СочиСириус-2016/Направление BigData. Проект «Медицинская диагностика по ЭКГ»

Воронцов К. В.

17 июля 2016 г.

Содержание

1 Введение **1**

2 Ход проекта **3**

3 Задания **4**
3.1 Применение готовых методов машинного обучения 4
3.2 Наивный линейный классификатор с отбором признаков 4
3.3 Альтернативные схемы кодирования 6
3.4 Поиск информативных словарных признаков классов 6
3.5 Учёт влияния индивидуальных особенностей людей 7
3.6 Метод стохастического градиента 7
3.7 Метод стохастического градиента для максимизации AUC 9

1 Введение

Технология информационного анализа электрокардиосигналов, разработанная проф. д.м.н. В. М. Успенским [1], позволяет диагностировать заболевания внутренних органов человека по электрокардиограмме. Задачей проекта является улучшение качества диагностики с помощью машинного обучения.

Исходные данные для выполнения проекта предоставлены автором диагностической методики В. М. Успенским. Это выборка 2515 обследований с подтверждёнными диагнозами по пяти заболеваниям: сахарный диабет, язвенная болезнь, узловой зоб щитовидной железы, ишемическая болезнь сердца, вегетососудистая дистония[1].

Классы. Обозначим через Y множество классов — диагностируемых заболеваний. Особым «нулевым» элементом $0 \in Y$ будем обозначать класс здоровых людей. Для каждого обследования x_i из *обучающей выборки* $X = \{x_1, \ldots, x_\ell\}$ известно множество классов $Y_i \subset Y$. Если обследуемый здоров, то $Y_i = \{0\}$; если у него имелось хотя бы одно заболевание, то $0 \notin Y_i$. Для каждого класса $y \in Y$ введём множество объектов обучающей выборки $X_y = \{x_i \in X : y \in Y_i\}$.

[1] Это лишь часть данных. Система «Скринфакс» диагностирует более 40 заболеваний. Исследования по расширению спектра заболеваний в настоящее время продолжаются.

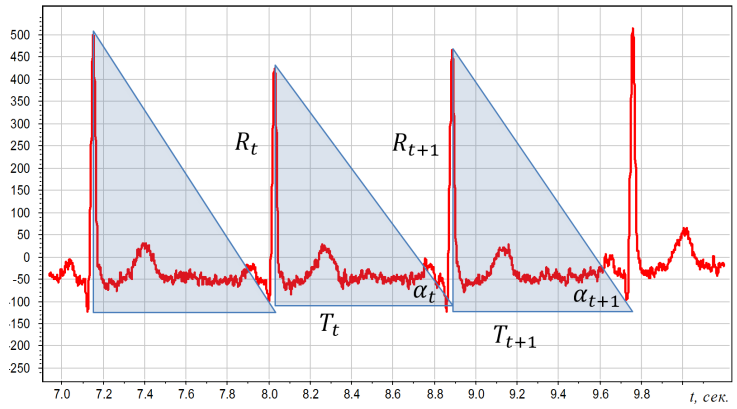

Рис. 1: Пример электрокардиограммы. Два последовательных кардиоцикла с амплитудами R_t, R_{t+1}, интервалами T_t, T_{t+1} и «фазовыми углами» α_t, α_{t+1}.

Предварительная обработка электрокардиограммы включает три этапа: *демодуляция* — обнаружение R-пиков, определение их амплитуд и RR-интервалов между R-пиками; *дискретизация* — преобразование последовательности интервалов и амплитуд в символьную последовательность (кодограмму); *векторизация* — преобразование кодограммы в вектор частот триграмм. Полученные векторы используются в качестве признаковых описаний объектов в алгоритмах машинного обучения.

Демодуляция. На первом этапе предварительной обработки электрокардиограмма обследования x преобразуется в последовательность пар $(R_1, T_1), \ldots, (R_N, T_N)$, где R_t — амплитуда R-пика t-го кардиоцикла, T_t — длина RR-интервала между R-пиками последовательных кардиоциклов, рис. 1. Также вводится отношение этих двух величин, $\alpha_t = \operatorname{arctg} R_t/T_t$. Последовательность $R_1, \ldots, R_N$ называется *амплитудограммой*, последовательность $T_1, \ldots, T_N$ — *интервалограммой* исходной ЭКГ. В текущей реализации диагностической системы $N = 600$.

Дискретизация. Диагностическую ценность имеют не сами величины R_t, T_t, α_t, а знаки их приращений в последовательных кардиоциклах:

$$dR_t = R_{t+1} - R_t, \qquad dT_t = T_{t+1} - T_t, \qquad d\alpha_t = \alpha_{t+1} - \alpha_t \tag{1}$$

Возможны лишь 6 из 8 сочетаний знаков приращений этих трёх величин. Они кодируются буквами 6-символьного алфавита $\mathcal{A} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$, рис. 2. Таким образом, электрокардиограмма обследования x преобразуется в последовательность «$s_1 \ldots s_{N-1}$» символов алфавита $\mathcal{A}$, называемую *кодограммой*. Цепочки символов алфавита $\mathcal{A}$ называются *кодовыми словами* или просто словами. Слова длины k называются также k-граммами. Число всех возможных k-грамм равно $|\mathcal{A}|^k$. В текущей реализации диагностической системы используются триграммы. Множество всех триграмм образует словарь из $n = 6^3 = 216$ слов. Будем обозначать слова словаря через $v_j = «v_{j1} \ldots v_{jk}»$, $v_{ji} \in \mathcal{A}$, $j = 1, \ldots, n$.

Векторизация. Частота k-граммы v_j в кодограмме обследования x определяется как отношение числа вхождений слова v_j в кодограмму к общему числу слов длины k

$$
\begin{aligned}
s_t = \mathtt{A}: &\quad R_t < R_{t+1}, \quad T_t < T_{t+1}, \quad \alpha_t < \alpha_{t+1};\\
s_t = \mathtt{B}: &\quad R_t \geqslant R_{t+1}, \quad T_t \geqslant T_{t+1}, \quad \alpha_t < \alpha_{t+1};\\
s_t = \mathtt{C}: &\quad R_t < R_{t+1}, \quad T_t \geqslant T_{t+1}, \quad \alpha_t < \alpha_{t+1};\\
s_t = \mathtt{D}: &\quad R_t \geqslant R_{t+1}, \quad T_t < T_{t+1}, \quad \alpha_t \geqslant \alpha_{t+1};\\
s_t = \mathtt{E}: &\quad R_t < R_{t+1}, \quad T_t < T_{t+1}, \quad \alpha_t \geqslant \alpha_{t+1};\\
s_t = \mathtt{F}: &\quad R_t \geqslant R_{t+1}, \quad T_t \geqslant T_{t+1}, \quad \alpha_t \geqslant \alpha_{t+1}.
\end{aligned}
$$

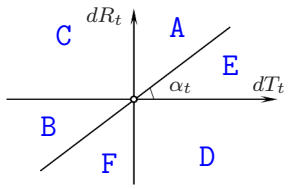

Рис. 2: Кодирование интервалов и амплитуд в символьную строку — кодограмму.

```
DBFEACFDAAFBABDDAADFAAFFEACFEACFBAEFFAABFFAAFFAAFFAAFFAAEBFAEBFEAAFCAAFFAAD
FCAFFAADFCADFCCDFDACFFACDFAEFFACFFEADFCAFBCADFFECFFAAFFAAFFAEFFCACFCAEFFCAD
DAADBFAAFFAEBFAABFACDFFAAFBAADFAADFDAAFCECFCEDFCEEFCAEFBECBBBAADBAACFFAAFFA
CFFCECFDAABDAEFFAAFFCEDBFAAFFAEFFAEFBACFBAEDFEAAFFCAFFDAAFFAEBDAADBBADFDAFF
EABFCCAFDEEBDECFFACFFAABFAADFBAAFFACFFFAEFFACFFACFFCECFBAAFFFAAFFFAAFFAADFB
AABFACDFDAEFFAADBAAEFFEAFBCECFDECCFBAAFFAADFDACDFAAFFAADFCAADFAEFBAAFFCADFE
AFFCECFCECFFAAFFABCFDAAAFFADBFCAEFFAABFACBFAAEBFAEBFCAFFBAAFFAAFFDACFDAABFB
CAFFAECFFACFFACDFCADFDAABFAAEDDABBFCACDBAAFFAAFFCADFAADFDACFFAEDFCACFCAEBCE
```

1. FFA - 42	17. EFF - 10	33. CEC - 6	49. EAC - 3
2. FAA - 33	18. DAA - 10	34. ADB - 5	50. DDA - 3
3. AFF - 32	19. ECF - 9	35. FFE - 5	51. CAC - 3
4. AAF - 30	20. FFC - 9	36. EBF - 5	52. EDF - 3
5. ADF - 18	21. FEA - 9	37. CFD - 5	53. EFB - 3
6. FCA - 18	22. DFC - 8	38. AFB - 4	54. DBA - 3
7. ACF - 17	23. ABF - 8	39. AAE - 4	55. FCC - 2
8. AAD - 15	24. AAB - 8	40. CFC - 4	56. AFC - 2
9. CFF - 14	25. FCE - 8	41. CAE - 4	57. EAA - 2
10. AEF - 13	26. AEB - 7	42. DAC - 4	58. CED - 2
11. FDA - 13	27. DFD - 7	43. DBF - 4	59. CAA - 2
12. FAE - 12	28. ACD - 6	44. BFC - 4	60. BCA - 2
13. FAC - 12	29. CDF - 6	45. CFB - 4	61. BBA - 2
14. FBA - 11	30. DFA - 6	46. AED - 3	62. DFF - 2
15. BFA - 11	31. CAF - 6	47. FFF - 3	63. BDA - 2
16. BAA - 11	32. CAD - 6	48. FBC - 3	64. DAE - 2

Рис. 3: Пример кодограммы и её векторного представления.

в кодограмме, которое равно $N - k$:

$$f_j(x) = \frac{1}{N-k} \sum_{t=1}^{N-k} \prod_{i=1}^{k} \big[s_{t+i-1} = v_{ji}\big]. \tag{2}$$

Таким образом, кодограмма обследования x преобразуется в вектор признаков $\big(f_j(x), \ldots, f_n(x)\big)$. Он имеет фиксированную размерность n, что и позволяет применять методы машинного обучения к обучающей выборке векторизованных электрокардиограмм. Пример кодограммы и её векторного представления показан на рис. 3.

2 Ход проекта

Проект имеет две фазы. По окончании проекта команда готовит доклад с презентацией для научной конференции проектной смены.

Первая фаза — соревновательная. Каждый участник проекта осваивает приёмы программирования на Python, изучает алгоритмы анализа данных и машинного обучения и реализует собственный диагностический алгоритм. Можно использовать как готовые методы машинного обучения, так и разрабатывать собственные. Для сравнения решений участников организовано соревнование на платформе Kaggle in Class.

Вторая фаза — кооперативная. Задания для первой фазы подобраны таким образом, что решения участников могут комбинироваться в любых сочетаниях. В момент перехода проекта из первой фазы во вторую организуется мозговой штурм, в ходе которого участники проекта рассказывают друг другу свои решения, генерируют новые идеи и договариваются, какие комбинации решений они будут реализовывать на протяжении второй фазы проекта.

Целью проекта является улучшение качества диагностики. Это типичное прикладное исследование на стыке медицинской диагностики и машинного обучения, в котором предлагаемые комбинированные решения сравниваются с базовыми вариантами диагностических алгоритмов.

3 Задания

3.1 Применение готовых методов машинного обучения

Пакет `scikit-learn` языка Python содержит много готовых методов машинного обучения. Участникам проекта предоставляется программа, которая настраивает один из стандартных методов и формирует данные для загрузки решения на Kaggle.

Задание. Протестировать как можно больше готовых методов: LR, SVM, SVM-RBF, xgboost, ElasticNet и др. При этом необходимо разобраться, какие параметры имеет каждый метод, в чём их содержательный смысл, в каком диапазоне стоит перебирать значения каждого из этих параметров, и как оценивать качество решения самостоятельно, не загружая его в Kaggle.

3.2 Наивный линейный классификатор с отбором признаков

Линейными моделями классификации называются диагностические правила, в которых положительное решение по заболеванию $y \in Y$ принимается по значению взвешенной суммы признаков:

$$a_y(x) = \big[\text{score}(x, w_y) \geqslant w_{0y}\big], \qquad \text{score}(x, w_y) = \sum_{j=1}^{n} w_{yj} f_j(x), \tag{3}$$

где $f_j(x)$ — j-й признак обследования x; $w_y = (w_{y1}, \ldots, w_{yn})$ — вектор весов признаков; w_{0y} — порог принятия диагностического решения; $\text{score}(x, w_y)$ — линейная оценка принадлежности объекта x классу y.

Признаками могут быть как частоты триграмм (2), так и другие функции, вычисляемые по данным обследования x. В частности, в качестве признаков можно использовать бинаризованные частоты k-грамм $\big[f_j(x) \geqslant A\big]$, называемые также *встречаемостями* k-грамм. Экспериментально установлено, что при длине записи $N = 600$ значение параметр $A = \frac{2}{N-k}$ является оптимальным. Можно также подбирать этот параметр отдельно для каждого заболевания.

При обучении диагностического правила для болезни y веса признаков w_{yj} оптимизируются по обучающей выборке.

Диагностическим эталоном заболевания y называется множество триграмм, для которых оптимальные веса определены как положительные: $E_y = \{j\colon w_{yj} > 0\}$.

Наивный линейный классификатор основан на эвристических формулах весов, определяемых для каждого признака независимо. Определим долю обучающих объектов класса y, для которых $f_j(x) \geqslant A$ и для которых $f_j(x) < A$:

$$N_{y1}^j = \frac{\sum_{i=1}^{\ell}[y \in Y_i]\,[f_j(x) \geqslant A]}{\sum_{i=1}^{\ell}[y \in Y_i]};$$
$$N_{y0}^j = \frac{\sum_{i=1}^{\ell}[y \in Y_i]\,[f_j(x) < A]}{\sum_{i=1}^{\ell}[y \in Y_i]};$$

в частности, N_{01}^j — доля здоровых, у которых j-я триграмма частая.

Разумно предположить, что вес w_{yj} должен быть тем больше, чем больше N_{y1}^j и N_{00}^j и чем меньше N_{y0}^j и N_{01}^j. Можно пробовать разные формулы весов:

$$w_{yj} = \frac{N_{y1}^j}{N_{01}^j}, \qquad w_{yj} = \frac{N_{y1}^j N_{00}^j}{N_{y0}^j N_{01}^j},$$
$$w_{yj} = \log\frac{N_{y1}^j}{N_{01}^j}, \qquad w_{yj} = \log\frac{N_{y1}^j N_{00}^j}{N_{y0}^j N_{01}^j},$$
$$w_{yj} = \sqrt{N_{y1}^j} - \sqrt{N_{01}^j}, \qquad w_{yj} = \sqrt{N_{y1}^j N_{00}^j} - \sqrt{N_{y0}^j N_{01}^j}.$$

С помощью этих формул диагностическая модель для болезни y обучается по выборке объектов двух классов — больных y и здоровых 0.

Возможна и другая стратегия обучения, когда классификатор обучается отличать больных класса y не только от здоровых, но и от больных любыми другими болезнями. Для этого достаточно заменить в формулах N_{00}^j на $N_{\bar{y}0}^j$ и N_{01}^j на $N_{\bar{y}1}^j$:

$$N_{\bar{y}1}^j = \frac{\sum_{i=1}^{\ell}[y \notin Y_i]\,[f_j(x) \geqslant A]}{\sum_{i=1}^{\ell}[y \notin Y_i]};$$
$$N_{\bar{y}0}^j = \frac{\sum_{i=1}^{\ell}[y \notin Y_i]\,[f_j(x) < A]}{\sum_{i=1}^{\ell}[y \notin Y_i]};$$

Какую именно стратегию многоклассовой классификации выбрать, зависит от целей исследования.

Отбор признаков — это задача поиска оптимального подмножества признаков, обеспечивающего наилучшее качество диагностики. В общем случае это требует решения 2^n задач для каждого из подмножеств признаков. Однако на практике неплохо зарекомендовали себя простые эвристические способы отбора признаков. Самый простой из них — *сортировка с отсечением*: признаки сортируются по убыванию значений w_{yj}, признаки с самыми близкими к нулю весовыми коэффициентами отбрасываются, остаются только K признаков. Обычно выборка делится на обучающую и тестовую; веса признаков вычисляются по обучающей выборке; выбор оптимального числа признаков K производится по оценке качества на тестовой выборке. Можно использовать для сортировки одну формулу весов признаков, а в линейном классификаторе — другую. Параметры метода — число K и тип формулы весов для сортировки и для классификатора — подбираются экспериментально.

Задание. Реализовать линейный наивный классификатор с отбором признаков, с перебором вариантов формул весов и с подбором параметра K.

3.3 Альтернативные схемы кодирования

Предлагается наряду с приращениями (1) ввести новые величины, измеряющие соотношения между интервалами и/или амплитудами не двух, а трёх соседних кардиоциклов.

В частности, приращения

$$d'R_t = R_{t+2} - R_t, \qquad d'T_t = T_{t+2} - T_t, \qquad d'\alpha_t = \alpha_{t+2} - \alpha_t$$

несут дополнительную информацию о том, приводит ли знакопеременное изменение соответствующей величины (последовательное уменьшение–увеличение или увеличение–уменьшение) к её суммарному уменьшению или увеличению.

Другой вариант приращений

$$d''R_t = dR_{t+1} - dR_t, \qquad d''T_t = dT_{t+1} - dT_t, \qquad d''\alpha_t = d\alpha_{t+1} - d\alpha_t$$

несёт дополнительную информацию о выпуклости или вогнутости соответствующей величины как функции времени на двух последовательных интервалах.

Добавление одного знака приращения в схему кодирования приводит к увеличению мощности алфавита не более чем вдвое. Теоретическая часть задания состоит в том, чтобы аккуратно рассмотреть все варианты и проверить, нет ли среди них запрещённых, и какова минимальная мощность алфавита, необходимая для кодирования всех допустимых комбинаций приращений.

При выборе новых способов кодирования можно опираться на следующие ранее полученные эмпирические наблюдения: вариабельность интервалов T_t несёт основную диагностическую информацию; добавление к ней данных о вариабельности амплитуд R_t даёт небольшой прирост качества диагностики (1–2%); добавление к ней данных о вариабельности углов α_t даёт заметный прирост качества до 7%.

Введение в схему кодирования величин, охватывающих три интервала, может сокращать необходимый объём словаря. Например, введение $d'T_t$ или $d''T_t$ расширяет алфавит до 12 символов. Биграммы в этом алфавите охватывают четыре кардиоцикла точно так же, как триграммы в 6-символьном алфавите. Однако при использовании 12-символьных биграмм вместо 6-символьных триграмм объём словаря сокращается с $6^3 = 216$ до $12^2 = 144$.

Задание. Реализовать на выбор предложенные схемы кодирования или свои собственные. Сформировать по ним частотные признаковые описания обследований обучающей выборки. Протестировать различные методы машинного обучения на признаковых описаниях, полученных с помощью альтернативных схем кодирования.

3.4 Поиск информативных словарных признаков классов

В биоинформатике активно используются алгоритмы поиска общих подстрок в массивах символьных последовательностей. При этом совпадение подстрок может

пониматься с точностью до заданного числа вставок и замен. Предлагается использовать готовые алгоритмы для поиска подстрок, часто встречающихся в кодограммах больных класса y, но редко или вообще никогда не встречающихся в кодограммах «нулевого» класса здоровых людей. Допустим, мы нашли множество S_y таких подстрок для каждого класса y. Наличие подстроки $s \in S_y$ в кодограмме обследования x (с точностью до заданного числа вставок и замен) является хорошим бинарным признаком для диагностики класса y. Будем обозначать его $f_s(x)$ и называть *словарным признаком* данного класса.

Задание. Расширить признаковое описание обследования x словарными признаками $f_s(x)$ для всех подстрок $s \in S_y$ и классов $y \in Y$. Протестировать различные методы машинного обучения на признаковых описаниях со словарными признаками классов, найденными путём поиска информативных подстрок классов.

3.5 Учёт влияния индивидуальных особенностей людей

В. М. Успенский выдвинул гипотезу, что распределение частот букв является скорее индивидуальной особенностью обследуемого человека, чем характеристикой его заболеваний. В своём алгоритме он использует «коэффициент гармонии» $g(x)$, равный отношению числа символов $\{\mathtt{A}, \mathtt{B}, \mathtt{E}, \mathtt{F}\}$ к числу символов $\{\mathtt{C}, \mathtt{D}\}$ в кодограмме x. Экспериментально замечено, что диагностические эталоны заболеваний различаются у людей с низкими и высокими значениями $g(x)$. Предлагается обобщить эту гипотезу, и для произвольного подмножества символов $B \subset \mathcal{A}$ ввести функцию $g_B(x)$ как долю символов из B в кодограмме обследования x.

Предполагая, что при низких значениях $g_B(x)$ важны одни признаки, а при высоких — другие, введём вместо каждого признака $f_j(x)$ составную конструкцию

$$f'_j(x) = f_j(x)\big((1-\gamma)g_B(x) + (1 - g_B(x))\gamma\big),$$

где γ — параметр модели. Существуют два подхода к его оптимизации.

Первый вариант — подобрать по сетке значение γ, одинаковое для всех признаков.

Второй вариант — предоставить методу классификации возможность определить параметр $\gamma = \gamma_j$ индивидуально для каждого признака f_j. В линейной модели для этого достаточно удвоить число признаков, добавив в модель вместе с каждым признаком $f_j(x)$ второй признак $f_j(x)g_B(x)$.

Задание. Реализовать и проверить оба подхода к оптимизации параметра γ. Провести оптимизацию множества B, проверив гипотезу В. М. Успенского, что оптимальным выбором является подмножество $B = \{\mathtt{C}, \mathtt{D}\}$.

3.6 Метод стохастического градиента

Рассмотрим линейную модель классификации (3). Для классификации объекта x вычисляются линейные оценки $\mathrm{score}(x, w_y)$ принадлежности объекта x каждому из классов y. Пока не будем использовать пороги, полагая $w_{0y} = 0$. Медицинская диагностика — это задача с пересекающимися классами: каждый объект может принадлежать нескольким классам одновременно.

Для обучения линейного классификатора необходимо найти вектор весов w_y для каждого класса $y \in Y$, пользуясь данными из обучающей выборки.

Введём понятие *отступа* объекта x_i относительно класса y:

$$M_i(w_y) = \text{score}(x_i, w_y)\big([y \in Y_i] - [y \notin Y_i]\big) = \begin{cases} +\text{score}(x_i, w_y), & y \in Y_i; \\ -\text{score}(x_i, w_y), & y \notin Y_i. \end{cases}$$

Знак отступа показывает, есть ли ошибка на объекте x_i относительно класса y. Если $M_i(w_y) > 0$, то объект правильно относится или не относится к классу y. Если $M_i(w_y) < 0$, то объект ошибочно относится или не относится к классу y. Абсолютная величина отступа показывает, насколько далеко объект x_i находится от границы класса y. Чем выше значение отступа, тем надёжнее объект классифицируется относительно класса y.

Введём гладкую *функцию потерь* $\mathscr{L}(M)$ как убывающую функцию отступа M. Поставим оптимизационную задачу обучения линейной модели классификации — найти такие значения весов w_{yj}, чтобы суммарные потери были минимальны:

$$Q(w) = \sum_{i=1}^{\ell} \sum_{y \in Y} \mathscr{L}\big(M_i(w_y)\big) \to \min_w .$$

Для минимизации $Q(w)$ воспользуемся методом стохастического градиента. Это итерационный процесс, на каждом шаге которого векторы w_y немного изменяются в направлении наискорейшего убывания случайно взятого i-го слагаемого.

Вход: выборка $(x_i, Y_i)_{i=1}^{\ell}$;
Выход: веса w_{yj}, $y \in Y$, $j = 1 \ldots, n$;

1 инициализировать веса w_{yj}, $y \in Y$, $j = 1, \ldots, n$;
2 **повторять**
3 выбрать случайный объект (x_i, Y_i) из обучающей выборки;
4 **для всех** $y \in Y$
5 выбрать величину градиентного шага h_y;
6 вычислить отступ $M_i(w_y)$;
7 выполнить градиентный шаг:
 $w_{yj} := w_{yj} - h_y \mathscr{L}'\big(M_i(w_y)\big) f_j(x_i)\big([y \in Y_i] - [y \notin Y_i]\big)$, $j = 1, \ldots, n$;
8 **пока** процесс не сойдётся;

Задание. Реализовать оптимизацию весов в линейном классификаторе методом стохастического градиента. Рекомендуется взять функцию $\mathscr{L}(M) = \ln(1 + e^{-M})$, используемую в логистической регрессии. Её производная $\mathscr{L}'(M) = -(1 + e^{M})^{-1}$. В качестве инициализации рекомендуется взять линейный наивный классификатор.

3.7 Метод стохастического градиента для максимизации AUC

Рассмотрим альтернативный способ обучения многоклассовой линейной модели, основанный на максимизации площади под ROC-кривой для каждого класса $y \in Y$:

$$\mathrm{AUC}_y(w_y) = \frac{\sum\limits_{i=1}^{\ell}\sum\limits_{s=1}^{\ell}\big[y \in Y_i,\ y \notin Y_s\big]\big[\mathrm{score}(x_s, w_y) < \mathrm{score}(x_i, w_y)\big]}{\sum\limits_{i=1}^{\ell}\sum\limits_{s=1}^{\ell}\big[y \in Y_i,\ y \notin Y_s\big]} \to \max_{w_y}.$$

В этом случае также возникает понятие отступа, однако теперь отступ определяется не для отдельного объекта, а для пары объектов x_i, x_s, из которых первый принадлежит классу y, а второй не принадлежит:

$$M_{is}(w_y) = \mathrm{score}(x_i, w_y) - \mathrm{score}(x_s, w_y).$$

Отрицательный отступ означает, что вектор w_y не позволяет разделить эти два объекта без ошибки. Как и в предыдущем разделе, введём гладкую убывающую функцию отступа $\mathscr{L}(M)$ и поставим оптимизационную задачу — найти такие значения весов w_{yj}, чтобы суммарные потери были минимальны:

$$Q_{\mathrm{AUC}}(w) = \sum_{i=1}^{\ell}\sum_{s=1}^{\ell}\sum_{y \in Y}\big[y \in Y_i,\ y \notin Y_s\big]\mathscr{L}\big(M_{is}(w_y)\big) \to \min_{w}.$$

Задача распадается на $|Y|$ независимых подзадач по классам, поэтому нормировочные множители были отброшены. Для минимизации $Q_{\mathrm{AUC}}(w)$ снова воспользуемся методом стохастического градиента. Только теперь на каждом шаге будем брать случайно не один объект, а два.

Вход: выборка $(x_i, Y_i)_{i=1}^{\ell}$;
Выход: веса $w_{yj},\ \ y \in Y,\ \ j = 1 \ldots, n$;

1 инициализировать веса $w_{yj},\ \ y \in Y,\ \ j = 1, \ldots, n$;
2 **повторять**
3 выбрать случайную пару объектов (x_i, Y_i), (x_s, Y_s);
4 **для всех** $y \in Y_i \setminus Y_s$
5 выбрать величину градиентного шага h_y;
6 вычислить отступы $M_{is}(w_y)$;
7 выполнить градиентный шаг:
$w_{yj} := w_{yj} - h_y\mathscr{L}'\big(M_{is}(w_y)\big)\big(f_j(x_i) - f_j(x_s)\big),\ \ j = 1, \ldots, n$;
8 **пока** процесс не сойдётся;

Задание. Реализовать оптимизацию весов в линейном классификаторе методом стохастического градиента. Рекомендуется взять функцию $\mathscr{L}(M) = \ln(1 + e^{-M})$, используемую в логистической регрессии. Её производная $\mathscr{L}'(M) = -(1 + e^{M})^{-1}$. В качестве инициализации рекомендуется взять линейный наивный классификатор.

Список литературы

[1] *Успенский В. М.* Информационная функция сердца. Теория и практика диагностики заболеваний внутренних органов методом информационного анализа электрокардиосигналов. — М.: Экономика и информатика, 2008. — 116 с.