

## Задание №2.3. Ядерный персептрон

Рассмотрим задачу бинарной классификации с метками классов  $y \in \{0,1\}$ . Персептрон использует гипотезу вида  $h_\theta(x) = g(\theta^T x)$ , где  $g(z) = \text{sign}(z)$ . Напомним, что  $\text{sign}(z) = 1$ , если  $z \geq 0$ , и ноль в противном случае. В данном упражнении мы рассмотрим алгоритм обучения персептрона, похожий на стохастический градиентный спуск, когда градиент для обновления веса  $\theta$  вычисляется на основе всего лишь одного примера из обучающей выборки. Разница будет заключаться в том, что наш алгоритм обучения будет делать всего лишь один проход по всей обучающей выборке. Правило обновления весов для указанной версии алгоритма обучения персептрона выглядит следующим образом:

$$\theta^{(i+1)} := \theta^{(i)} + \alpha \left( y^{(i+1)} - h_\theta(x^{(i+1)}) \right) x^{(i+1)},$$

где  $\theta^{(i)}$  – это значение параметров после того, как алгоритм обработал первые  $i$  примеров из обучающей выборки. В самом начале мы инициализируем  $\theta^{(0)}$  нулями.

### Вопрос №1

[1 балл]

Пусть  $K$  – это ядро Мерсера, соответствующее какой-либо трансформации  $\phi$  очень высокой размерности. Предположим, что размерность  $\phi$  настолько высока (может даже равна  $\infty$ ), что явно вычислять и работать с вектором значений  $\phi(x)$  не представляется возможным. Опишите, как бы вы использовали «трюк с ядром», чтобы заставить персептрон работать с вашим пространством признаков, задаваемым трансформацией  $\phi$ , без явного вычисления  $\phi$ .

*Замечание:* Можете не беспокоиться о свободном члене. Можете считать, что  $\phi(x)_0 = 1$ .

Ваше описание должно включать следующие пункты.

1. Каким образом вы будете (неявно) представлять вектор параметров  $\theta^{(i)}$ , в том числе и его изначальное значение  $\theta^{(0)}$ . Заметьте, что теперь  $\theta^{(i)}$  является вектором той же размерности, что и вектор признаков  $\phi(x)$ .
2. Как можно эффективно делать прогноз на новом значении  $x$ . То есть как вы будете вычислять  $h_{\theta^{(i)}}(x) = g(\theta^{(i)T} \phi(x))$ , используя ваше представление  $\theta^{(i)}$ .
3. Как нужно модифицировать правило обновления весов  $\theta$ , приведенное выше, чтобы обновлять эти веса, получив очередной пример обучающей выборки  $(x^{(i+1)}, y^{(i+1)})$ .

### Вопрос №2

[3 балла]

Реализуйте свои идеи в коде с помощью функций `initial_state`, `predict` и `update_state`, шаблоны которых приведены в файле `perceptron.py`.

Мы предоставили два ядра – линейное и гауссово. Запустите `perceptron.py`, чтобы обучить свой «ядерный» персептрон на выборке `train.csv`. Скрипт после этого протестирует полученную модель на выборке `test.csv` и сохранит результаты в файл (в том числе и графики). Добавьте оба графика в файл-отчет, подписав, какой график какому ядру соответствует.

### Вопрос №3

[1 балл]

Одно из данных вам ядер показывает очень плохую точность классификации. Какое именно и почему, как вы думаете, это происходит?