Линейные модели классификации

Пусть $X = \boldsymbol{R}^d$ – пространство объектов, $Y = \{-1, +1\}$ – множество допустимых ответов, $X = \{(x_i, y_i)\}_{i=1}^{l}$ – обучающая выборка.

Линейная модель классификации определяется следующим образом:

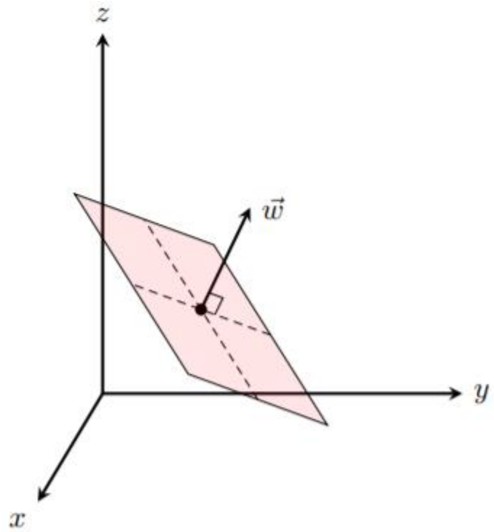
$$a(x) = \operatorname{sign}\left(\langle w, x\rangle + w_0\right) = \operatorname{sign}\left(\sum_{j=1}^{d} w_j x_j + w_0\right),$$

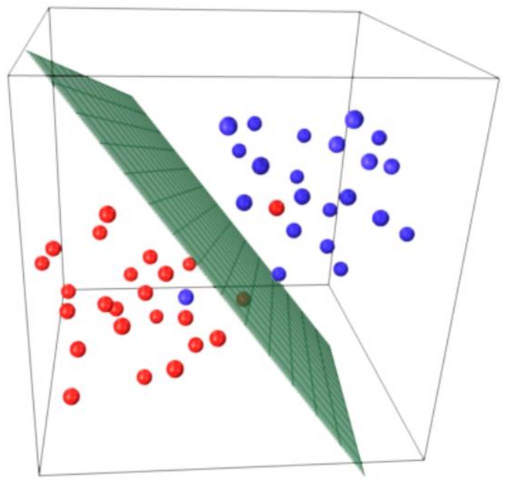
где ω – вектор весов, ω_0 – *сдвиг*.

Если не сказано иначе, мы будем считать, что среди признаков есть константа, $x_{d+1} = 1$. В этом случае нет необходимости вводить сдвиг ω_0, и линейный классификатор можно задавать как

$$a(x) = \operatorname{sign}\langle w, x\rangle.$$

Выражение $\langle \omega, x \rangle = 0$ является уравнением некоторой плоскости в пространстве признаков.





При этом для точек по одну сторону от этой плоскости скалярное произведение $\langle \omega, x \rangle$ будет положительным, а с другой — отрицательным. Таким образом, линейный классификатор проводит плоскость в пространстве признаков и относит объекты по разные стороны от нее к разным классам.

Согласно геометрическому смыслу скалярного произведения, расстояние от конкретного объекта, который имеет признаковое описание x, до гиперплоскости $\langle \omega, x \rangle = 0$ равно $\frac{|\langle \omega, x \rangle|}{\|\omega\|}$. С этим связано такое важное понятие *отступа* в задачах линейной классификации.

Обучение линейного классификатора

В случае линейной классификации естественный способ определить качество того или иного алгоритма — вычислить для объектов обучающей выборки долю правильных ответов

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i].$$

$a(x_i) = y_i$ — метка класса определенная алгоритмом совпадает с истинной меткой класса.

Нам будет удобнее решать задачу минимизации, поэтому будем вместо этого использовать долю неправильных ответов:

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i] = \frac{1}{\ell} \sum_{i=1}^{\ell} [\text{sign}\langle w, x_i \rangle \neq y_i] \to \min_{w} \qquad (1.1)$$

Введем новую величину $M_i = y_i\langle \omega, x_i \rangle$ – отступ ($margin$).

$$Q(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}[\underbrace{y_i\langle w, x_i\rangle}_{M_i} < 0] \to \min_w$$

Знак отступа говорит о корректности ответа классификатора (положительный отступ соответствует правильному ответу, отрицательный — неправильному), а его абсолютная величина характеризует степень уверенности классификатора в своём ответе. Напомним, что скалярное произведение $\langle \omega, x \rangle$ пропорционально расстоянию от разделяющей гиперплоскости до объекта; соответственно, чем ближе отступ к нулю, тем ближе объект к границе классов, тем ниже уверенность в его принадлежности.

Функционал (1.1) оценивает ошибку алгоритма на объекте x с помощью *пороговой функции потерь* $L(M) = [M < 0]$, где аргументом функции является отступ M. Оценим эту функцию сверху: $L(M) \leqslant \tilde{L}(M)$.

После этого можно получить верхнюю оценку на функционал (1.1):

$$Q(a, X) \leqslant \frac{1}{\ell} \sum_{i=1}^{\ell} \tilde{L}(y_i \langle w, x_i \rangle) \to \min_w$$

Если верхняя оценка $\tilde{L}(M)$ является гладкой, то и данная верхняя оценка будет гладкой. В этом случае её можно будет минимизировать с помощью, например, *градиентного спуска*. Если верхнюю оценку удастся приблизить к нулю, то и доля неправильных ответов тоже будет близка к нулю.

Приведем несколько примеров верхних оценок:

1. $\tilde{L}(M) = \log\left(1 + e^{-M}\right)$ — логистическая функция потерь
2. $\tilde{L}(M) = (1 - M)_+ = \max(0, 1 - M)$ — кусочно-линейная функция потерь (используется в методе опорных векторов)
3. $\tilde{L}(M) = (-M)_+ = \max(0, -M)$ — кусочно-линейная функция потерь (соответствует персептрону Розенблатта)
4. $\tilde{L}(M) = e^{-M}$ — экспоненциальная функция потерь
5. $\tilde{L}(M) = 2/(1 + e^{M})$ — сигмоидная функция потерь

Любая из них подойдёт для обучения линейного классификатора.

В случае логистической функции потерь функционал ошибки имеет вид:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\ln\left(\exp(-M_i)\right) = \frac{1}{\ell}\sum_{i=1}^{\ell}\ln\left(\exp(-y_i\langle w, x_i\rangle)\right).$$

Получившееся выражение является гладким, а, следовательно, можно использовать, например, метод градиентного спуска. Следует обратить внимание, что в случае, если число ошибок стало равно нулю, все равно в ходе обучения алгоритма линейной классификации будут увеличиваться отступы, то есть будет увеличиваться уверенность в полученных результата