Задание

Решаем задачу бинарной классификации: то есть целевой признак имеет два значения, согласно традиции: «+» и «-». Есть обучающая выборка (train dataset), для которой известны метки классов.

Есть тестовая выборка (test dataset), для которой по значениям других признаков надо восстановить значение класса.

Кроме того, в первой постановке предполагается, что признаки также имеют бинарную природу.

Используем метод ленивой классификации, то есть не строим все множество классификаторов на основе обучающей выборки, а принимаем решения по классификации для конкретного объекта, как только он поступает.

Для решения задачи предлагается использовать следующий метод на основе «генераторов»

Метод на основе «генераторов»

Контексты для плюс- и минус-примеров рассматриваются по отдельности.

Для классифицируемого объекта g следует выполнить:

- Для каждого объекта из плюс-контекста рассмотреть пересечение его описания (g_i^+) и описания классифицируемого (g'), и проверить, вкладывается ли оно в описание какого-либоиз минус-примеров (g_i^-)

- Для каждого объекта из минус-контекста рассмотреть пересечение его описания (g_i^-) и описания классифицируемого (g'), и проверить, вкладывается ли оно в описание какого=либо из плюс-примеров (g_i^+)

Необходимо исследовать возможные пороговые функции классификации (например, классифицировать объект соответствующим образом, если для пересечения поддержка больше порогового значения min_supp или само пересечение больше (меньше) порогового значения min_card (max_card), разрешающие некоторую долю ошибки (параметр) на обучающей выборке.

Здесь и далее используются следующие обозначения:

$|.|$ – мощность, $(.)^+, (.)^-$ - операторы «штрихов» в положительном и отрицательном контекстах.

g' – описание классифицируемого примера («штрих» во всем контексте, то есть в объединении положительного и отрицательного контекстов).

Примеры вычислимых характеристик, которые могут участвовать в итоговом правиле классификации:

$|g' \cap g_i^+|$ – мощность пересечения

$|(g' \cap g_i^+)^+|$ – поддержка (релизована в прилагаемом скрипте)

$|(g' \cap g_i^+)^-|$ – достоверность

Пример аггрегатной функции:

$$Aggr_i|g' \cap g_i^+|: \sum_{i \in i^+} |(g' \cap g_i^+)^+|$$

Для Вашего удобства в файлах train#.csv, test#.csv помещены обучающие и тестовые данные (по массиву Tic-Tac-Toe из репозитория UCI Machine Learning Repository). Приветствуется использование других массиово данных из репозитория UCI Machine Learning Repository. Необходимо подобрать параметры модели ленивого обучения, дающие наибольшую среднюю (по всем #) точность по скользящему контролю (кросс-валидации) на тестовых выборках.

Сроки

01/12 - рабочий код для всего цикла анализа алгоритмов

08/12 - реализация и отчет по собственным доработкам для одного dataset'a

15/12 - реализация поиска оптимальных параметров обучения на нескольких dataset'ax, поиск "глобального" оптимума

22/12 - поддержка нешкалированных данных, финальный отчет

Модификации

1) собственный набор данных (dataset)

2) пересмотр пространства признаков в имеющихся dataset'ax

3) реализация ленивой распределенной схемы

4) поддержка онлайн схемы поступления данных (учет корректировки реального значения после прогноза)2) пересмотр пространства признаков в имеющихся dataset'ax

Краткий туториал по работе с github'ом

https://guides.github.com/activities/hello-world/

Общая схема работы алгоритма

На входе есть файл с данными, для которых известен целевой класс.

Используется метод скользящего контроля для оценки качества алгоритма. То есть файл делится на k частей – обучаемся на (k-1)й, на k-й предсказываем, по итогам вычисляем метрики качества, усредняем по k запускам

Для запуска алгоритма признаки должны быть бинарными – необходимо написать реализацию шкалирования (одна из модификаций – отказаться от данного шага)

В разработанном методе должны быть параметры алгоритма (например, минимальная поддержка в предсказываемом классе).

Подбор оптимального параметра зависит от целевой метрики

Отчет представляет собой описание идеи модификации алгоритма + анализ поведения при разных значениях параметра

Основой для сравнения качества являются: сложность, усредненные по скользящему контролю метрики качества

Необходимым условием сдачи является размещение исходного кода в пределах github’a.