МО лекция 1

Очень вводная - основные понятия

МО = обширный подраздел ИИ, изучающий методы построения алгоритмов, способных обучаться

Виды данных: таблицы, текст, изображения, звук, логи
Виды переменных: категориальные, бинарные, числовые, признаки со сложной структурой (изображения)
Виды задач: с и без учителя

Примеры: чтение по губам, игра в шахматы, в го, ответы на вопросы по тексту

Матрица объект-признак
Признаки, факторы описывают объекты

Типы задач: классификация, регрессия, ранжирование, кластеризация, снижение размерности (+ визуализация), оценивание плотности (приближение распределения объектов)

Функционал ошибки (функция потерь) - то, что мы минимизируем в задаче
Метрика качества - нужны для сравнения моделей и для оценки качества модели

МО лекция 2

Векторное и матричное дифференцирование

Closed form OLS: $w = X^TX^{-1}X^Ty$
Closed form L1 regularized OLS

GD
- Формула градиентного спуска
- Теорема о градиенте
- Понятие градиентного шага (learning rate)
- Варианты инициации весов
- Критерии останова
- Stochastic GD

- Batch and mini-catch GD
- ADAGRAD
- RMSPROP

Понятие переобучения

Метод максимального праавдоподобия
Метод моментов

МО лекция 3

Если не убирать выбросы - лучше использовать MAE, если выбросили, то можно MSE
Доказательство дифференцирования MAE - 50:00, MSE - 44:00

MAPE, SMAPE, etc.

Переобучение:
- Test error сильно больше, чем на train
- Очень большие веса

Регуляризация
L1 - зануляет веса + есть closed form solution (проверить)

МО лекция 4

CV:
- **K-fold**
- **LOO** (по сути если в K-fold folds = количество объектов, тогда это LOO)
- **Complete**
- Leave-k-out
- Expanding window
- Moving window
- Stratified

K-fold чаще всего используют, на практике от 3 до 5 folds чаще всего оптимально

Параметры - величины, которые настраиваются по обучающей выборке
Гиперпараметры модели - величины, контролирующие процесс обучения

(мы сами выбираем, например, количество folds, градиентный шаг)

Гиперпараметры мы находим кросс-валидацией:

1. Train, Test split
2. K-fold по train sample
3. Find optimal hyperparameters
4. Fit the model with optimal hyperparameters with the whole train sample
5. Predict on X_test

Кодирование категориальных переменных:

- One-hot-encoding (dummy variables) - несколько столбцов из нулей и единиц
  Проблемы: много столбцов, то есть раздуваем размер данных, это может привести к переобучению + в тесте может появится новая категория, которой не было в тренировочной выборке
- Счётчик - заменяем категорию на вероятность получить определенный таргет для этой категории. Если несколько типов таргетов, тогда делаем несколько столбцов с вероятностями такого таргета для каждой категории
  Проблемы: переобучение очень вероятно
  - Сглаживание - редкие категории заполняем глобальным средним
  - Отложенная выборка - считать счетчики только на части данных, особенно переобучение заметно на мелких категориях
  - Кросс-валидация (похоже на отложенную выборку)
  - Expanding mean (использовать только предыдущие объекты)
  - Добавить случайные шумы

МО лекция 5

Бинарная классификация - линейная регрессия:

- Линейная регрессия - пусть y = {-1;1}, тогда мы делаем регрессию в виде y = sign(X*W) и получаем бинарный классификатор (если y равен 0, то попадаем на разделяющую границу - плоскость линейная или линия)
- В качестве функционала ошибки мы можем использовать долю неправильных предсказаний - мы ее должны минимизировать
- Отступ: M_i = y_i * (w,x_i) -> если правильно, то sign(M) > 0, если неправильно, то sign(M) < 0, отступ может быть равен 0, если попали в разделяющую плоскость
- Таким образом, мы можем минимизировать Q(a,x) = 1/n * \sum [M_i < 0] - это пороговая функция потерь
- Абсолютная величина отступа - показывает уровень уверенности классификатора в ответе (чем ближе M к нулю, тем меньше уверенности в ответе)

- Пороговую функцию потерь (разрывную) тяжело минимизировать -> берем другую похожую
  - Логистическая функция log(1 + e^-M)
  - Сигмоидная 2/(1+e^-M)
  - Экспоненциальная e^-M
  - Кусочно-линейная (персептрон)
  - Кусочно-линейная (SVM)
  - Пороговая

Логистическая регрессия
- sigma(z) = 1 / (1+e^-z), тогда a(x,w) = sigma(w^T x)
- Здесь вероятностный смысл - предсказываем не классы, а вероятность класса
- p >= 0.5 -> y= +1
- p < 0.5 -> y= -1
- Мы можем взять даже квадратичную функцию потерь, потому что предсказываем вероятность (непрерывную величину), но это влечет некоторые проблемы, так как штраф маленький
- Поэтому берем log-loss ошибку - она из двух слагаемых, одно из которых обнуляется: идея в том, что при верном штраф 0, при неверном штраф очень большой -> inf
- Модуль вообще не подходит

Предсказание - сигмоида
Минимизируем - log-loss

Условия на функцию потерь: формула + вывод 1:05

Вероятностная постановка задачи p(y=+1 | x) полезна, потому что объекты с одинаковым признаком могут иметь разные значения целевой переменной

(w,x) = w^T x = log[(p(y=+1 | x) / (p(y=-1 | x)]

Log-loss в более приятном виде - подставим вместо сигмы саму функцию, тогда получим более приятный вид
L(b,X) = \sum log[1 + e^(-y_i)(w,x)]

Перспетрон Розенбланта - предшественник нейронных сетей и простейшая модель классификации
Пример весов для логического и: -1.5, 1, 1

МО лекция 6

SVM - support vector machine (метод опорных векторов)

- Выборка линейно разделима, если существует такой вектор параметров $w*$, что соответствующий классификатор $a(x)$ не допускает ошибок на этой выборке (линия идеально разделяет классы) - это идеальный случай
- Задача SVM для линейно разделимой выборки - максимизировать ширину разделяющей полосы (построить максимально уверенный классификатор)

- Нормируем параметры w и w_0 так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$

Тогда расстояние от точки x_0 до разделяющей гиперплоскости, задаваемой классификатором:

$$\rho(x_0, a) = \frac{|(w, x_0) + w_0|}{||w||}$$

- Расстояние до ближайшего объекта $x \in X$:

$$\min_{x \in X} \frac{|(w, x) + w_0|}{||w||} = \frac{1}{||w||} \min_{x \in X} |(w, x) + w_0| = \frac{\mathbf{1}}{||\boldsymbol{w}||}$$

ОПТИМИЗАЦИОННАЯ ЗАДАЧА SVM ДЛЯ РАЗДЕЛИМОЙ ВЫБОРКИ

$$\begin{cases} \frac{1}{2} ||w||^2 \to \min\limits_{w} \\ y_i\big((w, x_i) + w_0\big) \geq 1, i = 1, \dots, l \end{cases}$$

Такая задача имеет единственное решение (максимизация ширины разделяющей полосы)
Уравнение ограничение - все отступы больше единицы (выборка линейна разделима)

SVM - линейно неразделимая выборка
Существует хотя бы один объект $x \in X$, что yi $((w,xi)$ $+w0)$ <1 - то есть для одного из объектов отступ меньше 1

Алгоритм не уврен на объектах, которые ближе к раделяющей полосе -> мы будем их штрафовать
Смягчим ограничения, введя штрафы $\xi i \geq 0$: yi $((w,xi)$ $+w0)$ $\geq 1-\xi i, i=1,\ldots,l$
здесь ξi - это штраф

И получается нам нужно минимизировать штрафы (сделать поуже), но при этом максимизировать отступ (сделать пошире). Решение - это некоторый компромисс

Задача минимизации - приведение в более простой вид (безусловная задача из условной)

$$\begin{cases} \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{l}\xi_i \to \min\limits_{w,w_0,\xi_i} (1) \\ y_i\left((w,x_i)+w_0\right) \geq 1-\xi_i, i=1,\ldots,l\ (2) \\ \xi_i \geq 0, i=1,\ldots,l\ (3) \end{cases}$$

Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i\left((w,x_i)+w_0\right) = 1 - M_i(w,w_0) = 1 - M_i \\ \xi_i \geq 0 \end{cases}$$

Данная задача выпуклая и имеет единственное решение

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - M_i \\ \xi_i \geq 0 \end{cases} \Rightarrow \xi_i = \max(0, 1 - M_i)$$

Получаем безусловную задачу оптимизации:

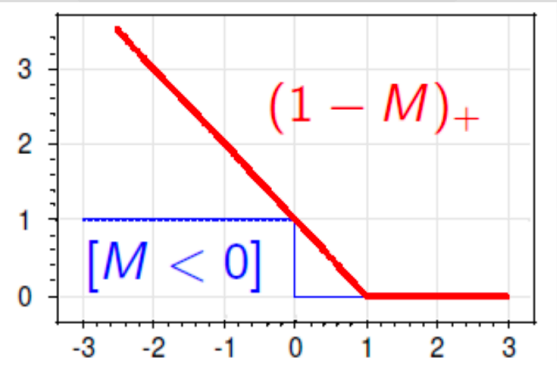
$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{l} \max(0, 1 - M_i) \rightarrow \min_{w, w_0}$$

Уравнение (2) - показывает, что выборка не является линейно разделимой
Неравенство (3) - показывает штрафы за попадание в разделяющую полосу

Первое слагаемое - сумма штрафов, второе слагаемое - регуляризация

На задачу оптимизации SVM можно смотреть, как на *оптимизацию функции потерь* $L(M) = max(0, 1 - M) = (1 - M)_+$ *с регуляризацией*:

$$Q(a, X) = \sum_{i=1}^{l} (1 - M_i)_+ + \frac{1}{2C}||w||^2 \rightarrow \min_{w, w_0}$$



С точки зрения математики SVM - лучший линейный классификатор - самые уверенные оценки

Есть гиперпараметр c > 0, чем больше c - тем более узкая полоса (штрафы сильнее и имеют больший вес)

$$\begin{cases} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \to \min\limits_{w,w_0,\xi_i} (1) \\ y_i\big((w,x_i)+w_0\big) \geq 1-\xi_i, i=1,\dots,l\ (2) \\ \xi_i \geq 0, i=1,\dots,l\ (3) \end{cases}$$

Откуда название? Каждый объект - это вектор. Попадающие в разделяющую полосу - опорные нарушители, также есть опорные граничные объекты и периферийные объекты. И мы обращаем внимание только на опорных нарушителей и на опорные граничные объекты, остальные не волнуют, поэтому алгоритм называется методом опорных векторов

SVM (максимизация разделяющей полосы, не умеет вероятности предсказывать) по своей сути совсем отличается от логистической регрессии (вероятность класса, метод максимального правдоподобия, log-loss)

-

Метрики качества классификации

1. Accuracy - плохо отражает качество при несбалансированной выборке. Несбалансированность выборки - примеры: больных меньше, чем здоровых / возвратов кредитов больше, чем невозвратов. То есть в таких случаях метрика будет высокой, а на деле его полезность нулевая

$$accuracy(a,X) = \frac{1}{n}\sum_{i=1}^{n}[a(x_i) = y_i]$$

Модель 2 лучше, потому что умеет предсказывать второй класс

e.g. Model 1 First class 950/1000 Acc = 0,95
Second 0

Model 2 First class 925/940 Acc = 0.95
Second 25/60

2. Confusion Matrix (TP,FP,TN,FN) - матрица ошибок

3. Precision (точность) = TP / (TP + FP)
   Насколько можно доверять классификаторы, если он выдал положительный результат (одобрил кредит). Какая доля из всех заемщиков вернет кредит

4. Recall (полнота) = TP / (TP + FN)
   Как много объектов положительного класса находит классификатор. Какой доли действительно положительных заемщиков алгоритм выдал кредит

5. F-мера - учитывает точность и полноту (это среднее гармоническое)
   F(a,X) = 2 * Precision * Recall / (Precision + Recall) = F1 score

Иногда нам важнее точность или наоборот важнее полнота -> мы можем регулировать наш классификатор (обычно если p > 0.5, то +1, но на самом деле мы можем брать p > 0.7 или p > 0.3)
Если порог t=0 - тогда полнота будет 1 (максимальной), при этом точность будет низкой, то есть лучше такой порог не брать
При увеличении t полнота уменьшается (могут появиться объекты положительного класса, которые мы не нашли), ***а точность возрастает*** (появляются объекты положительного класса).

6. FPR (false positive rate) = FP / (FP + TN)

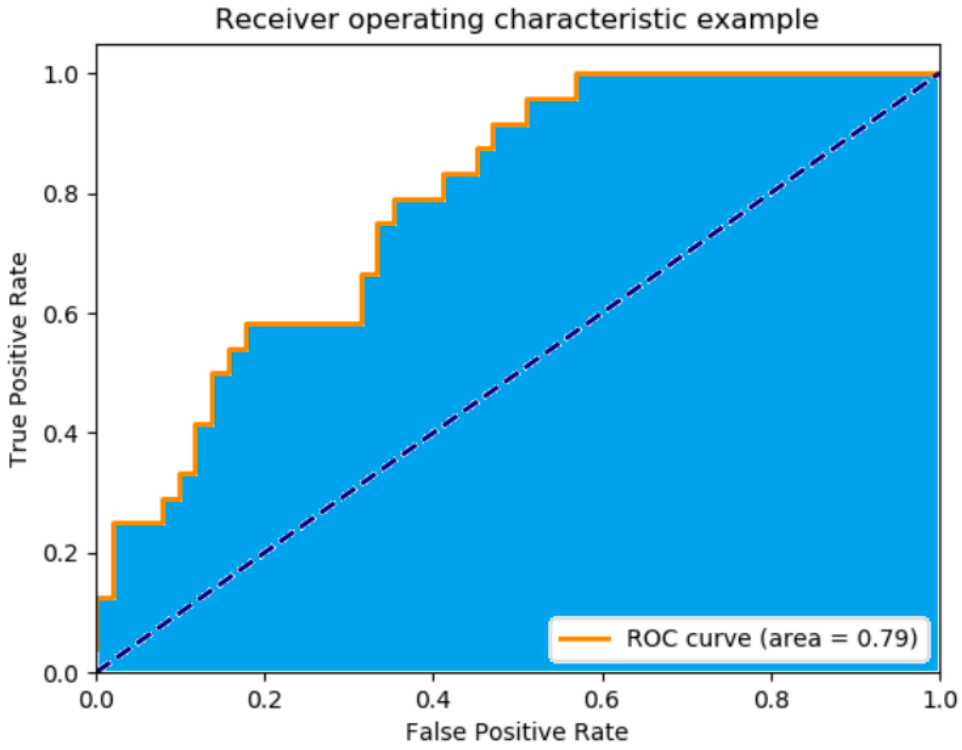
7. TPR (true positive rate) = TP / (TP + FN)

8. ROC-AUC (receiver operating characteristic - area under curve) - интегральная метрика, в таком случае она показывает результаты для всех возможных порогов, а не для какого-то одного

   Для каждого порога (если объектов 1000, то можно взять 1001 порог) мы посчитаем FPR и TPR

   Строим график на плоскости с осями FPR, TPR, смотрим на площадь под кривой (

В идеале - это ломанная линия от (0,0) до (1,0) до (1,1) - то есть чем ближе к левому верхнему углу - тем лучше. Для идеального классификатора ROC-AUC = 1, при случайном (рандом 1 или -1) классификакторе ROC-AUC будет 0.5 (под диагональной линией)



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
y	−1	+1	−1	+1	+1

- Упорядочим объекты по убыванию предсказаний:

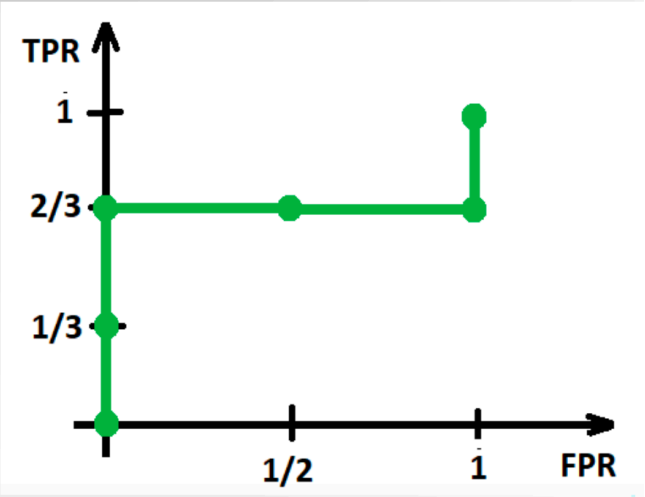
$(0.7, 0.4, 0.2, 0.1, 0.05)$

5 шаг: $t = 0$, то есть

$a(x) = [b(x) > 0]$

$$TPR = \frac{3}{3+0} = 1,$$

$$FPR = \frac{2}{2+0} = 1.$$

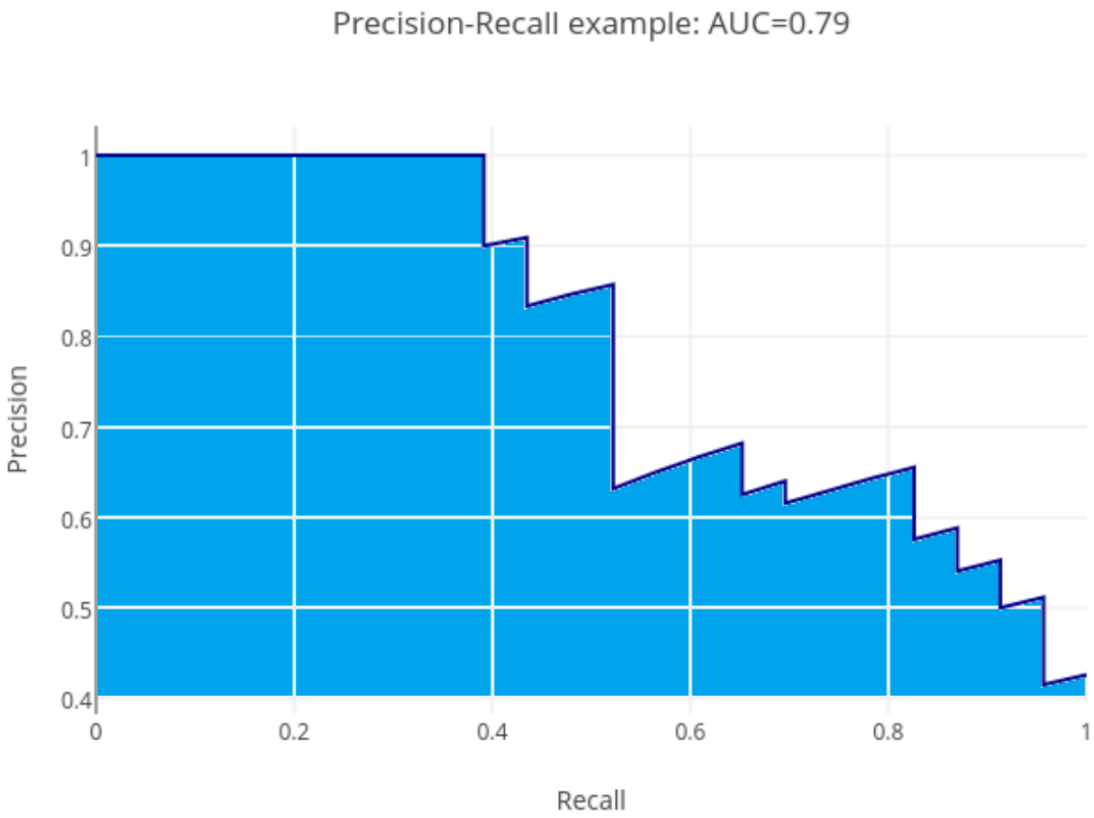


9. Индекс Джинни - удвоенная площадь между главной диагональю и ROC кривой
   Gini = 2AUC - 1

   Идейно это очень похоже на ROC-AUC

10. Precision-recall AUC (AUC-PR)
    В случае малой доли объектов положительного класса (дисбаланс классов) AUC-ROC может давать неадекватно хороший результат - но это очень спорное мнение, базово лучше использовать ROC-AUC

    Идет слева сверху вправо вниз, можно объяснить формулами, но по логике построения похожа на ROC
    В качестве метрики берем тоже площадь по кривой


Precision-Recall example: AUC=0.79
Precision
1
0.9
0.8
0.7
0.6
0.5
0.4
0
0.2
0.4
0.6
0.8
1
Recall