

**В. В. Шебанов** – студент кафедры компьютерной математики и программирования

**М. Д. Поляк** – научный руководитель

## МЕТОД ОПОРНЫХ ВЕКТОРОВ В РАСПОЗНАВАНИИ ИЗОБРАЖЕНИЙ

Распознавание образов невероятно обширная тема, которая вызывает неподдельный интерес в наше время. Причиной этого является, в основном то, что результаты теоретических исследований в этой области с лёгкостью находят применение в коммерческих продуктах, военной промышленности и многих других сферах.

Благодаря большому количеству работ в этой сфере, было предложено много методов решения проблемы распознавания образов. Наиболее популярны из них различные виды нейронных сетей, а также механизм опорных векторов. Метод нейронных сетей уже показал множество положительных результатов и лёг в основу различных интеллектуальных систем, однако, он с трудом поддаётся четкому математическому описанию, что затрудняет его изучение и использование. В свою очередь метод опорных векторов имеет более или менее чёткое описание, что позволяет делать выводы относительно его эффективности относительно других методов распознавания.

Для экспериментов были выбраны фотографии человеческих лиц. Целью данного исследования является изучение метода опорных векторов на основе реализации алгоритма распознавания лиц.

Задачи, которые необходимо решить:

- разработать алгоритм распознавания лиц с помощью SVM с использованием метода один-против-всех;

- проанализировать точность данного метода

Так как в этой работе будет исследован метод опорных векторов, то сначала нужно описать основы самого метода.

Метод опорных векторов – это набор схожих алгоритмов вида «обучение с учителем», использующихся для задач классификации и регрессионного анализа. Этот метод принадлежит к семейству линейных классификаторов. Он может также рассматриваться как специальный случай регуляризации по А. Н. Тихонову.

Рассмотрим процесс бинарной классификации. Пусть существует множество векторов,  $x_i \in R^n$ ,  $i = 1, 2, \dots, N$ , где каждый вектор принадлежит к одному из двух классов, обозначаемых  $y_i \in \{-1, 1\}$ . В предположении, что входные данные линейно разделимы, задача классификации с максимальным зазором — разделить два класса такой гиперплоскостью, чтобы максимизировать расстояние между ней самой и разделяемыми классами.

Оптимальная разделяющая гиперплоскость (ОРГ) имеет следующую форму:

$$f(x) = \sum_{i=1}^l a_i y_i x_i \cdot x + b, \quad (1)$$

Коэффициенты  $a_i$  и  $b$  в уравнении (1) являются решением проблемы квадратичного программирования [1]. Обратим внимание, что реально суммирование идёт не по всей выборке, а только по опорным векторам, для которых  $a \neq 0$ . Именно это свойство разреженности (sparsity) отличает SVM от других линейных разделителей: дискриминанта Фишера, логистической регрессии и однослойного персептрона. Классификация новой точки  $x$  выполняется путём вычисления знака правой стороны уравнения (1). В дальнейшем будет использоваться

$$d(x) = \frac{\sum_{i=1}^l a_i x_i y_i \cdot x + b}{\left\| \sum_{i=1}^l a_i y_i x_i \right\|} \quad (2)$$

для выполнения классификации в случае наличия более чем двух классов. Знак  $d$  это результат классификации для  $x$ , а  $|d|$  это расстояние от  $x$  до гиперплоскости. Очевидно, что чем больше  $|d|$ , тем надежнее результат классификации. Результат может быть расширен на случай нелинейных разделя-

ющих поверхностей. Каждой точки  $x$  сопоставляется  $z = \Phi(x)$ , принадлежащая к пространству большей размерности. Далее перепишем произведение двух точек  $z$ , как функцию  $K(x, y) = \Phi(x) \cdot \Phi(y)$ . Тогда уравнение (1) примет вид:

$$f(x) = \sum_{i=1}^l y_i a_i K(x, x_i) + b. \quad (3)$$

Заметим, что  $f(x)$  не зависит от размерности  $K$ . Одним из важнейших семейств  $K$ -функций является полином вида:

$$K(x, y) = (1 + x \cdot y)^d, \quad (4)$$

где  $d$  – это степень полинома.

Существуют две основные стратегия решения проблемы классификации для случая с более чем двумя классами:

- метод один-против-всех. Обучаются  $q$  классификаторов. Каждый отделяет один класс от всех остальных;
- метод попарной классификации. Обучаются  $q(q-1)/2$  классификаторов. Каждый отделяет пару классов от всех остальных. Попарный классификатор представляется деревом, где каждый узел это отдельный классификатор.

На данный момент не существует аналитической оценки и сравнения этих двух алгоритмов, однако метод один-против-всех может считаться предпочтительным, так как для его реализации необходимо количество классификаторов, которое линейным образом зависит от количества классов. Вычислительная сложность этих алгоритмов похожа. Сложность первого алгоритма –  $O(q)$ , сложность второго –  $O(q-1)$ . Однако, недавние экспериментальные исследования [2] в этой области показали схожую скорость исполнения для двух этих методов.

Напомним, что функцией ядра SVM называется некоторая нелинейная функция, которая заменят произведение в формуле (1). Такой переход иногда называют «kernel-trick», потому что он позволяет строить нелинейные решающие поверхности путём замены всего одной операции.

Формальное определение: функция  $K: X \times X \rightarrow R$  называется ядром (kernel function), если она представима в виде  $K(x, x') = \langle \psi(x), \psi(x') \rangle$ , при некотором отображении  $\psi: X \rightarrow H$ , где  $H$  – пространство со скалярным произведением.

Ядром SVM может являться не любая функция. Ограничения на функцию описывает теорем Мерсера.

Теорема 1 (Мерсер, 1909). Функция  $K(x, x')$  является ядром тогда и только тогда, когда она симметрична,  $K(x, x') = K(x', x)$ , и неотрицательно определена:

$$\iint_{x \times x} K(x, x') g(x) g(x') dx dx' \geq 0 \text{ для любой функции } g: X \rightarrow R.$$

Полиномиальное ядро. Полиномиальное ядро не стационарно и чаще всего используется для работы с нормализованными данными.

$$k(x, y) = (ax_i \cdot x_j + c)^d. \quad (5)$$

Параметрами ядра являются: множитель  $a$ , константа  $c$  и степень полинома  $d$ .

Ядро Гаусса. Это ядро является примером радиальной базисной функции.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \quad (6)$$

Так же, иногда, оно записывается следующим образом:

$$k(x, y) = \exp(-\gamma \|x - y\|^2), \quad (7)$$

Для данной функции очень важен параметр  $\sigma$ , от которого очень сильно зависит способность обучающейся машины к обобщению. Если сделать его слишком большим, то экспонента будет вести себя как линейная функция, что затрудняет распознавание классов, которые линейно неразделимы. Если же сделать его слишком маленьким, то итоговая решающая поверхность будет плохо работать даже на слабо зашумлённых данных.

Данное ядро очень мощный инструмент и подходит для работы с практически любыми данными, однако, требуется дополнительное время для выбора оптимального параметра  $\sigma$ .

Преимущества SVM:

- решается задача квадратичного программирования, имеющая единственное решение. Методы оптимизации в этом случае существенно более эффективны;
- принцип оптимальной разделяющей гиперплоскости приводит к максимизации ширины разделяющей полосы между классами, следовательно, к более уверенной классификации. Например, градиентные нейросетевые методы выбирают положение разделяющей гиперплоскости произвольным образом, как придётся.

Недостатки SVM:

- метод опорных векторов неустойчив по отношению к шуму в исходных данных. Если обучающая выборка содержит шумовые выбросы, они будут существенным образом учтены при построении разделяющей гиперплоскости;
- до сих пор не разработаны общие методы построения ядер, наиболее подходящих для конкретной задачи. Построение адекватного ядра является искусством и, как правило, опирается на априорные знания о предметной области. На практике, вполне разумные функции  $K(x, x')$ , выведенные из содержательных соображений, далеко не всегда оказываются положительно определёнными.

Как и в случае с нейронными сетями, выбор ядра является сложной задачей и решается исследователем в каждом отдельном случае. Хотя в работе Tom Howley [3] сделаны попытки автоматизировать процесс выбора функции ядра. Автор предлагает использовать подход, так называемого, генетического программирования для поиска оптимальной функции, подходящей под условия данной задачи и удовлетворяющей теореме Мерсера.

Таким образом, можно сделать вывод, что алгоритм довольно перспективный и при должном подборе параметров сможет решать поставленную перед ним задачу – распознавание лиц.

Была использована база изображений University of Essex. В данной базе содержатся изображения 150 людей, по 20 фотографий на человека.



Рис. 1 Пример изображения из тестовой выборки

Для обучения SVM были использованы первые 15 фотографий, оставшиеся же 5 были использованы для тестов, чтобы проверить способность машины к обобщению.

В качестве функции ядра было выбрано ядро Гаусса. Алгоритм обучения SVM состоит из 3 частей:

- 1) получение и предобработка изображений;
- 2) обучение SVM;
- 3) запись в БД.

Процесс предобработки состоит из следующих этапов:

- получение всех картинок для обучения;
- изменение размера всех картинок до размера 50x50 пикселей;
- преобразование цвета пикселей картинок в серые тона;

— преобразование картинки в вектор целых чисел, где каждому тону серого сопоставляется число от 0 до 255.

Далее идет процесс обучения:

- выбор параметров для ядра;
- обучение SVM.

Полученные результаты сохраняются в базу данных.

Алгоритм распознавания состоит из следующих этапов.

1. Получение и предобработка изображения.
2. Распознавание изображения каждым SVM из базы данных.
3. Выбор поверхности, расстояние от которой до входного изображения максимально.
4. Выдать в качестве результата имя, ассоциированное с поверхностью, выбранной на предыдущем шаге.

Первый этап идентичен для обучения и распознавания, таким образом производятся все те же действия, что при обучении.

Далее, на втором этапе нам нужно классифицировать входную картинку.

- применяем SVM;
- получаем расстояние до поверхности;
- выбираем максимальное расстояние;
- считаем ответом SVM получивший этот результат;
- возвращаем имя человека.

В таблицах 1 и 2 приведены результаты исследования скорости работы алгоритмов обучения и распознавания. Как видно из результатов, время записи в базу данных уменьшилось. При измерении точности распознавания, алгоритм выдавал верный ответ с вероятностью 93%.

Таблица 1.  
Оценка времени работы алгоритма обучения с использованием Redis

Операция	Время, мс	Время, с
Обучение	6715.9	6.72
Выбор параметров ядра	6401.7	6.4
Процесс обучения	15.4	0.02
Запись в БД	74.9	0.07

Таблица 2.  
Оценка времени работы алгоритма распознавания с использованием Redis

Операция	Время, мс	Время, с
Распознавание	50429.3	50.43
Предобработка	36	0.04
Выбор значения из БД	334.5	0.33
Применение поверхности	1.2	0.001

Результатом данного исследования стал алгоритм, который с довольно хорошей точностью может распознавать лица.

Из плюсов работы SVM и метода один-против-всех:

- высокая точность;
- данная схема легко масштабируется. Например, данное решение можно легко комбинировать с технологией MapReduce [4].

Направления по улучшению работы алгоритма:

- применение техники Virtual SVM [5];
- оптимизация скорости распознавания.

Очевидно, что метод опорных векторов – это простой и эффективный способ классификации изображений и их распознавания.

#### **Библиографический список**

1. *V. Vapnik* Statistical learning theory. John Wiley and Sons, New York, 1998.
2. *C. Nakajima, M. Pontil, B. Heisele, and T. Poggio*. Person recognition in image sequences: The mit espresso machine system. submitted to IEEE Transactions On Neural Networks, 2000.
3. *Tom Howley, Michael G. Maden*. The Genetic Kernel Support Vector Machine: Description and Evaluation. Department of Information Technology, National University of Ireland, Galway, Ireland, 2005.
4. *Jeffrey Dean and Sanjay Ghem*a. MapReduce: Simplified Data Processing on Large Clusters. Google, Inc.
5. *Bernhard Scholkopf, Chris Burges and Vladimir Vapnik*. Incorporating Invariances in Support Vector Machine Learning. International Conference on Artificial Neural Networks, Springer Verlag, 1996.