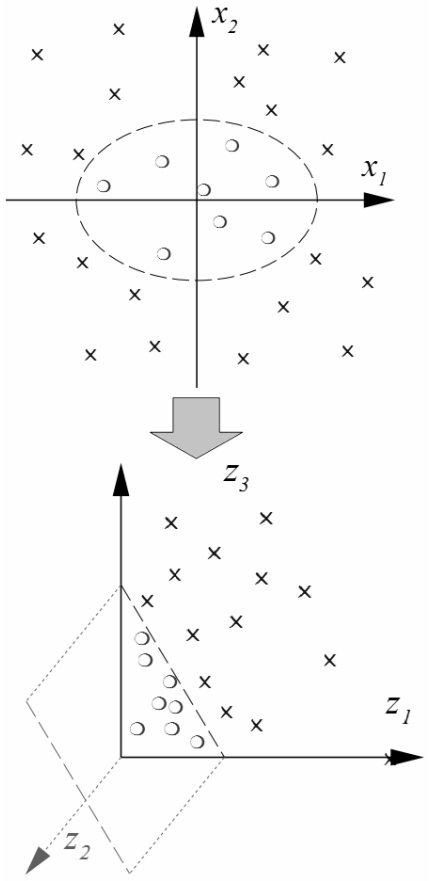

Рис. 8.

Приклад мапінгу даних Мапінг даних позначається Φ і виконує відображення даних із вихідного гіперпростору R^d до евклідового простору $\aleph$

$$\Phi = R^d \mapsto \aleph. \qquad (15)$$

Отже, для того, щоб задача стала лінійно розділимою, до кожної з точок треба застосувати перетворення (15) і надалі оперувати не з самими точками, а з відповідними їм $\Phi(x)$. Логіка такого розподілення добре ілюструється рис. 9.

У розглянутих лагранжіанах часто застосовуються скалярні добутки точок. Для спрощення оптимізаційної задачі було розроблено спосіб, що називається kernel trick, в основі якого лежить перехід від скалярних добутків до так званих функцій ядра. При цьому кожен скалярний добуток замінюється нелінійною функцією ядра (скалярним добутком у просторі з більшою розмірністю). Функція ядра може бути представлена наступним чином [7 – 11]:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

Використання функції ядра взагалі дає можливість відійти від необхідності користуватись $\Phi(x)$. Користувач методу може навіть не знати, яку $\Phi(x)$ використовує та чи інша функція ядра.

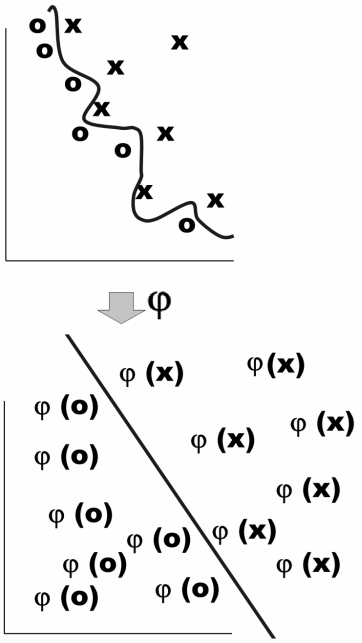

Рис. 9. Ілюстрація до застосування $\Phi(x)$ до вихідних даних

На етапі навчання SVM використовується для обчислення $f(x)$ за вихідними даними навчання з використанням формули (16)

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i) \cdot \Phi(x) + b = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b. \quad (16)$$

Найпростіше реалізувати нелінійний метод SVM на базі L_D, або так званого lagrangian trick із застосуванням kernel trick. При цьому максимізується рівняння (17) відносно рівностей (4) та (5). Функція ядра при цьому може бути представлена рівнянням (18)

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j. \quad (17)$$

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^p, \quad (18)$$

де p – деякий параметр, що підлягає налаштуванню користувачем.

Висновок

Метод опорних векторів зводить навчання класифікатора до оптимізаційної задачі, яка розв'язується евристичними алгоритмами. Для побудування нелінійних класифікаторів використовується розширення простору та функції ядер. Метод SVM у тестах перемагає інші методи за швидкістю та точністю категоризації. При різному підході до вибору ядер метод може емулювати роботу інших математичних методів. SVM може працювати як нейронна мережа, проте таке використання обмежує його призначення, оскільки метод значно перевершує їх за можливостями.

SVM може бути успішно застосований для керування складними електромеханічними системами, він може забезпечити адаптивність алгоритмів керування, виконувати функції спостерігача, ідентифікатора невідомих параметрів, деякої еталонної моделі, за його допомогою можна керувати складними нелінійними об'єктами, а також об'єктами зі стохастичними параметрами.

ЛІТЕРАТУРА

1. Burges C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2:121–167, 1998
2. Burges C.J.C., Knirsch P., Haratsch R. Support vector web page: http://svm.research.bell-labs.com. Technical report, Lucent Technologies, 1996.
3. Cortes C., Vapnik V. Support vector networks. Machine Learning, 20. — P. 273—297, 1995.
4. Bartlett P., Shawe-Taylor J. Generalization performance of support vector machines and ot he r pattern classifiers // Advances in Kernel Methods: MIT Press, Cambridge , USA , 1998.
5. Platt J. C. Fast training support vector machines using sequential minimal optimization // Advances in Kernel Methods / Ed. by B . Schölkopf, C. C. Burges, A. J. Smola. MIT Press, 1999. P. 185—208.
6. Vapnik V., Chapelle O. Bounds on error expectation for support vector m achine s // Neural Computation. — 2000. — Vol. 12, no. 9. — P. 2013—2036.
7. Smola A. J., Schölkopf B. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. Algorithmica, 22:211 — 231, 1998.
8. Nello Cristianini, John Shawe-Taylor. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000. — 204 p.
9. John Shawe-Taylor, Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004. — 462 p.
10. Ingo Steinwart, Andreas Christmann,; Support Vector Machines, Springer-Verlag, New York, 2008. — 602 p.
11. Muller K., Mika S.An Introduction to Kernel-Based Learning Algorithms, IEEE Neural Networks. — 2001. — №12(2), — P. 181—201.

пост.09.04.13