Сафиуллин Амир

Решение БДЗ содержит следующие методы:

Алгоритм 1. vlob(plus, minus, x_test, y_test)

Алгоритм основан на нормированной сумме мощности пересечения признаков неизвестного примера с примерами-(+) и примерами-(-).

То же самое для отрицательных.

$$\text{Pos} = \frac{1}{|G^+|} \sum_{i \in G^+} |g' \cap g_i^+|$$

$$\text{Neg} = \frac{1}{|G^-|} \sum_{i \in G^-} |g' \cap g_i^-|$$

Неизвестный пример относится к тому набору, где эта сумма больше, т.е. если Pos > Neg то положительно классифицируем, иначе отрицательно.

Алгоритм 2. is_in_intent(plus, minus, x_test, y_test)

Пересекаем с положительным и проверяем чтобы пересечение не вкладывалось ни в одно отрицательное. если все так, то начисляем голос в виде "относительной мощности пересечения".

То же самое для отрицательных.

$$\text{Pos} = \frac{1}{|G^+|} \sum_{i \in G^+} \begin{cases} \frac{1}{|g'|} |g' \cap g_i^+|, & \text{если } |(g' \cap g_i^+) \cap g_k^-| == 0, k \in G^- \\ 0, & \text{Иначе} \end{cases}$$

$$\text{Neg} = \frac{1}{|G^-|} \sum_{i \in G^-} \begin{cases} \frac{1}{|g'|} |g' \cap g_i^-|, & \text{если } |(g' \cap g_i^-) \cap g_k^+| == 0, k \in G^+ \\ 0, & \text{Иначе} \end{cases}$$

Где сумма накопленных "голосов" больше - туда и классифицируем, т.е. если Pos > Neg то положительно классифицируем, иначе если Neg > Pos то классифицируем отрицательно, иначе, т.е. в случае равенства смотрим по поддержке, как в алгоритме 3.2 (с порогом 10.)

Алгоритм 3.1. is_in_int1(plus, minus, x_test, y_test)

Для неизвестного примера начисляем голос пропорционально "относительной мощности

пересечения" пересечения (признаков неизвестного примера и (+)примера) с остальными (+)-примерами, если это пересечение вкладывается в них.

Для минуса то же самое.

$$\text{Pos} = \frac{1}{|G^+|}\sum_{i\in G^+}\frac{1}{|g_i^+||G^+|}\left|(g' \cap g_i^+) \cap g_k^+\right|,\ k \in G^+, k != i$$

$$\text{Neg} = \frac{1}{|G^-|}\sum_{i\in G^-}\frac{1}{|g_i^-||G^-|}\left|(g' \cap g_i^-) \cap g_k^-\right|,\ k \in G^-, k != i$$

Пример классифицируется туда, где сумма «голосов» больше.

Алгоритм 3.2. is_in_int2(plus, minus, x_test, y_test)

Лучший порог = 10; лучший порог искался с помощью скользящего контроля максимизуруя accuracy

Для неизвестного примера начисляем голоса пропорционально поддержке в положительных примерах и мощности пересечения с положительным примером, если она больше порога.

Для минуса то же самое.

$$\text{Pos} = \frac{1}{|G^+|}\sum_{i\in G^+}\begin{cases}\frac{|g' \cap g_i^+|}{|g_i^+||G^+|}\left|(g' \cap g_i^+)^+\right|, \text{если } \left|(g' \cap g_i^+)^+\right| > tresh \\ \text{Иначе } 0\end{cases}$$

$$\text{Neg} = \frac{1}{|G^-|}\sum_{i\in G^-}\begin{cases}\frac{|g' \cap g_i^-|}{|g_i^-||G^-|}\left|(g' \cap g_i^-)^-\right|, \text{если } \left|(g' \cap g_i^-)^-\right| > tresh \\ \text{Иначе } 0\end{cases}$$

Пример классифицируется туда, где сумма «голосов» больше.

Алгоритм 4. costs_and_penalty(plus, minus, x_test, y_test, tresh)

Объединение алгоритмов 2 и 3.1. Начисляем голоса, вдобавок задаем штраф (отнимаем голоса) для неизвестного примера если пересечение с плюс примером вкладывается в отрицательный пример и штраф если пересечение с минус примером вкладывается в положительный.

Pos $=$

$$\frac{1}{|G^+|}\sum_{i\in G^+}\begin{cases}\frac{1}{|g_i^+|}\left|g' \cap g_i^+\right|, \text{если } \left|(g' \cap g_i^+)^-\right| == 0 \\ -\frac{1}{|G^-|}\frac{1}{|G^+|}\frac{1}{|g_i^+|}\frac{1}{|G^+|}\left|(g' \cap g_i^+)^+\right| * \left|(g' \cap g_i^+)^-\right| * \left|g' \cap g_i^+\right|, \text{Иначе если } \left|(g' \cap g_i^+)^+\right| > tresh\end{cases}$$

Neg $=$

$$\frac{1}{|G^-|}\sum_{i\in G^-}\begin{cases}\frac{1}{|g_i^-|}\left|g' \cap g_i^-\right|, \text{если } \left|(g' \cap g_i^-)^+\right| == 0 \\ -\frac{1}{|G^-|}\frac{1}{|G^-|}\frac{1}{|g_i^-|}\frac{1}{|G^+|}\left|(g' \cap g_i^-)^-\right| * \left|(g' \cap g_i^-)^+\right| * \left|g' \cap g_i^-\right|, \text{Иначе если } \left|(g' \cap g_i^-)^-\right| > tresh\end{cases}$$

Классифицируем туда, где сумма голосов больше.

Ниже приведена таблица со средними значениями accuracy для каждого алгоритма (1-4), и для трех наиболее популярных — SVC, Random forests, k-Nearest Neighbor.

Алгоритмы	**Среднее значение по accuracy (на tic tac)**
1	0,659
2	0,99
3.1	0,933
3.2	0,979
4	0,989
SVC	0,998
Random Forests	0,986
kNN	0,982

Алгоритмы	**Среднее значение по accuracy (на kr vs kp)**
1	0,799
2	0,979