Работа № 3.5 Метод опорных векторов

1. Теоретические сведения

Метод опорных векторов (МОВ, англ. Support Vector Machine, SVM) - это техника машинного обучения с учителем. Она используется в классификации, может быть применена к регрессионным задачам.

Метод определяет границу принятия решения (далее ГПР) вместе с максимальным зазором, который разделяет почти все точки на два класса, оставляя место для неправильной классификации.

Цель МОВ — определить гиперплоскость (также называется «разделяющей» или ГПР), которая разделяет точки на два класса.

Для ее визуализации представим двумерный набор данных:

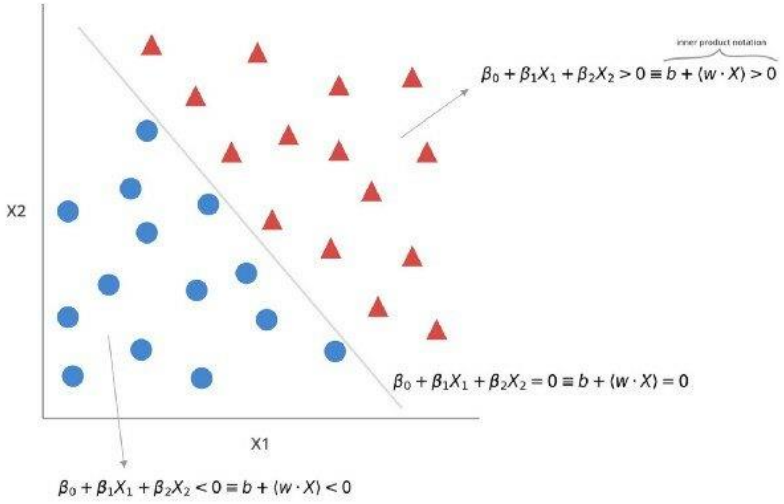


Рисунок. Гиперплоскость, которая полностью разделяет точки на двакласса.

Математическая постановка

Пусть даны два линейно разделимых класса объектов

Мы можем описать все точки разделяющей гиперплоскости используя вектор-нормаль к этой гиперплоскости:

$$\langle w, x \rangle - b = 0$$

Данная разделяющая гиперплоскость находится в «разделяющей полосе», которую мы также можем задать уравнениями:

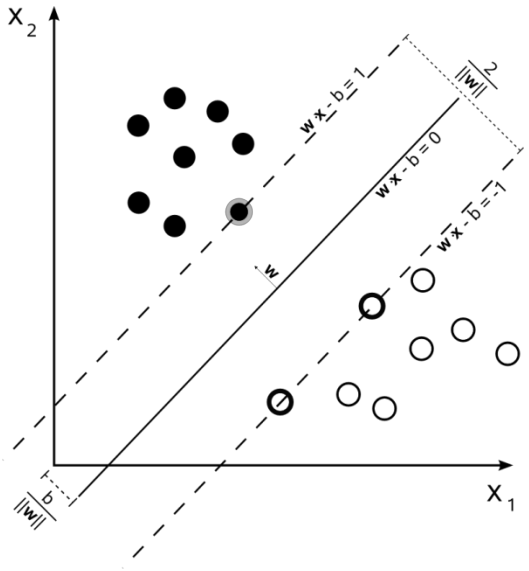
$$\langle w, x \rangle - b = -1 \qquad \langle w, x \rangle - b = 1$$

Можно найти ширину данной полосы как

Для машины опорных векторов необходимо найти разделяющую гиперплоскость, которая задает полосу максимальной ширины. Задача оптимизации:

$$\frac{1}{2}\|w\|^2 \to min$$

$$y_i(\langle w, x_i \rangle - b) \geq 1, \forall i = 1, \dots, n$$



Решение оптимизационной задачи

Для решения оптимизационной задачи с прошлого слайда можно воспользоваться известным методом множителей Лагранжа (Lagrangianrelaxation). Коэффициенты лямбда неотрицательны, поэтому если некоторые точки нарушают ограничения, заданные задачей, значение функции увеличивается. Устремив эту функцию к минимуму по w, b мы будем стремиться к тому, чтобы как можно больше точек не нарушало ограничения. Таким образом, 2-ая часть уравнения будет вносить неположительную часть в функцию и максимальное значение функции будет $\frac{\|w\|^2}{2}$. Максимизируя после этого по неотрицательным лямбда мы получим некоторую нижнюю границу оптимального значения.

$$L_P = \frac{\|w\|^2}{2} - \sum_{i=1}^{n} \lambda_i (y_i(< w, x_i > + b) - 1) \rightarrow \max_{\lambda_i \geq 0} \min_{w,b}$$

Взяв частные производные по w, b и приравняв к нулю, мы избавляемся от минимизации по этим переменным и можем заменить w = sum(y_i*lambda_i*x_i), 0 = sum(y_i*lambda_i). Можно перейти к двойственной проблеме, которая максимизируется по неотрицательным лямбда. Для этого нужно взять производные по переменным минимизации (w, b) и полученные значения подставить в прямую задачу. Получается двойственная задача, которую можно максимизировать только по лямбдам.

$$\frac{\partial L_P}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^{n} \lambda_i y_i x_i \, \frac{\partial L_P}{\partial b} = 0 \quad \rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0$$

Для вычисления двойственной функции достаточно знать метки классов, а также скалярное произведение двух точек. Значения w и b можно затем найти после решения задачи для lambda.

Двойственная задача максимизации может быть решена с помощью градиентного спуска, где вектор лямбд итерационно обновляется с помощью вектора частных производных функции L_D.

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j < x_i, x_j > \rightarrow \max_{\lambda_i \geq 0}$$

Случай линейно неразделимой выборки

Формулировка епсилон как функции потерь и ограничений снизу эквивалентна. Ошибка епсилон равна 0 если объект расположен за пределами “разделяющей” полосы.

Ошибка от 0 до 1 говорит о том, что объект расположен внутри полосы, но с правильной стороны от разделяющей гиперплоскости. Ошибка больше 1 говорит о том, что объект классифицирован неверно. Данная формулировка позволяет классифицировать объекты с ошибкой, что обязательно происходит для линейно неразделимой выборки.

Для неразделимых классов вводится функция потерь

$$\varepsilon_i = max\big(0, 1 - y_i(\langle w, x_i\rangle - b)\big)$$

Задача оптимизации:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\varepsilon_i \to min$$

$$y_i(\langle w, x_i\rangle - b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0, \qquad \forall i = 1, \dots, n$$

Данная форма SVM называется C-classification.

Метод множителей Лагранжа также решает оптимизационную задачу для случая C-classification. Добавляются новые слагаемые с переменной эпсилон и минимизация L_P происходит по трём переменным $-w, b, \varepsilon$. Решение оптимизационной задачи для постановки soft-margin мало отличается от исходного решения оптимизационной задачи, поскольку частные производные по w, b остаются те же самые. Добавляется лишь производная по эпсилон, которая даёт дополнительное ограничение на лямбда – лямбда от 0 до C. Поскольку бета также является множителем Лагранжа и больше или равно нуля, а лямбда также больше нуля, то лямбда не может быть больше C.

Подставив найденные производные в исходную задачу, мы получим ту же самую двойственную проблему, потому что все слагаемые с эпсилон сократятся. Максимизируя двойственную проблему по всем лямбда от 0 до C можно найти решение задачи.

- Метод множителей Лагранжа:

$$L_P = \frac{\|w\|^2}{2} + C\sum_{i=1}^{n}\varepsilon_i - \sum_{i=1}^{n}\lambda_i(y_i(< w, x_i > + b) - 1 + \varepsilon_i) - \sum_{i=1}^{n}\beta_i\varepsilon_i \to \max_{\lambda_i \geq 0}\min_{w,b,\varepsilon_i}$$

- Производные:

$$\frac{\partial L_P}{\partial w} = 0 \;\to\; w = \sum_{i=1}^{n}\lambda_i y_i x_i \qquad \frac{\partial L_P}{\partial b} = 0 \;\to \sum_{i=1}^{n}\lambda_i y_i = 0$$

$$\frac{\partial L_P}{\partial \varepsilon_i} = 0 \;\to\; C - \lambda_i = \beta_i \geq 0, \forall i$$

- Двойственная проблема:

$$L_D = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j < x_i, x_j > \to \max_{0 \leq \lambda_i \leq C}$$

Предсказание

Функция предсказания представляет из себя просто определение расположения точки относительно разделяющей гиперплоскости. W заменена на значение своей производной. В функции предсказания также используется функция ядра (то есть простое скалярное произведение, либо одна из функций, его заменяющих) для внутреннего произведения двух точек. Если какая-то точка не лежит на границе «разделяющей» полосы, то максимальное значение функции будет достигаться если соответствующая лямбда = 0 (см. постановку прямой задачи с множителями лагранжа). Те точки, для которых лямбда не равно нулю и есть опорные, поскольку только от них зависит результат функции предсказания.

По сути, процесс обучения SVM на тренировочной выборке просто представляет из себя процесс решения двойственной задачи оптимизации.

Для предсказания результата алгоритма, используется функция sign:

$$F(z) = sign(\sum_{i=1}^{n} \lambda_i y_i \langle x_i, z \rangle + b)$$

Для $\lambda_i=0$ точка x_i не является «опорной», таким образом, в сумму, которая определяет класс нового объекта, влияние вносят только «опорные» точки.

Использование метода SVM

- Использован набор данных BanknoteAuthentification
- В качестве тестовой выборки взяты 107 последних объектов класса «0» (настоящие банкноты) и 119 объектов класса «1» (фальшивки). Остальные объекты используются в качестве обучающей выборки.
- Исходные параметры алгоритмов SVM взяты одинаковыми для разных библиотек.

Ядра (KernelTrick)

Вместо скалярного произведения точек в двойственной проблеме можно использовать специальные функции, называемые **ядрами**.

Линейное ядро—это аналог применения линейных преобразований к пространству объектов. Предположим, вы увеличиваете исходное пространство объектов возведением во вторую степень. Вы применили квадратичную функцию к исходному набору объектов. Теперь в этом расширенном пространстве есть оригинальная функция и ее квадратичная версия. Здесь неявно существует функция, которая сопоставляет эти два пространственных объекта.

$$x_1, x_2, x_3 \rightarrow x_1, x_1^2, x_2, x_2^2, x_3, x_3^2$$

Расширение пространства объектов с помощью квадратичной версии исходных.

Данные функции переводят пространство, в котором находятся наши точки в пространство большей размерности, что может привести к улучшенной разделимости классов. С полиномиальными ядрами вы проецируете исходное пространство объектов в полиномиальное. Граница, разделяющая классы, определяется полиномом более высокого порядка.

Использование ядер отличает классификаторы от метода опорных векторов, что открывает путь к решению более сложных задач. Но увеличение пространства признаков означает рост вычислительных требований. При большом пространстве функций подгонка модели станет дорогостоящей с точки зрения времени и ресурсов.

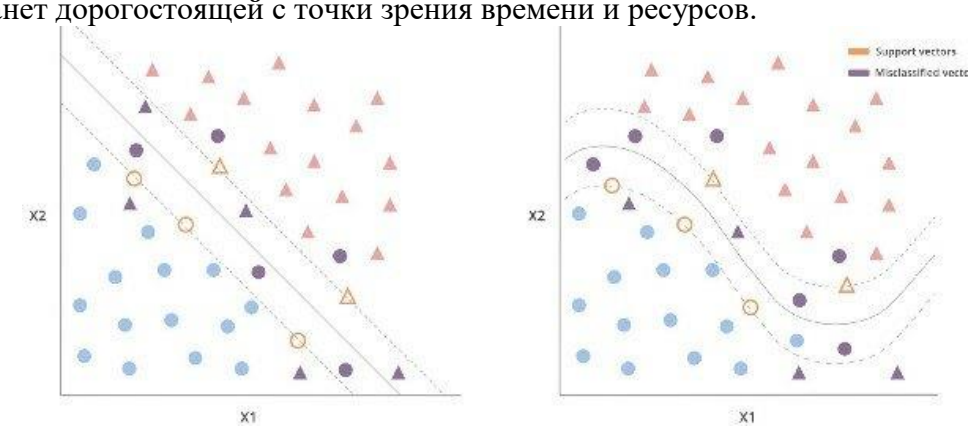


Рисунок. Граница решения и запасдля МОВ, наряду с соответствующими опорными векторами, используют линейное (справа) и полиномиальное ядро(слева).

Плюсы и минусы SVM

- Плюсы:
  - это наиболее быстрый метод нахождения решающих функций;

  - метод сводится к решению задачи квадратичного программирования в выпуклой области, которая всегда имеет единственное решение;
  - метод находит разделяющую полосу максимальной ширины (для заданных параметров), что позволяет в дальнейшем осуществлять более уверенную классификацию (и интерпретацию);
- Минусы
  - метод чувствителен к шумам и стандартизации данных;
  - не существует общего подхода к автоматическому выбору ядра, его параметров и построению спрямляющего подпространства в целом в случае линейной неразделимости классов.

2. Задача – выявление фальшивых банкнот

БазаBanknoteauthentification:
https://archive.ics.uci.edu/ml/datasets/banknote+authentication
Объекты представляют из себя характеристики изображений банкнот
- 1372 объекта
- 4 признака:
  - энтропия изображения
  - коэффициенты дисперсии, ассиметрии и эксцесса вейвлет-преобразования изображения
- Класс (фальшивые или настоящие)

Способ решения – классификация.
Методклассификации – SVM (Support Vector Machine).
Положительные стороны SVM:
- быстрый метод классификации;
- метод сводится к решению задачи квадратичного программирования в выпуклой области, которая обычно имеет единственное решение;
- метод позволяет осуществлять более уверенную классификацию, чем другие линейные методы.

Практическое задание

1. Проанализировать разные результаты для набора данных BanknoteAuthentification, в чём разница базовых настроек алгоритма в разных инструментах?
2. Найти наилучшие параметры для данных BanknoteAuthentification, используя технику кросс-валидации.
3. Возможно ли улучшить точность алгоритма для данных AdultIncome, используя другие параметры (gamma, C, параметры, связанные с SVM)? Подобрать параметры, дающие большую точность или показать, что для большого набора параметров точность улучшить не удаётся.