Школа анализа данных
Машинное обучение, часть 1
Контрольная работа

Задача 1

1. Нет, не повышается. Так как мы выдаем результат алгоритма по k ближайшим соседям, то при маленьком числе классов алгоритму легко подогнаться под обучающую выборку, т.е. переобучиться.

2. Да, повышается. Максимальная глубина дерева подразумевает, что дерево строится до тех пор, пока в каждом листе не останется по одному объекту, тогда алгоритм подстроится под обучающую выборку и будет плохо работать на новых данных.

3. Нет, не повышается. На самом деле здесь прямо противоположный ответ к пункту 2, т.к. на большой глубине появляются разбиения по не самым информативынм признакам, которые мы можем отсечь, например, с помощью ограничения на минимальное количество элементов в листе.

4. Да, повышается. Чем больше C, тем больше число правильно классифицированных объектов, т.е. алгоритм подгоняется под обучающую выборку.

5. Да, повышается, т.к. при больших значениях γ область влияния опорных векторов включает только опорный вектор.

6. Да, повышается. При увеличении степени, разделяющая полоса будет становится более сложной и проходить через большее число опорных векторов, что может привести к переобучению.

7. Нет, не повышается. Увеличение деревьев в бэггинге не изменяет смещение и уменьшает разброс, поэтому не возникает переобучение.

8. Нет, не повышается. При очень маленькой ширине ядра плотность стремится учесть все элементы, т.е. пытается подстроится под каждый элемент. При увеличении ширины плотность получается более сглаженной и без резких скачков.

9. Да, повышается. При увеличении степени полинома кривая будет проходить через большее количество точек обучающей выборки, таким образом, алгоритм будет переобучаться.

10. Нет, не повышается. При увеличении коэффициента происходит зануление ненужных признаков, что уменьшает переобучение.

Задача 2

1. Да, влияет. KNN - метрический алгоритм, поэтому нормировка данных может улучшить алгоритм в том случае, когда признаки имеют очень разные масштабы, т.к. один из признаков может доминировать над другим.

2. Нет, не влияет, т.к. получается эквивалентное решение, как для нормализованных данных, так и не нормализованных, однако нормализация позволяет быстрее сойтись к решению.

3. Нет, не влияет. Для построения решающего дерева используется правило, по которому разбиваются объекты, попавшие в данную вершину, а оно не зависит от нормализации данных.

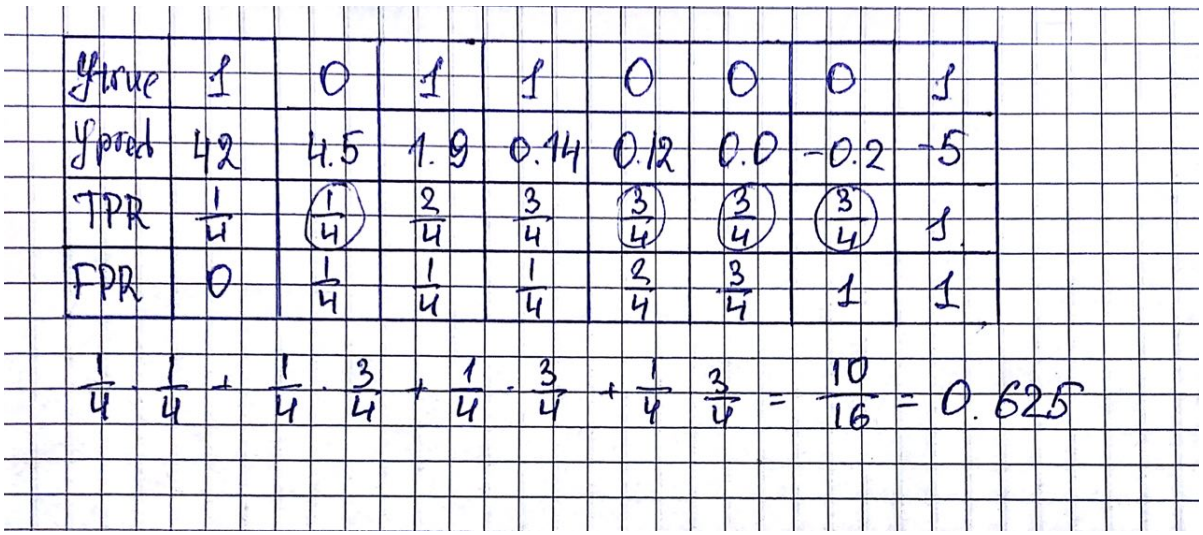
4. Нет, не влияет. Пояснение такое же, как в пункте 2.

5. Нет, не влияет. Случайный лес состоит из деревьев, которые мы считаем не зависимыми. Следовательно, нормализация не влияет.

6. Да, влияет. Это связано с тем, что масштаб признака влияет на то, будет ли регуляризация применяться к данному признаку.

7. Нет, не влияет. Пояснение такое же, как в пункте 2.

Задача 3

1. 

Ответ: roc-auc = 0.625

2. истинный класс = [1, 0, 1, 1, 1, 1, 0, 1]
предсказание = [-5, -0.2, 0., 0.12, 0.14, 1.9, 4.5, 42]

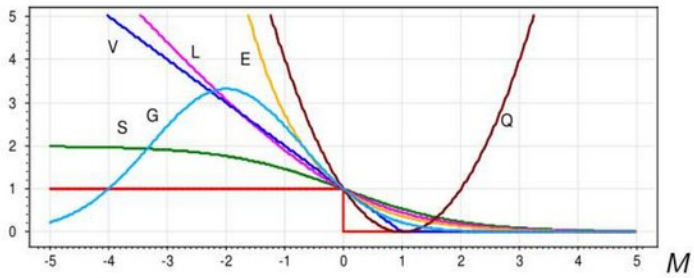
Задача 4

1. $y = \text{sign}\langle w, x\rangle$, где x - вектор параметров объекта, y - предсказанный класс (-1 или 1). Отступ $M = y\langle w, x\rangle$ положителен, если классификатор предсказал верный класс, иначе отступ отрицательный.

2. Функция потерь нужна для того, чтобы измерять, на сколько отличается предсказание модели от истинного значения. Обычно функция потерь больше штрафует те объекты, на которых алгоритм выдает не верный результат. Для классификации, например, можно использовать пороговую функцию потерь $[y_{pred} \neq y_{true}]$, а для регресии квадратичную $(y_{pred} - y_{true})^2$.

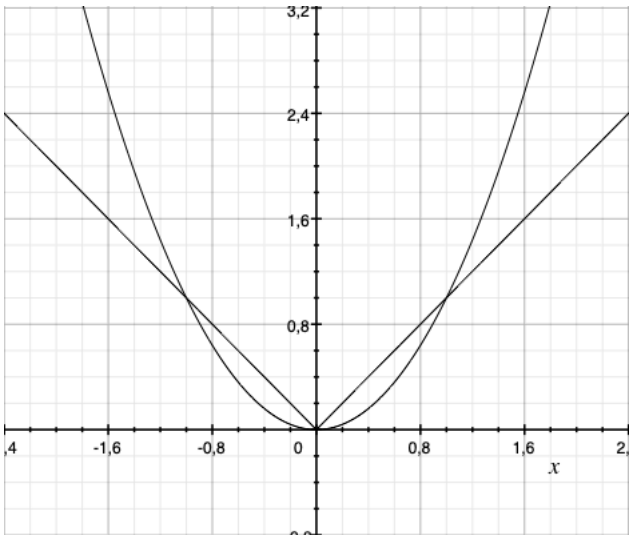
   Для классификации можно оценивать отступ, обычно график функции потерь такой, что при росте M, функция потерь убывает.

Функции потерь $\mathscr{L}(M)$ в задачах классификации на два класса



$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LogitBoost);
$G(M) = \exp(-cM(M + s))$ — гауссовская (BrownBoost);
$Q(M) = (1 - M)^2$ — квадратичная;
$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная;
$V(M) = (1 - M)_+$ — кусочно-линейная (SVM);

Функции потерь для регрессии (абсолютная и квадратичная)



В качестве негладкой и немонотонной функции потерь можно привести абсолютную функцию потерь $|y - y'|$.

Эмпирический риск - это функционал качества, характеризующий среднюю ошибку алгоритма на выборке X^n:

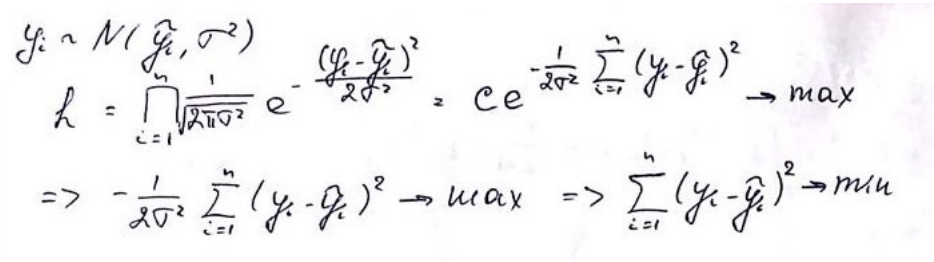
$$Q(X^n) \leq \frac{1}{n} \sum_{i=1}^{n} L(M(x_i))$$

3. На обучающей выборке алгоритм без регуляризации даст меньшее значение функции потерь,

так как он склонен к переобучению, потому что нет регуляризатора, который бы сдерживал веса.

Однозначно ответить про функцию потерь на тестовой выборке нельзя. С одной стороны, регуляризация помогает бороться с переобучением, поэтому потерь на тестовой выборке может быть меньше, но с другой стороны при больших значениях коэффициента регуляризации алгоритм использует меньше признаков, поэтому потерь больше.

4. Максимизация правдоподобия для распределения Лапласа приводит к минимизации суммы модулей отклонений.



Задача 5

$$LOOCV = \frac{1}{l}\sum_{i=1}^{l} Q(a(X^l \setminus x_i), x_i)$$

В алгоритме SVM можно разделить все объекты на: периферийные, опорные граничные, опорные нарушители. Заметим, что при обучении алгоритма на всей выборке без периферийного объекта, опорные вектора не изменяться, поэтому периферийный объект классифицируется верно. Совершенно инная ситуация, если обучается на всей выборке, кроме опорного объекта. В этом случае может быть ошибка на этом объекте. Всего $\#SV$ - число опорных векторов, но тогда из рассуждения справедлива оценка:

$$LOOCV \leq \frac{\#SV}{l}$$

Задача 6

$$\mu_0 = (0.5, 1.5)^T, \quad \mu_1 = (1.5, 0.5)^T, \quad \sigma^2 = 0.1$$

$$P(y=1|x) = \frac{P(y=1)P(x|y=1)}{P(y=0)P(x|y=0) + P(y=1)P(x|y=1)} = \frac{1}{1 + exp(\ln\frac{P(y=0)}{P(y=1)} + \ln\frac{P(x|y=0)}{P(x|y=1)})} =$$

$$= \frac{1}{1 + exp\left(\ln\frac{P(y=0)}{P(y=1)} + \sum_{i=1}^{2}\ln\frac{P(x_i|y=0)}{P(x_i|y=1)}\right)} = \frac{1}{1 + exp\left(\ln\frac{P(y=0)}{P(y=1)} + \sum_{i=1}^{2}\left(\frac{\mu_{0,i}-\mu_{1,i}}{\sigma_i^2}x_i + \frac{\mu_{1,i}^2-\mu_{0,i}^2}{2\sigma_i^2}\right)\right)}$$

Подставляя средние значения и дисперсию, получаем, что разделяющая прямая $x_1 = x_2$.
Задача 7

$$P(y=1|x) = \frac{P(y=1)P(x|y=1)}{P(y=0)P(x|y=0) + P(y=1)P(x|y=1)} = \frac{1}{1 + exp(\ln\frac{P(y=0)}{P(y=1)} + \ln\frac{P(x|y=0)}{P(x|y=1)})}$$

$$\ln\frac{P(y=0)}{P(y=1)}+\ln\frac{P(x|y=0)}{P(x|y=1)}=\ln\frac{P(x|y=0)}{P(x|y=1)}=\ln\frac{Ce^{-\frac{1}{2\sigma^2}(x-\mu_0)^T(x-\mu_0)}}{Ce^{-\frac{1}{2\sigma^2}(x-\mu_1)^T(x-\mu_1)}}=$$

$$=\frac{1}{2\sigma^2}\Big[-(x-\mu_0)^T(x-\mu_0)+(x-\mu_1)^T(x-\mu)\Big]$$

Учитывая, что количество объектов в каждом классе одинаковое, разделяющая поверхность вычисляется:

$$-(x-\mu_0)^T(x-\mu_0)+(x-\mu_1)^T(x-\mu)=0$$

$$x^Tx-2x^T\mu_1+\mu_1^T\mu_1-x^Tx+2x^T\mu_0-\mu_0^T\mu_0=0$$

$$2x^T(\mu_0-\mu_1)-\mu_0^T\mu_0+\mu_1^T\mu_1=0$$

$$4x_1-4x_2-40x_3+224=0$$

Задача 8

Предсказание класса случайно:

$$Ep_{\text{ошибки}}=E\frac{\sum\limits_{x_i\in R_m}[y_i\neq a(x_i)]}{N_m}=\frac{1}{N_m}\sum_{x_i\in R_m}E[y_i\neq a(x_i)]=\frac{1}{N_m}\sum_{x_i\in R_m}(1-p_{m,yi})=$$

$$=\frac{1}{N_m}\sum_{k=1}^{K}\sum_{x_i\in R_m}[y_i=k](1-p_{mk})=\sum_{k=1}^{K}p_{mk}(1-p_{mk})$$

Предсказание преобладающего в листе класса:

$$Ep_{\text{ошибки}}=E\frac{\sum\limits_{x_i\in R_m}[y_i\neq a(x_i)]}{N_m}=\frac{1}{N_m}\sum_{x_i\in R_m}E[y_i\neq a(x_i)]=\frac{1}{N_m}\sum_{x_i\in R_m}(1-\max_k p_{mk})=1-\max_k p_{mk}$$

Предсказание преобладающего в листе класса приводит к меньшей веротяности ошибки.

Задача 9

$$K(x,y)=\prod_{i=1}^{d}(1+x_iy_i)=1+x_1y_1+x_2y_2+...+x_dy_d+...+x_1y_1...x_dy_d$$

$$\varphi(x)=(1,x_1,\ldots,x_d,x_1x_2,\ldots,x_{d-1}x_d,\ldots,x_1x_2...x_d)$$

Задача 10