Домашнее задание №1 по курсу «Машинное обучение»: основы машинного обучения

**Скавыш Максим**

**Задание 1**

Пусть обратимая функция, X – сл. величина.  
Доказать: если для любого t > 0 , то для любого c вероятностью как минимум выполняется   
**Решение:**  
для любого t > 0 выполняется .  
Пусть обратимая функция, тогда имеет место .  
Обозначим для любого фиксированного :   
Перепишем как и так как вероятность не может быть меньше 0, то накладываем на ограничение .  
Выражение из чего следует что если для любого t > 0, , то для любого c вероятностью как минимум выполняется .

**Задание 2**

**Решение:**

Рассмотрим класс полиномиальных классификаторов:

и построим в нем классификатор, совпадающий с из условия.

Пусть тренировачная выборка  
Пусть   
Построим полином который будет принимать 0-вые значения только в , таким полиномом является полином с корнями и его можно представить в виде: .

Рассмотрим полином он всегда и принимает значение 0 только в точках , тогда классификатор вида будет принадлежать классу и совпадать с классификатором .  
**Какой вывод можно сделать о ERM-парадигме в классе пороговых полиномиальных классификаторов?**В классе пороговых полиномиальных классификаторов обязательно найдется классификатором с полиномом степени не меньше имеющий ошибку на тренировочной выборке равной 0, однако это будет приводить к переобучению и тому, что на всем D такой классификатор будет иметь большую погрешность.

**Задание 3**

1. Пусть алгоритм A выбирает наименьший прямоугольник, содержащий все точки положительного класса. Докажите, что A является реализацией ERM-алгоритма

Выполнено предположение о реализуемости найдётся такая гипотеза ∈ H, что найдётся такая гипотеза ∈ H, что

Пусть алгоритм A выбирает наименьший прямоугольник, содержащий все точки положительного класса. Тогда на тренировочной выборке гипотеза полученная алгоритмом будет иметь минимально возможную эмпирическую ошибку

ERM-алгоритм выбирает самую лучшую гипотезу по отношению к тренировочной выборке: A является реализацией ERM-алгоритма.

1. Реализуйте программу
2. Постройте график true risk в зависимости от m. Запустите программу для всех m от 1 до n (n выберите в зависимости от показателей алгоритма). Какой в среднем понадобился размер выборки, чтоб true risk 10%? 1%? 0.1%?  
   <https://github.com/MaksimSkavysh/Math/blob/master/ml-homework_1.py-Copy2.ipynb>

10% ~ m в среднем 67

1% ~ m в среднем 550  
0.1% ~ m в среднем 4000

