Классификация и оценка качества классификации

Михаил Старцев, Влад Шахуро

Обзор задания

Данное задание — знакомство с классификаторами kNN и SVM, а также с базовыми приёмами оценки и сравнения алгоритмов машинного обучени.

Описание задания

Как в базовой, так и в бонусной частях, необходимо сравнить kNN (`ClassificationKNN` в Matlab, `sklearn.neighbors.KNeighborsClassifier` в Python) с SVM (библиотека `liblinear` для Matlab, `sklearn.svm.SVC` в Python) с разными параметрами. Дополнительный **1 балл** получит половина участников, алгоритмы которых показали лучшие результаты на скрытой выборке.

Помимо кода проверяться будут файлы с графиками зависимости точности классификации от параметров — `kNN-base.png`, `SVM-base.png`, `kNN-bonus.png`, `SVM-bonus.png`. В базовой части для каждого классификатора достаточно перебора минимум 3 параметров, в бонусной — 10 параметров.

Базовая часть (2 балла)

Сравнение классификаторов и выбор параметров следует производить путём запуска на части исходной обучающей выборки, которая для непосредственно для обучения не использовалась. Требуется случайным образом отделить 20% обучающей выборки, обучиться на оставшихся 80%, протестировать на отделённой части.

Бонусная часть (2 балла)

Для сравнения классификаторов нужно воспользоваться скользящим контролем (для этого можно использовать библиотечные функции или написать свою реализацию). В результате скользящего контроля получается вектор точностей классификатора на разных запусках, для сравнения следует использовать среднее значение точности и выборочную дисперсию этого вектора (для отображения дисперсии точности на графиках можно использовать box plot). Для поиска оптимальных параметров можно использовать grid search с равномерной или логарифмической шкалой.

Интерфейс программы, данные и скрипт для тестирования

Необходимо реализовать функцию `fit_and_classify`, принимающую на вход обучающую выборку (матрицу векторов-признаков, где каждая строка соответствует одному объекту обучающей выборки), метки объектов обучающей выборки (вектор числовых меток), и тестовую выборку (в том же формате, что и обучающая выборка), и выдающую на выходе вектор предсказанных меток для объектов тестовой выборки. Функция должна реализовывать классификацию выбранным на стадии сравнения качества классификатором с соответствующими параметрами. Скрипт для тестирования `fit_and_classify_test` принимает в качестве параметров два пути к файлам с тренировочной и тестовой выборками и печатает точность распознавания тестовой выборки.

Обучающая выборка представляет собой матрицу объектов-признаков и вектор ответов: файл `train.csv` содержит столбцы $F1 \dots F86, Y$, где в первых столбцах содержатся признаки, в последнем — метка класса объекта. Признаки объектов представляют из себя HOG для изображений рукописных цифр, метка — сама цифра (1, 2, 3).

Для воспроизводимости результатов при разбиении выборки или скользящем контроле следует задать порождающий элемент генератору псевдослучайных чисел (seed).

Для использования `liblinear` в Matlab (прилагается с каркасом) её нужно скомпилировать, вызвав скрипт `make.m` из папки `liblinear-1.94/matlab`, и добавить эту папку в путь в среде Matlab (*«Add to path»*).

Полезные ресурсы

Box plot.
Сайт библиотеки `liblinear`. Там можно скачать бинарные файлы для Windows и прочитать руководству по подбору параметров SVM (grid search).