Машинное обучение (СБТ): задание 1

Составитель: Виктор Кантор

31 марта 2017 г.

Организационные вопросы

Дедлайн

На это задание дается две недели до 14 апреля.

Сдача задания

1. Заведите свой репозиторий на github для домашних заданий по курсу
2. Выполненные задания сохраните в ipynb (в случае практических задач) и pdf (в случае теоретических) и сделайте push в свой репозиторий. Желательно для каждого домашнего задания в дальнейшем заводить отдельную папку.
3. Пришлите на почту xead.wl@gmail.com ссылку на выложенное на github решение задания. На всякий случай продублируйте в письме ipynb и pdf файлы. Тему письма укажите в формате «SBT_ML2017 <фамилия> <имя>, Задание 1 (Introduction and trees)», например: «SBT_ML2017 Страуструп Бьярн, Задание 1 (Introduction and trees)»
4. Если есть принципиальное желание оформлять теоретические задачи в ipynb в Markdown-ячейках, это не запрещается. Также не запрещается решать теоретические задачи на бумаге, оформлять их аккуратно и разборчиво, и присылать pdf со сканами, вместо того, чтобы набирать в LaTeX.

Контрольные вопросы

Ниже приводится список вопросов, с ответами на которые может быть полезно разобраться для комфортного выполнения задания.

Основные понятия

1. Что такое задачи классификации, кластеризации и регрессии? Какие из них относятся к supervised learning, а какие - к unsupervised?
2. Что такое переобучение и недообучение? Как их можно детектировать?
3. Что такое обучающая и тестовая выборки, кросс-валидация? Как устроена k-fold cross validation?

Простые методы

1. Как работает kNN в задаче классификации?
2. Как работает kNN с весами объектов в задаче классификации и в задаче регрессии?
3. Как работает наивный байесовский классификатор, в чем заключается его «наивность»?
4. Как приближается исходная зависимость y от x в линейной регрессии и как настраиваются веса в ней?

Python, numpy, scipy, matplotlib

1. Типы данных list, tuple, dict, set, str, unicode, hashable и unhashable типы. Управляющие конструкции в python (циклы, условные операторы), объявление функций. Map и reduce, list comprehensions, генераторы, лямбда-выражения. Чтение и запись в файл в Python.
2. Зачем нужны numpy и scipy? Какой тип данных в numpy используется для работы с многомерными массивами? Отличия в индексации двумерного ndarray и списка списков.
3. Как в scipy решить численно несложную оптимизационную задачу? Какие методы оптимизации в нем представлены?
4. Как по списку значений x и списку значений y в этих точках построить график y(x) в matplotlib?

Метрики качества в задачах классификации и регрессии

1. Как вычисляются и в каких задачах (классификации/регрессии) применяются метрики: accuracy, precision, recall, F1-measure, ROC-AUC, log loss, MSE, MAE, RMSE?
2. Решается задача бинарной классификации (с двумя классами — 0 и 1), в которой примеры из класса 0 составляют 95% выборки. Какие метрики из перечисленных в предыдущем вопросе предпочтительней использовать?
3. К оценке какой величины для распределения y при условии x приводят MSE и MAE?
4. Можно ли при таргетах из множества $Y = \{0; 1\}$ использовать для оценки $P(y = 1|x)$ не log loss, а MSE?

Scikit-learn и pandas

1. Как в sklearn обучить модель на обучающей выборке и получить прогнозы на тестовой?
2. Какие есть средства для измерения качества модели в sklearn? Как посчтитать качество в k-fold cross validation?
3. Какие метрики можно использовать в cross_val_score из sklearn?
4. Как считать выборку из csv в pandas DataFrame? А как записать DataFrame в файл? Как указывать при чтении/записи кодировку, используемые разделители, наличие/отсутствие заголовков у колонок?
5. Как по списку значений x и списку значений y в этих точках построить график y(x) в matplotlib?

Деревья

1. Как выглядит решающее дерево? Как применяется уже построенное для задачи классификации дерево? А для задачи регрессии?
2. Как строятся решающие деревья? (рекомендуется обратиться к материалам лекций или документации sklearn)
3. Как выглядят энтропийный критерий, критерий Джини и среднеквадратичное отклонение, используемое как критерий в задаче регрессии?
4. Что такое node impurity и goodness of split? Как они связаны?
5. Какие преимущества и недостатки есть у деревьев? (полезно как подумать самостоятельно, так и обратиться к документации sklearn)
6. Есть ли разница (с точки зрения вида получаемого в итоге дерева): строить каждое разбиение в дереве, максимизируя информативность, или строить каждое разбиение, минимизируя «ошибку», как было предложено на первой лекции про деревья?

1 Метод k ближайших соседей

10% баллов за задание, оценочное время выполнения: 20 минут

Сгенерируйте обучающую выборку из описанных двумя признаками объектов нескольких классов и визуализируйте разделяющие поверхности, получаемые при решении задачи классификации методом k ближайших соседей для разных k. Попробуйте подобрать оптимальное значение количества соседей k с помощью 5-fold cross-validation, построив график зависимости accuracy в кросс-валидации от k.

2 Наивный байесовский классификатор

20% баллов за задание, оценочное время выполнения: 40 минут

Загрузите датасеты digits и breast_cancer из sklearn.datasets. Выведите несколько строчек из обучающих выборок и посмотрите на признаки. С помощью sklearn.model_selection.cross_val_score с настройками по умолчанию и вызова метода mean() у возвращаемого этой функцией numpy.ndarray, сравните качество работы наивных байесовских классификаторов на этих двух датасетах. Для сравнения предлагается использовать BernoulliNB, MultinomialNB и GaussianNB. Насколько полученные результаты согласуются с вашими ожиданиями?

Два датасета, конечно, еще не повод делать далеко идущие выводы, но при желании вы можете продолжить исследование на других выборках (например, из UCI репозитория).

Ответьте (прямо в ipynb блокноте с вашими экспериментами) на вопросы:

1. Каким получилось максимальное качество классификации на датасете breast_cancer?
2. Каким получилось максимальное качество классификации на датасете digits?
3. Какие утверждения из приведенных ниже верны?
   (a) На вещественных признаках лучше всего сработал наивный байесовский классификатор с распределением Бернулли
   (b) На вещественных признаках лучше всего сработал наивный байесовский классификатор с мультиномиальным распределением
   (c) Мультиномиальное распределение лучше показало себя на выборке с целыми неотрицательными значениями признаков
   (d) На вещественных признаках лучше всего сработало нормальное распределение

3 Метрики в задаче регрессии

40% баллов за задание, оценочное время выполнения: 120 минут

Сгенерируйте датасет из 500 точек на плоскости, для которых $y = 0.5x + 1 + \varepsilon$, где ε распределено нормально с матожиданием 0 и дисперсией 0.2.

1. Визуализируйте выборку.
2. Восстановите по выборке зависимость $y(x)$, считая, что зависимость имеет вид $y = kx+b$, и минимизируя MSE на обучающей выборке, воспользовавшись scipy.optimize.minimize. Визуализируйте восстановленную прямую.
3. Добавьте теперь в выборку 75 точек, для которых $y = -1+\varepsilon$, а x принимает различные значения из того же диапазона, что и у уже имевшихся точек в обучающей выборке. По новой расширенной выборке снова попробуйте восстановить зависимость $y(x) = kx + b$ двумя способами: минимизируя MSE и минимизируя MAE. Визуализируйте полученные прямые.
4. На основе полученных графиков сделайте вывод об устойчивости моделей, оптимизирующих MSE и MAE к выбросам.

4 Применение решающего дерева

20% баллов за задание, оценочное время выполнения 30 минут + установка GraphViz

Постройте решающее дерево из sklearn на датасете german credit data из UCI репозитория и визуализируйте его. Попробуйте проинтерпретировать первые несколько разбиений, изучив описание признаков. Постройте графики зависимости качества на кросс-валидации и на обучающей выборке от глубины дерева

5 Реализация решающего дерева (опциональная часть)

50% баллов за задание, оценочное время выполнения 3-4 часа

В этом задании предлагается использовать датасет boston из sklearn.datasets. Оставьте последние 25% объектов для контроля качества, разделив X и y на X_train, y_train и X_test, y_test.

Реализуйте свой класс DecisionTree, имеющий методы fit и predict, позволяющие соответственно обучить решающее дерево по матрице признаков X_train и ответам y_train, а затем спрогнозировать ответы на тестовой выборке X_test. Оцените качество работы вашего дерева на тестовой выборке.

Рекомендации по реализации дерева:

1. Обучение дерева можно реализовать простым жадным рекурсивным алгоритмом — каждый раз выбирайте наилучшее разбиение (номер признака и порог по нему) по уместному на ваш взгляд критерию из рассмотренных на первой лекции о решающих деревьях (MSE, gini, энтропийный критерий, ошибка классификации)

2. Выбор наилучшего разбиения можно сделать простым перебором по признакам и порогам.

3. Пороги можно перебирать из заранее заданного множества порогов на обучающей выборке - например, взяв все пороги между принимаемыми значениями координат, либо взяв случайный набор порогов, либо взяв пороги по квантилям значений каждого признака (посчитать квантили будет несложно с помощью scipy). Если возможных порогов будет слишком много, выбор наилучшего разбиения может оказаться слишком долгой операцией.

4. Сделайте возможным передавать в конструктор класса ограничение по глубине дерева и заканчивайте построение дерева при достижении этого ограничения.

5. Можно реализовать отдельный класс для решающего правила вида «k-ый признак меньше порога» и отдельный класс для дерева. Также вам предстоит подумать, как хранить разбиения внутри дерева, чтобы их было удобно использовать.

6. Какие-то из решений вы можете подсмотреть в чужих реализациях дерева, но от вас не требуется написать применимую на практике библиотеку - только максимально простую демонстрацию того, как строится и применяется решающее дерево.

Примечание: на случай, если эта задача покажется большой, — среди студентов ФИВТ МФТИ были примеры ее выполнения за одну пару с написанием в процессе хорошего, продуманного, понятного и задокументированного кода.

6 Теоретические задачи

30% баллов за задание

6.1 Наивный байес и центроидный классификатор

Покажите, что если в наивном байесовском классификаторе классы имеют одинаковые априорные вероятности, а плотность распределения признаков в каждом классе имеет вид $P(x^{(k)}|y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^{(k)}-\mu_{yk})^2}{2\sigma^2}}$, $x^{(k)}, k = 1, \cdots, n$ — признаки объекта x классификация сводится к отнесению объекта x к классу y, центр которого μ_y ближе всего к x.

6.2 ROC-AUC случайных ответов

Покажите, что «треугольный ROC-AUC» (см.лекцию 2) в случае, когда классификатор дает случайные ответы — $a(x) = 1$ с вероятностью p и $a(x) = 0$ с вероятностью $1 - p$, будет в среднем равен 0.5, независимо от p и доли класса 1 в обучающей выборке.

6.3 Ошибка 1NN и оптимального байесовского классификатора

Утверждается, что метод одного ближайшего соседа асимптотически (при условии, что максимальное по всем точкам выборки расстояние до ближайшего соседа стремится к нулю) имеет матожидание ошибки не более чем вдвое больше по сравнению с оптимальным байесовским классификатором (который это матожидание минимизирует).

Покажите это, рассмотрев задачу бинарной классификации. Достаточно рассмотреть вероятность ошибки на фиксированном объекте x, т.к. матожидание ошибок на выборке размера V будет просто произведением V на эту вероятность. Байесовский классификатор ошибается на объекте x с вероятностью:

$$E_B = \min\{P(1|x), P(0|x)\}$$

Условные вероятности будем считать непрерывными функциями от $x \in R^m$, чтобы иметь возможность делать предельные переходы. Метод ближайшего соседа ошибается с вероятностью:

$$E_N = P(y \neq y_n)$$

Здесь y - настоящий класс x, а y_n - класс ближайшего соседа x_n к объекту x в предположении, что в обучающей выборке n объектов, равномерно заполняющих пространство.

Докажите исходное утверждение, выписав выражение для E_N (принадлежность к классам 0 и 1 для объектов x и x_n считать независимыми событиями) и осуществив предельный переход по n.

6.4 Ответы в листьях регрессионного дерева

Какая стратегия поведения в листьях регрессионного дерева приводит к меньшему матожиданию ошибки по MSE: отвечать средним значением таргета на объектах обучающей выборки, попавших в лист, или отвечать таргетом для случайного объекта из листа (считая все объекты равновероятными)?

6.5 Линейные модели в деревьях

Одна из частых идей — попытаться улучшить регрессионное дерево, выдавая вместо константных ответов в листьях ответ линейной регрессии, обученной на объектах из этого листа. Как правило такая стратегия не дает никакого ощутимого выигрыша. Попробуйте объяснить, почему? Как стоит модифицировать построение разбиений в дереве по MSE, чтобы при разбиении получались множества, на которых линейные модели должны работать неплохо?