Логистическая регрессия

Данное задание основано на материалах лекций по логистической регрессии.

Вы научитесь:

- работать с логистической регрессией
- реализовывать градиентный спуск для ее настройки
- использовать регуляризацию

Введение

Логистическая регрессия — один из видов линейных классификаторов. Одной из ее особенностей является возможность оценивания вероятностей классов, тогда как большинство линейных классификаторов могут выдавать только номера классов.

Логистическая регрессия использует достаточно сложный функционал качества, который не допускает записи решения в явном виде (в отличие от, например, линейной регрессии). Тем не менее, логистическую регрессию можно настраивать с помощью градиентного спуска.

Мы будем работать с выборкой, содержащей два признака. Будем считать, что ответы лежат в множестве {-1, 1}. Для настройки логистической регрессии мы будем решать следующую задачу:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\log\left(1+\exp(-y_i(w_1x_{i1}+w_2x_{i2}))\right)+\frac{1}{2}C\|\left(w\right)\|^2\to\min_{w_1,w_2}$$

Здесь x_{i1} и x_{i2} — значение первого и второго признаков соответственно на объекте x_i. В этом задании мы будем рассматривать алгоритмы без свободного члена, чтобы упростить работу.

Градиентный шаг для весов будет заключаться в одновременном обновлении весов w_1 и w_2 по следующим формулам (проверьте сами, что здесь действительно выписана производная нашего функционала):

$$w_1 := w_1 + k\frac{1}{\ell}\sum_{i=1}^{\ell} y_i x_{i1}\left(1 - \frac{1}{1+\exp(-y_i(w_1 x_{i1} + w_2 x_{i2}))}\right) - kCw_1$$

$$w_2 := w_2 + k\frac{1}{\ell}\sum_{i=1}^{\ell} y_i x_{i2}\left(1 - \frac{1}{1+\exp(-y_i(w_1 x_{i1} + w_2 x_{i2}))}\right) - kCw_2$$

Здесь k — размер шага.

Линейные методы могут переобучаться и давать плохое качество из-за различных проблем в данных: мультиколлинеарности, зашумленности и т.д. Чтобы избежать этого, следует использовать регуляризацию — она позволяет понизить сложность модели и не допустить переобучения. Сила регуляризации определяется коэффициентом C в формулах, указанных выше.

Реализация в Scikit-Learn

В этом задании мы предлагаем вам самостоятельно реализовать градиентный спуск.

В качестве метрики качества будем использовать AUC-ROC (Area Under ROC-Curve). Она предназначена для алгоритмов бинарной классификации, выдающих оценку принадлежности объекта к одному из классов. По сути, значение этой метрики является агрегацией показателей качества всех алгоритмов, которые можно получить, выбирая какой-либо порог для оценки принадлежности.

В Scikit-Learn метрика AUC реализована функцией sklearn.metrics.roc_auc_score. В качестве первого аргумента ей передается вектор истинных ответов, в качестве второго — вектор с оценками принадлежности объектов к первому классу.

Материалы

- Подробнее о логистической регрессии и предсказании вероятностей с ее помощью: `https://github.com/esokolov/ml-course-msu/blob/master/ML15/lecture-notes/Sem10_linear.pdf`
- Подробнее о градиентах и градиентном спуске: `https://github.com/esokolov/ml-course-msu/blob/master/ML15/lecture-notes/Sem07_linear.pdf`

Инструкция по выполнению

1. Загрузите данные из файла data-logistic.csv. Это двумерная выборка, целевая переменная на которой принимает значения -1 или 1.
2. Убедитесь, что выше выписаны правильные формулы для градиентного спуска. Обратите внимание, что мы используем полноценный градиентный спуск, а не его стохастический вариант!
3. Реализуйте градиентный спуск для обычной и L2-регуляризованной (с коэффициентом регуляризации 10) логистической регрессии. Используйте длину шага k=0.1. В качестве начального приближения используйте вектор (0, 0).
4. Запустите градиентный спуск и доведите до сходимости (евклидово расстояние между векторами весов на соседних итерациях должно быть не больше 1e-5). Рекомендуется ограничить сверху число итераций десятью тысячами.
5. Какое значение принимает AUC-ROC на обучении без регуляризации и при ее использовании? Эти величины будут ответом на задание. В качестве ответа приведите два числа через пробел. Обратите внимание, что на вход функции roc_auc_score нужно подавать оценки вероятностей, подсчитанные обученным алгоритмом. Для этого воспользуйтесь сигмоидной функцией: $a(x) = 1/(1 + exp(-w_1x_1 - w_2x_2))$.
6. Попробуйте поменять длину шага. Будет ли сходиться алгоритм, если делать более длинные шаги? Как меняется число итераций при уменьшении длины шага?

7. Попробуйте менять начальное приближение. Влияет ли оно на что-нибудь?

Если ответом является нецелое число, то целую и дробную часть необходимо разграничивать точкой, например, 0.421. При необходимости округляйте дробную часть до трех знаков.

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке. Обратите внимание, что отправляемые файлы не должны содержать пустую строку в конце. Данный нюанс является ограничением платформы Coursera. Мы работаем над тем, чтобы убрать это ограничение.