Конспект лекции

А.Н. Шевляков

Машинное обучение

Отбор оптимального числа признаков и ансамбли алгоритмов

# Голосование по большинству (комитет)

**Проблема**. Есть несколько алгоритмов классификации (регрессии) *С1,С2,…,Сk*, которые обладают точностью *t1,t2,…,tk.*

Можно ли из них построить новый алгоритм с большей точностью?

Основные методы:

* голосование по большинству
* взвешенное голосование
* бустинг (когда алгоритм *Ck* улучшает *Ck-1*)

# Взвешенное голосование AdaBoost

## Взвешенное голосование (на примере задачи классификации)

Пусть классификаторы *С1,С2,…,Сk* относят объект *А* к классу 1 или -1 (результат работы *Ci* для объекта *А* обозначим через *Ci(A)*).

Окончательное решение зависит от весов *wi*:

если *w1С1(A)+ w2С2(A)+…+ wkСk(A)<0,* то А из класса -1.

если *w1С1(A)+ w2С2(A)+…+ wkСk(A)>0,* то А из класса 1.

Очень похоже на линейный классификатор (здесь тоже нужно оптимально настроить веса *wi*).

По аналогии вводим понятие отступа объекта Аi с целевым признаком *yi*:

Mi =yi(*w1С1(А*i*)+ w2С2(А*i*)+…+ wkСk(А*i*)*).

Значения весов находятся из условия минимизации выражения:

где *n* – объем тренировочной выборки, *Mi* – отступ *i*-го объекта.

Как и для линейных классификаторов данное выражение мажорируется дифференцируемой функцией.

Если в качестве мажорирующей отступ функции выбрать функцию е-Mi, то получим известный алгоритм AdaBoost.

AdaBoost – это итерационный алгоритм, основанный на следующих идеях:

1**.** Вводится понятие «вес объекта», которое позволяет получить взвешенную ошибку каждого классификатора на тренировочной выборке.

2**.** На очередной итерации мы настраиваем вес *wi* при классификаторе *Ci* с минимальной взвешенной ошибкой.

3. Объекты тренировочной выборки, на которых ошибается *Ci*, на следующей итерации приобретают бОльший вес.

Таким образом, на каждой итерации мы ищем классификатор, который лучше всех классифицирует «трудные» объекты, то есть объекты с большим весом.

## AdaBoost (формальное описание)

Пусть *uj* – вес j-го объекта из тренировочной выборки. Вначале все *uj* равны 1/n (n – объем тренировочной выборки).

Ошибка *Bad(i)* классификатора *Ci –* это сумма весов объектов, которые он классифицирует неправильно.

**1.** Найти классификатор *Сl* с минимальной ошибкой *Bad(l)*.

**2.** Вычисляем новый вес классификатора *Сl*

*wl=0.5ln[(1-Bad(l))/Bad(l)].*

**3.** Обновляем веса объектов

*uj:=ujexp(-wlyjCl(Aj))*

и нормируем их:

*uj:=uj/(u1+…un)*

Шаги 1-3 нужно повторять пока:

1. не надоест;
2. точность на тестовой выборке не стабилизируется.

Итоговый ответ – это правило:

если *w1С1(A)+ w2С2(A)+…+ wkСk(A)<0,* то А из класса -1.

если *w1С1(A)+ w2С2(A)+…+ wkСk(A)>0,* то А из класса 1.

Если некоторые объекты постоянно приобретают большие веса, то это, скорее всего, выбросы (еще один способ борьбы с ними).