Школа анализа данных
Машинное обучение, часть 1
Домашнее задание №3

Решите предложенные задачи. Решения необходимо оформить в виде PDF документа. Каждая задача должна быть подробно обоснована, задачи без обоснования не засчитываются. Решения пишутся в свободной форме, однако так, чтобы проверяющие смогли разобраться. Если проверяющие не смогут разобраться в решении какой-нибудь задачи, то она автоматически не засчитывается.

Задача 1 (1 балл). Метрики качества.

1. (0.5 балла) Как известно, Precision-Recall (PR) кривая строится по точкам, соответствующим результатам классификации при различных значениях порога. Эта кривая соединяет две диагонально противоположные точки квадрата $[0, 1]^2$. Предложите алгоритм, который за $O(n \log n)$ вычисляет площадь под PR-кривой (иногда эту величину называют Average Precision (AP), хотя не все согласны с такой терминологией).

   Решение:

$$Precision = \frac{\sum\limits_{i=1}^{n}[a(x_i) = 1][y_i = 1]}{\sum\limits_{i=1}^{n}[a(x_i) = 1]}$$

$$Recall = \frac{\sum\limits_{i=1}^{n}[a(x_i) = 1][y_i = 1]}{\sum\limits_{i=1}^{n}[y_i = 1]}$$

   Алгоритм:

   $n_+ = \sum\limits_{i=1}^{n}[y_i = 1]$ $O(n)$

   Упорядочить выборку X^l по убыванию $f(x_i, w)$ $O(n\log n)$

   $(Precision_0, Recall_0) = (0, 0)$

   $count = 0$

   $AP = 0$

   for $i = 1, \ldots, n$: $O(n)$

   if $y_i = 1$:

   $count = count + 1$

   $Precision_i = \frac{count}{i}$

   $Recall_i = Recall_{i-1} + \frac{1}{n_+}$

$AP = AP + \frac{1}{n_+} \frac{Precision_i + Precision_{i-1}}{2}$

else:

$Precision_i = \frac{count}{i}$

$Recall_i = Recall_{i-1}$

Асимптотика алгоритма: $O(n) + O(n\log(n)) + O(n) = O(n\log(n))$

2. (0.5 балла) Студент Александр решает конкурс на kaggle. Решается задача бинарной классификации, участники ранжируются по метрике F1. Александр заметил, что его положение на лидерборде сильно зависит от выбора порога, по которым происходит отсечение на этапе проставления меток. Он может перебрать всевозможные пороги (из тех, которые есть смысл рассматривать) и выбрать тот, который максимизирует значение F1 на его отложенной выборке. Однако, данных очень много, а по оценкам Александра этот алгоритм требует порядка $O(n^2)$ операций. Предложите алгоритм, который работает быстрее, но при этом находит оптимальный порог.

Решение:

Чтобы найти оптимальный порог, можно немного изменить алгоритм из пункта 1 (асимптотика остается такой же): на каждом шаге алгоритма будем подсчитывать не AP, а максимальное F1-меры и оптимальный порог, при котором максимум F1-меры достигается. Для этого рассмотрим отсортированные значения алгоритма по убыванию: $f_1 \geq f_2 \cdots \geq f_n$. Будем искать такой порог $t = \frac{f_i + f_{i+1}}{2}$, который бы максимизировал $F1 = \frac{2 * precission * recall}{precision + recall}$

Алгоритм:

$n_+ = \sum_{i=1}^{n} [y_i = 1]$ $O(n)$

Упорядочить выборку X^l по убыванию $f(x_i, w)$ $O(n\log n)$

$(Precision_0, Recall_0) = (0, 0)$

$count = 0$

$F1 = 0$

$t = null$

for $i = 1, \ldots, n$: $O(n)$

if $y_i = 1$:

$count = count + 1$

$Precision_i = \frac{count}{i}$

$Recall_i = Recall_{i-1} + \frac{1}{n_+}$

else:

$Precision_i = \frac{count}{i}$

$Recall_i = Recall_{i-1}$

$F1_{cur} = \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i}$

if $F1 > F1_{cur}$: $F1 = F1_{cur}$, $t = \frac{f_i + f_{i+1}}{2}$

Задача 2 (1.5 балла). Метрики качества. Неудачный выбор метрики.

Костя участвует в конкурсе по анализу данных, в котором нужно решить задачу бинарной классификации, в которой для оценивания качества используется функционал ошибки Mean Absolute Error (MAE):

$$Q(\vec{y}, \tilde{\vec{y}}) = \frac{1}{l}\sum_{i=1}^{l} |y_i - \tilde{y}_i|, \quad \vec{y} \in \{0,1\}^l, \quad \tilde{\vec{y}} \in [0,1]^l,$$

где l — количество обучающих объектов, $\vec{y}$ — вектор истинных классов объектов, $\tilde{\vec{y}}$ — вектор предсказанных «степеней принадлежности» классу 1, качество которого и оценивается. Костя заметил, что качество предсказания на скрытой выборке, которая доступна только организаторам конкурса, часто улучшается, если сдвинуть прогноз $\tilde{y}_i$ в один из концов отрезка $[0,1]$ для каждого объекта i. Объясните, почему Костины действия привели к улучшению качества предсказания. При каких обстоятельствах Костя мог провернуть такой трюк? Всегда ли такое возможно?

Подсказка: Сделайте предположение, что объект с номером i принадлежит классу 1 с истинной вероятностью p_i.

***** Оцениваться будут рассуждения, каким условиям должны удовлетворять предсказания, чтобы идея Кости сработала, а также подтверждающие и/или опровергающие примеры.

Решение: Найдем матожидание функционала ошибки:

$$E(Q(y,\hat{y})|x) = E(|y_i - \hat{y_i}|\big|x_i) = |1-\hat{y_i}|p(y_i=1|x_i) + |0-\hat{y_i}|(1-p(y_i=1|x_i)) = (1-\hat{y_i})p(y_i=1|x_i) + \hat{y_i}(1-p(y_i=1|x_i)),$$

где $p(y_i = 1|x_i)$-истинная вероятность того, что объект с номером i принадлежит классу 1

$$\frac{\partial E(Q(y,\hat{y})|x)}{\partial \hat{y_i}} = 1 - 2p(y_i=1|x_i) = 0 => p(y_i=1|x_i) = \frac{1}{2}$$

Если $p(y_i = 1|x_i) = \frac{1}{2}$, то $E(Q(y,\hat{y})) = \frac{1}{2}$. Получаем, что для любого предсказания вероятности матожидание функционала ошибки равна $\frac{1}{2}$, поэтому будут предсказаны не верные вероятности.

Если истинная вероятность $p(y_i = 1|x_i) \neq \frac{1}{2}$, тогда минимум матожидания будет достигаться на концах отрезка $[0,1]$, т.е. $minE(Q(y,\hat{y})\big|x) = min(p(y_i|x_i), 1 - p(y_i|x_i))$.

Можно сделать следующие выводы:

1. данный функционал ошибки не позволяет предсказывать верные вероятности принадлежности к классу 1
2. Костя смог улучшить свои предсказания, сдвинув их в один из концов отрезка $[0,1]$, т.к. истинные вероятности не равны $\frac{1}{2}$ и его модель не предсказывает верные вероятности в силу неверно выбранного функционала ошибки

Приведем примеры:

1. $y_{true} = [1,1,1,0,0], y_{prob} = [0.4, 0.5, 0.4, 0.3, 0.1]$ (предсказанные вероятности)

   MAE в данном случае будет равен 0.42. Теперь попробуем сдвинуть все предсказания в 1, тогда MAE равен 0.4. => оценка улучшилась
2. Аналогично можно построить пример, когда все предсказания сдвигаются в 0 и MAE уменьшается.
3. $y_{true} = [1,1,1,0,0,0], y_{prob} = [0.7, 0.5, 0.6, 0.4, 0.3, 0.6]$ (предсказанные вероятности)

   MAE в данном примере увеличивается с 0.417 до 0.5, если сдвинуть предсказания в 0 или 1.
4. $y_{true} = [1,1,0,0,0], y_{prob} = [0.4, 0.5, 0.4, 0.3, 0.1]$

   MAE = 0.38, MAE для всех предсказаний, равных 1, равен 0.6, а для всех предсказаний, равных 0, равен 0.4

Из примеров можно сделать вывод, что у Кости могла быть тестовая выборка, в которой объектов одного класса больше, поэтому MAE **может** уменьшится (например, в 4 примере данная тактика не работает, а в 1 работает). Также следует отметить, что для правильных предсказаний вероятностей не нужно использовать данный функционал ошибки.

Задача 3 (1 балл). ROC AUC.

Определим понятие доли дефектных пар ответов классификатора. Пусть дан классификатор $a(x)$, который возвращает оценки принадлежности объектов классам: чем больше ответ классификатора, тем более он уверен в том, что данный объект относится к классу «+1». Отсортируем все объекты по неубыванию ответа классификатора a: $x_{(1)}, \ldots, x_{(\ell)}$. Обозначим истинные ответы на этих объектах через $y_{(1)}, \ldots, y_{(\ell)}$. Тогда доля дефектных пар записывается как

$$\mathrm{DP}(a, X^\ell) = \frac{2}{\ell(\ell-1)} \sum_{i<j}^{\ell} \left[y_{(i)} > y_{(j)}\right].$$

Как данный функционал связан с AUC (площадью под ROC-кривой)? Ответ должен быть дан в виде формулы, связывающей DP и AUC.

Решение:

Если рассмотреть алгоритм вычисления AUC, можно заметить, что ROC-кривая состоит из прямоугольников с шириной $\frac{1}{l_-}$ и высотой $\frac{1}{l_+} \sum\limits_{j=i+1}^{l} [y_{(j)} = 1]$.

$$AUC(a, X^l) = \frac{1}{l_-} \sum_{i=1}^{l} [y_{(i)} = -1] \frac{1}{l_+} \sum_{j=i+1}^{l} [y_{(j)} = 1] = \frac{1}{l_- l_+} \sum_{i=1}^{l} [y_{(i)} = -1] \sum_{j=i+1}^{l} [y_{(j)} = 1] = \frac{1}{l_- l_+} \sum_{i<j}^{l} [y_{(i)} < y_{(j)}] =$$

$$= \frac{1}{l_- l_+} \sum_{i<j}^{l} \Big(1 - [y_{(i)} > y_{(j)}] - [y_{(i)} = y_{(j)}]\Big) = \frac{1}{l_- l_+} \sum_{i<j}^{l} \Big(1 - [y_{(i)} = y_{(j)}]\Big) - \frac{1}{l_- l_+} \sum_{i<j}^{l} [y_{(i)} > y_{(j)}] =$$

$$= \frac{1}{l_- l_+} \sum_{i<j}^{l} [y_{(i)} \neq y_{(j)}] - \frac{1}{l_- l_+} \sum_{i<j}^{l} [y_{(i)} > y_{(j)}] = \frac{l_- l_+}{l_- l_+} - \frac{1}{l_- l_+} \sum_{i<j}^{l} [y_{(i)} > y_{(j)}] = 1 - \frac{l(l-1)}{2 l_- l_+} DP(a, X^l)$$

$$DP(a, X^l) = \frac{2 l_- l_+ (1 - AUC(a, X^l))}{l(l-1)}$$

Задача 4 (1.5 балла). Решающие деревья, индекс Джини.

Пусть имеется построенное решающее дерево для задачи многоклассовой классификации. Рассмотрим лист дерева с номером m и объекты R_m, попавшие в него. Обозначим за p_{mk} долю объектов k-го класса в листе m. *Индексом Джини* этого листа называется величина

$$\sum_{k=1}^{K} p_{mk}(1 - p_{mk}),$$

где K — общее количество классов. Индекс Джини обычно служит мерой того, насколько хорошо в данном листе выделен какой-то один класс.

1. (0.5 балла) Поставим в соответствие листу m алгоритм классификации $a(x)$, который предсказывает класс случайно, причем класс k выбирается с вероятностью p_{mk}. Покажите, что матожидание частоты ошибок этого алгоритма на объектах из R_m равно индексу Джини.

Решение:

Пусть ν - частота ошибок этого алгоритма на объектах из R_m. Найдем её матожидание:

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

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

2. (0.5 балла) Докажите, что стратегия предсказания преобладающего класса в листе приводит к меньшей вероятности ошибки, чем алгоритм из предыдущего пункта

Решение:

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

Действительно, стратегия предсказания преобладающего класса в листе приводит к меньшей вероятности ошибки, чем предсказание случайного класса.

3. (0.5 балла) *Дисперсией класса k* назовем дисперсию выборки $\{[y_i = k] : \ x_i \in R_m\}$, где y_i — класс объекта x_i, $[f]$ — индикатор истинности выражения f, равный 1 если f верно, и нулю в противном случае, а R_m — множество объектов в листе. Покажите, что сумма дисперсий всех классов в заданном листе равна его индексу Джини.

Решение:

Рассмотри случайную величину $\varepsilon = \begin{cases} 1, \text{класс k} \\ 0, \text{иначе} \end{cases}$

Матожидание этой случайной величины равняется $E\varepsilon = 1 * p_{mk} + 0 * (1 - p_{mk}) = p_{mk}$, а дисперсия $D\varepsilon = (1 - p_{mk})^2 p_{mk} + (0 - p_{mk})^2(1 - p_{mk}) = p_{mk}(1 - p_{mk})$

Следовательно, сумма дисперсий для всех классов будет равняться $\sum\limits_{k=1}^{K} p_{mk}(1 - p_{mk})$.

Задача 5 (2 балла). Рекомендательные системы, матричные разложения.

Допустим, мы хотим обучить рекомендательную систему, имея историю взаимодействия n пользователей и m товаров. Рассмотрим матрицу оценок $R \in \mathbb{R}^{n \times m}$, в которой известны лишь некоторые элементы: r_{ui} – оценка пользователя $1 \leq u \leq n$ для товара $1 \leq i \leq m$. Дополнительно предположим, что доля пар (u, i) с известными оценками относительно всех пар равна α, причем эти пары равномерно распределены по матрице оценок. Рассмотрим способ предсказания неизвестных оценок на основе модели Latent Factor Model, которая использует малоранговую аппроксимацию ранга r матрицы R:

$$\tilde{r}_{ui} = p_u^T q_i, \quad p_u \in \mathbb{R}^r, \quad q_i \in \mathbb{R}^r.$$

Параметры модели можно подбирать в ходе решения задачи восстановления пропущенных значений матрицы оценок (Matrix Completion):

$$P, Q = \underset{P \in \mathbb{R}^{r \times n}, Q \in \mathbb{R}^{r \times m}}{\arg\min} \sum_{(u,i) \in \text{known}} (r_{ui} - p_u^T q_i)^2.$$

Рассмотрим один из алгоритмов решения этой оптимизационной задачи – Alternating Least Squares (ALS). Этот алгоритм поочередно фиксирует матрицы P или Q и подбирает оптимальное значение другой матрицы, точно решая задачу с помощью линейного метода наименьших квадратов. Например, при фиксированном P:

$$Q = \arg\min_{Q \in \mathbb{R}^{r \times m}} \sum_{(u,i) \in \text{known}} (r_{ui} - p_u^T q_i)^2.$$

Выпишите явные формулы для вычисления оптимального Q при фиксированном P, а также покажите, что вычислительная сложность одного такого шага равна $O\left(r^2 m(\alpha n + r)\right)$.

Решение:

Пусть $L = \sum_{(u,i) \in \text{known}} (r_{ui} - p_u^T q_i)^2$. Тогда:

$$\frac{\partial L}{\partial q_i} = -\sum_{u \in \text{known}} 2(r_{ui} - p_u^T q_i) p_u = 0 \Rightarrow q_i = \left(\sum_{u \in \text{known}} p_u p_u^T \right)^{-1} \left(\sum_{u \in \text{known}} r_{ui} p_u \right)$$

Тогда $Q = (PP^T)^{-1} PR$. Теперь найдем асимптотику. q_i - столбец матрицы Q, тогда асимптотика нахождения q_i $O(r^3 + \alpha n r^2)$. Всего столбцов в матрице m, следовательно матрицу Q можно найти за $O(m(r^3 + \alpha n r^2)) = O(r^2 m(\alpha n + r))$.

Задача 6 (1 балл). Рекомендательные системы, матричные разложения.

В факторизационных машинах предсказание для объекта $x \in \mathbb{R}^d$ делается по формуле

$$a(x) = w_0 + \sum_{i=1}^{d} w_j x_j + \sum_{i=1}^{d} \sum_{j=i+1}^{d} x_i x_j \langle \vec{v}_i, \vec{v}_j \rangle.$$

Вычисление предсказания по этой формуле требует $O(rd^2)$ операций, где r — размер векторов $\vec{v}_1, \ldots, \vec{v}_d$. Покажите, что это же предсказание может быть найдено за $O(rd)$ операций.

Решение:

Распишем последнее слагаемое (с двумя суммами) и покажем, что его можно вычислить за $O(rd)$, тогда предсказание для объекта можно вычислить за требуемую асимптотику.

$$\sum_{i=1}^{d} \sum_{j=i+1}^{d} x_i x_j \langle \vec{v}_i, \vec{v}_j \rangle. = \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} x_i x_j \langle \vec{v}_i, \vec{v}_j \rangle - \frac{1}{2} \sum_{i=1}^{d} x_i x_i \langle \vec{v}_i, \vec{v}_i \rangle = \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{r} x_i x_j v_{ik} v_{jk} - \frac{1}{2} \sum_{i=1}^{d} \sum_{k=1}^{r} x_i^2 v_{ik}^2 =$$

$$= \frac{1}{2} \sum_{k=1}^{r} \left(\left(\sum_{i=1}^{d} x_i v_{ik} \right) \left(\sum_{j=1}^{d} v_{jk} x_j \right) - \sum_{i=1}^{d} x_i^2 v_{ik}^2 \right) = \frac{1}{2} \sum_{k=1}^{r} \left(\left(\sum_{i=1}^{d} x_i v_{ik} \right)^2 - \sum_{i=1}^{d} x_i^2 v_{ik}^2 \right)$$

Получаем асимптотику для вычисления преобразованного выражения: $O(2rd) = O(rd)$.

Задача 7 (1 балл). Матричные разложения.

Дана выборка $X^n = \{(x_1, y_1), \ldots, (x_n, y_n)\}, x_i, y_i \in \mathbb{R}, i = 1, \ldots, n$. В задаче линейной регрессии коэффициенты a, b, c уравнения прямой $ax + by + c = 0$ находятся через минимизацию среднеквадратичной ошибки по y.

В этой же задаче предлагается подобрать коэффициенты так, чтобы сумма квадратов расстояний от точек выборки до прямой была минимальна. Выведите формулы для вычисления a, b, c.

Задача 8 (1 балл). Рекомендательные системы.

Андрей и Денис решили воспользоваться машинным обучением, чтобы улучшить продажи в своем интернет-магазине. Для этого они разработали рекомендательную систему, предлагающую пользователю три товара в момент, когда он сформировал корзину и собрался делать заказ. Эта

система делает рекомендации как на основе товаров в корзине, так и на основе истории пользователя. Например, если пользователь хочет купить фотоаппарат, то система предложит ему сумку и штатив, поскольку их часто покупают в дополнение к фотоаппаратам. Или же, если данный пользователь часто покупает книги, то система может предложить ему купить вместе с фотоаппаратом недавно вышедший детектив.

Рекомендательная система у Андрея и Дениса получилась очень сложная и ресурсоемкая, и для ее стабильной работы необходимо закупить несколько мощных серверов, а также платить зарплату системному администратору, который будет следить за работой этих серверов и самой рекомсистемы. Чтобы проверить, имеет ли смысл идти на такие траты, они хотят понять, способен ли разработанный алгоритм делать правильные рекомендации.

Денис предлагает взять историю покупок пользователей, и для каждого пользователя разбить ее на две части: первые 70% взять в обучающую выборку, а последние 30% — в контроль. После этого следует настроить рекомендательный алгоритм по обучающей выборке и проверить, насколько хорошо он предсказывает пользователям покупки из контрольной выборки.

Андрей же утверждает, что нужно провести AB-тест: разбить всех пользователей интернет-магазина на две группы и показывать рекомендации лишь в одной из групп. После этого предлагается подсчитать число купленных товаров в первой и второй группах. Если в группе, где показывались рекомендации, это число окажется больше, то рекомендательную систему следует признать полезной.

Какие преимущества и недостатки вы видите в подходах Андрея и Дениса? Кому из них следует отдать предпочтение?

Решение:

Подход Дениса довольно прост, эта идея может применятся при тестировании любых алгоритмов машинного обучения, но какие минусы есть при тестировании конкретно рекомендательной системы? Самым очевидным ограничением является то, что оценить точность прогноза мы можем только на тех товарах, которые пользователь уже оценил (положил в корзину или смотрит). Таким образом, если рекомсистема не видела товар, то и предложить рекомендацию для него хорошо мы не сможем. Из плюсов отметим, что такое тестирование проводится быстрее, чем AB.

Думаю, что подход Андрея лучше, чем подход Дениса. При AB-тестировании мы можем посмотреть не только на то, купил ли пользователь товар из рекомендации, но и посмотрел ли его вообще (правда тут тоже есть свой минус, возможно, что конкретный пользователь просто не смотрит на рекомендации, поэтому надо смотреть на поведение всех пользователей в совокупности). Из минусов стоит отметить, что AB-тестирование следует делать какое-то длительное время, чтобы получить большую выборку и сделать статистически значимые выводы.

Стоит отметить, что на практике можно использовать оба варианта.