Доклады БГУИР
2008 *январь–март* *№ 1 (31)*

ИНФОРМАТИКА

УДК 681.3

ПОСТРОЕНИЕ КЛАССИФИКАТОРА НА ОСНОВЕ МАШИНЫ ОПОРНЫХ ВЕКТОРОВ ДЛЯ РАСПОЗНАВАНИЯ СИМВОЛОВ

И.И. ФРОЛОВ, Р.Х. САДЫХОВ

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь

Поступила в редакцию 15 января 2007

Машина опорных векторов представляет собой категорию универсальных нейронных сетей прямого распространения. Рассмотрена методика построения машины опорных векторов для создания классификатора, применяемого для распознавания и классификации символов.

Ключевые слова: машина опорных векторов, минимизация структурного риска, разделяющая гиперплоскость, спрямляющее пространство.

Введение

Машина опорных векторов — это алгоритм, обучающийся распознавать объекты двух классов. Для многоклассовой задачи принцип распознавания строится по принципу "один против всех". Сущность работы машин опорных векторов состоит в построении гиперплоскости, максимально разделяющей положительные и отрицательные точки. При этом среди всех таких гиперплоскостей находится та, для которой минимальное расстояние (зазор) до ближайших точек максимально. Это достигается благодаря принципиальному подходу, основанному на теории статического обучения, предложенной Владимиром Вапником [1, 8].

Машина опорных векторов может обеспечить хорошее качество распознавания в задаче классификации, не обладая априорными знаниями о предметной области конкретной задачи, т.к. работает с абстрактной векторной моделью представления данных. Именно это свойство является уникальным для машин опорных векторов.

Теоретическое обоснование метода

Рассмотрим простейший случай: линейные машины, обученные на разделенных данных (далее будет показано, что анализ для общего случая – нелинейные машины, обученные на линейно неразделяемых данных — задача, близкая к задаче квадратичного программирования). Отметим обучаемые данные, как $\{x_i, y_i\}, i = 1,...,l; y_i \in \{-1,+1\}, x_i \in R^n$. Предположим, имеется некоторая гиперплоскость, отделяющая положительные от отрицательных данных (разделяющая гиперплоскость) [4, 9]. Точки x, которые лежат на гиперплоскости, удовлетворяют условию

$$\langle w, x \rangle + b_0 = 0, \quad (1)$$

где w — нормаль к гиперплоскости, тогда $|b_0|/\|w\|$ — перпендикуляр к гиперплоскости, опущенный из начала координат, $\|w\|$ — евклидова норма вектора w. Пусть x_- и x_+ — две произвольные точки классов –1 и +1 соответственно, лежащие на границе разделяющей полосы (т.е. имеющие наименьшее расстояние от разделяющей гиперплоскости до ближайшей положительной (отрицательной) точки). Ширина разделяющей полосы определяется как $(x_- + x_+)$. Для случая линейной разделимости алгоритм ищет разделяющую гиперплоскость с наибольшей полосой разделения (рис. 1), что также можно представить в виде следующего набора ограничений:

$$\langle w, x_i \rangle + b_0 \geq +1 \quad \text{для } y_i = +1, \tag{2}$$

$$\langle w, x_i \rangle + b_0 \leq -1 \quad \text{для } y_i = -1 \tag{3}$$

либо можно объединить их одной системой неравенств:

$$y_i(\langle w, x_i \rangle + b_0) - 1 \geq 0 \,\forall i\,. \tag{4}$$

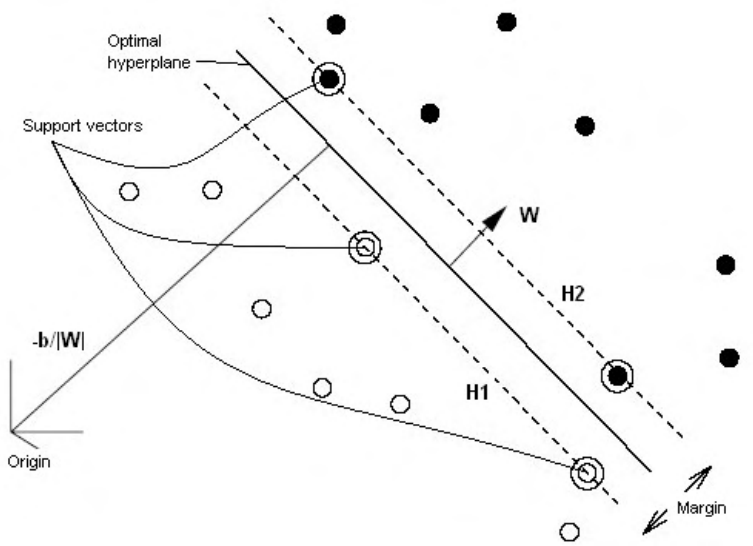

Рис. 1. Оптимальная гиперплоскость для линейно-разделимых образов: Optimal hyperplane — оптимальная разделяющая гиперплоскость; Support vectors — опорные векторы; Margin — разделяющая полоса; Origin — начало координат [4]

Точки, удовлетворяющие неравенству (2), лежат на границе полосы $H1$ $\langle w, x \rangle + b_0 = 1$ с нормалью w и перпендикуляром из начала координат $|1 - b_0|/\|w\|$. Аналогично точки, удовлетворяющие неравенству (3), лежат на границе полосы $H2$ $\langle w, x \rangle + b_0 = -1$ с противоположной нормалью w и перпендикуляром из начала координат $|-1 - b_0|/\|w\|$. Следовательно, $x_-=x_+=1/\|w\|$ и ширина полосы разделения равна $2/\|w\|$ [4, 5]. Необходимо отметить, что поверхности $H1$ и $H2$ параллельны (имеют общую нормаль) и ни одна из точек обучающей выборки не может лежать внутри этой полосы. При этом разделяющая гиперплоскость проходит точно посередине разделяющей полосы. Чтобы разделяющая гиперплоскость как можно дальше отстояла от точек выборки, ширина разделяющей полосы должна быть максимальной. Таким образом, необходимо найти такие значения параметров w и b_0, при которых норма вектора w минимальна при условиях (2), (3). Это задача квадратичного программирования.

Построение оптимальной разделяющей гиперплоскости сводится к минимизации квадратичной формы при l ограничениях-неравенствах относительно $n+1$ (исходя из определения размерности Вапника-Червоненкиса) переменных w, b_0:

$$\begin{cases} \langle w, w \rangle \to \min, \\ y_i(\langle w, x_i \rangle - b_0) \geq 1, i = 1, ..., l. \end{cases} \tag{5}$$

По теореме Куна–Таккера [2] эта задача эквивалентна двойственной задаче поиска седловой точки функции Лагранжа:

$$\begin{cases} L\langle w, b_0, \lambda \rangle = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{l} \lambda_i \left(y_i \left(\langle w, x_i \rangle - b_0 \right) - 1 \right) \to \min\limits_{w, b_0} \max\limits_{\lambda}, \\ \lambda_i \geq 0,\ i = 1, ..., l; \\ \lambda_i = 0 \ \textit{или} \ y_i(\langle w, x_i \rangle - b_0) = 1,\ i = 1, ..., l. \end{cases} \tag{6}$$

где $\lambda = (\lambda_1, ..., \lambda_l)$ — вектор двойственных переменных [2]. Последнее из трех условий (6) называется условием дополняющей нежесткости.

Чтобы обобщить машину опорных векторов на случай линейной неразделимости, позволим алгоритму допускать ошибки на обучающих объектах, но при этом постараемся, чтобы их было поменьше. Введем набор дополнительных переменных $\xi_i \geq 0$, характеризующих величину ошибки на объектах $x_i, i = 1, ..., l$ соответственно. Это позволяет смягчить ограничения-неравенства и одновременно ввести в функционал штрафное слагаемое за суммарные ошибки:

$$\begin{cases} \frac{1}{2}\langle w, w \rangle + C\sum_{i=1}^{l} \xi_i \to \min\limits_{w, b_0, \xi}, \\ y_i(\langle w, x_i \rangle - b_0) \geq 1 - \xi, i = 1, ..., l, \\ \xi \geq 0, i = 1, ..., l. \end{cases} \tag{7}$$

Предполагается, что если $\xi_i = 0$, то на объекте x_i ошибки нет. Если $\xi_i > 1$, то на объекте x_i допускается ошибка. Если $0 < \xi_i < 1$, то объект попадает внутрь разделяющей полосы [7], но корректно классифицируется машиной опорных векторов (распознаваемый объект относится к своему соответствующему классу).

Положительная константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки.

Функция Лагранжа для данной задачи имеет вид [2]

$$L(w, b_0, \xi, \lambda, \eta) = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{l} \lambda_i (y_i(\langle w, x_i \rangle - b_0) - 1) - \sum_{i=1}^{l} \xi_i(\lambda_i + \eta_i - C), \tag{8}$$

где $\eta = (\eta_1, ..., \eta_l)$ — вектор переменных, двойственных к переменным $\xi = (\xi_1, ..., \xi_l)$.

Существует еще один путь к решению проблемы линейной неразделимости. Это переход от исходного пространства признаковых описаний объектов X к новому пространству H с помощью некоторого преобразования $\psi : X \to H$ [7]. Если пространство H имеет достаточно высокую размерность, то можно надеяться, что в нем выборка окажется линейно разделимой (легко показать, что если выборка X^l не противоречива, то всегда найдется пространство раз-

мерности не более *l*, в котором она будет линейно разделима). Пространство *H* называется спрямляющим [7].

Если предположить, что признаковыми описаниями объектов являются векторы $\psi(x_i)$, а не векторы x_i [7,9], то построение машины опорных векторов проводится практически так же, как и ранее. Единственное отличие состоит в том, что скалярное произведение $\langle x, x' \rangle$ в пространстве *X* всюду заменяется на скалярное произведение $\langle \psi(x), \psi(x') \rangle$ в пространстве *H*. Отсюда вытекает естественное требование: пространство *H* должно быть наделено скалярным произведением, в частности, подойдет любое евклидово или гильбертово пространство.

Ядром называется функция $K : X \times X \to R$ [4, 6], представимая в виде $K(x, x') = \langle \psi(x), \psi(x') \rangle$ при некотором $\psi : X \to H$, где *H* — пространство со скалярным произведением.

Подведем некоторые итоги. Как следует из названия, основная идея создания этой машины опорных векторов состоит в выборе подмножества обучающих данных в качестве опорных векторов, которое представляет устойчивые свойства всей обучающей выборки.

Обучение машины опорных векторов сводится к задаче квадратичного программирования [2], что привлекательно по двум причинам:

– процесс обучения гарантированно сходится к глобальному минимуму на поверхности ошибки;

– вычисления могут быть реализованы достаточно эффективно.

Программная реализация классификатора

Для реализации классификатора на базе машины опорных векторов был использован язык программирования C++. Среди многих современных языков программирования язык C++ выделяется ориентацией на объектно-ориентированное программирование.

Ниже рассматривается методика построения классификатора, используемого для распознавания печатных символов. Однако следует отметить, что данный классификатор может быть применен и для других областей при условии представления данных для классификации в предложенном определенном формате.

Работу классификатора можно разбить на две составляющие части: обучение классификатора, т.е. построение модели, и непосредственно распознавание подаваемых на вход символов.

В соответствии с данной последовательностью на первом этапе разрабатывается программная реализация используемого математического аппарата машины опорных векторов для создания модели классификатора [5, 6]. Все выполняемые математические операции формализованы на языке программирования C++ непосредственно в программном коде. Достоинством данной реализации является переносимость полученного программного продукта. С другой стороны, использование пакетов автоматизированной обработки математической информации позволяет упростить программный код и увеличивает скорость вычислений. На выходе данной разработки получаем программный файл, предназначенный для построения модели обучения классификатора. При запуске данной программы в командной строке задаются параметры обучения и прописывается путь к файлу с образцами для обучения.

Вторым этапом работы классификатора является процесс распознавания и классификации. Для распознавания экспериментальных образцов запускается специально созданный программный файл, в основе реализации которого также лежит математический аппарат машины опорных векторов [5, 6, 10] для классификации данных с использованием конкретной, ранее созданной модели обучения. При запуске данной программы в командной строке задаются параметры распознавания, прописывается путь к модели обучения и к файлу с образцами данных, предназначенных для распознавания и классификации.

Тестирование классификатора. Проведение эксперимента

Для проверки корректной работы построенного классификатора был выбран буквенный ряд латинского алфавита, состоящий из 26 символов, от A до Z, и числовой ряд, состоящий из 10 символов, арабских цифр 0–9. Таким образом, полный набор данных содержит 36 классов. Для отображения символов использована матрица размером 5×8:

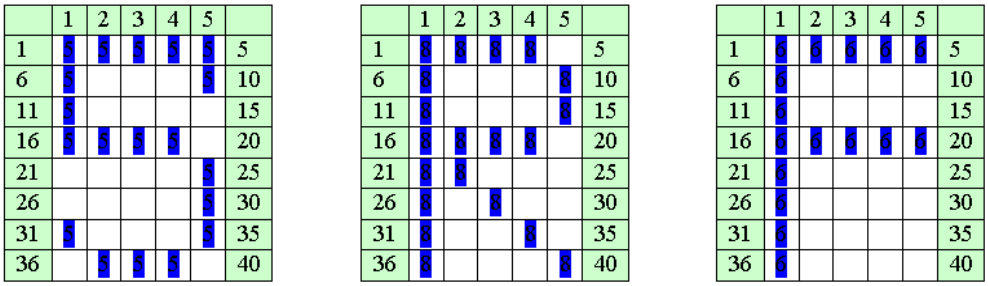

Рис. 2. Примеры представления символов (обучающих данных)

Каждый символ представляется в виде массива чисел с обозначением, т.е. меткой, класса, индекса признака образца и численным значением данного признака. Например, таким образом представлен символ цифры "5":

35 1:1 2:1 3:1 4:1 5:1 6:1 7:0 8:0 9:0 10:1 11:1 12:0 13:0 14:0 15:0 16:1 17:1 18:1 19:1 20:0 21:0 22:0 23:0 24:0 25:1 26:0 27:0 28:0 29:0 30:1 31:1 32:0 33:0 34:0 35:1 36:0 37:1 38:1 39:1 40:0,

где "35" в начале строки — метка класса, пары значений <1:1>, <2:1>, <3:1>,…, <40:0> — вектор признаков класса и их соответствующих значений. Если элемент матрицы образца (признак) заполнен (закрашен), то его численное значение равно "1", иначе "0", например для представленной выше цифры "5" 1:1 2:1, …, 39:1 40:0. Нумерация признаков осуществляется слева направо сверху вниз, таким образом, левый верхний угол матрицы соответствует 1-му признаку образца, а правый нижний — 40-му.

Для обучения классификатора (построения модели) было создано по 9 обучающих образцов для каждого класса, всего 324 образца-вектора. Для тестирования классификатора (классификации) было создано по 12 образцов для каждого класса. Порядок расстановки образцов для классификации (вектор тестируемых данных) выбирался случайным образом. Всего для тестирования было создано 432 образца. Правильно классифицированы (соотнесены к своим классам) были 409 из 432 образцов, что составляет 94,67% достоверной классификации.

Следует отметить, что время обучения классификатора составило 8,3 с, а тестирование классификатора (классификация) было проведено за 0,7 с. Такое распределение времени обучения связано в первую очередь с реализацией математических вычислений в программном коде, а также и с объемом вычислений, который прямо пропорционален объему данных для обучения классификатора, длине единичного вектора данных образца.

Выводы

Описанный метод распознавания и классификации данных на данном этапе экспериментальной разработки позволяет судить о достаточно высокой вероятности достоверной классификации данных. Главным достоинством данной программной реализации машины опорных векторов является возможность использования программы для классификации данных из различных областей науки, единственным требованием в этом случае является представление образцов данных в формате вектора признаков и соответствующих им значений. Следует отметить высокую скорость работы классификатора с использованием уже готовой модели.

Однако требует доработки вопрос об уменьшении времени обучения классификатора, что в первую очередь можно осуществить при перенесении нагрузки выполнения математических расчетов при использовании пакетов автоматизированной обработки математической информации (например, Matlab).

Таким образом, приведенный вид классификации данных может получить дальнейшее перспективное развитие при должной доработке и использоваться в работе систем распознавания образов, графической информации, портретов лиц.

THE QUALIFIER CONSTRUCTION ON THE BASIS OF THE SUPPORT VECTOR MACHINES FOR IDENTIFICATION OF SYMBOLS

I.I. FROLOV, R.Kh. SADYKHOV

Abstract

The support vector machine represents a category of universal neural networks of direct distribution. The technique of construction of the support vector machines for creation of the qualifier applied to identification and classification of symbols is considered.

Литература

1. *Вапник В.Н., Червоненкис А.Я.* Теория распознавания образов. Статистические проблемы обучения, М., 1974.
2. *Ларин Р.М., Плясунов А.В., Пяткин А.В.* Методы оптимизации. Примеры и задачи: Учеб. пособие. Новосибирск, 2003.
3. *Хайкин С.* Нейронные сети: полный курс. М., 2005.
4. *Burges C.J.C.* A Tutorial on Support Vector Machines for Pattern Recognition., Boston, 1998.
5. *Noble W.S., Pavlidis P.* Gist: Support vector machine and kernel principal components analysis software toolkit.
6. *Platt J., Cristianini N., Shawe-Taylor J.* // Large margin DAGs for multiclass classification. In Solla S.A., Leen T.K., and Muller K.-R., editors. Advances in Neural Information Processing Systems. MIT Press, 2000. Vol. 12, P. 547–553.
7. *Schölkopf B., Burges C.J.C., Smola A.J.* Advances in Kernel Methods. Support Vector Learning, MIT Press, Cambridge, USA, 1998.
8. *Vapnik V.* // NEC Journal of Advanced Technology. 2005. № 2.
9. *Vapnik V.* Statistical Learning Theory. Wiley, 1998.
10. *Vapnik V.N.* Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing. Communications and Control. Wiley, New York, 1998.