SkillFactory

Обзор некоторых моделей для классификации

Шпилевский Яромир

Ведущий разработчик First Line Software

Agenda

- Задача классификации.
- Решение с помощью SVM.
- Решение с помощью KNN.
- Решение с помощью Decision Tree.
- Решение с помощью Random Forest.

В Machine Learning (ML) требуется широта взглядов

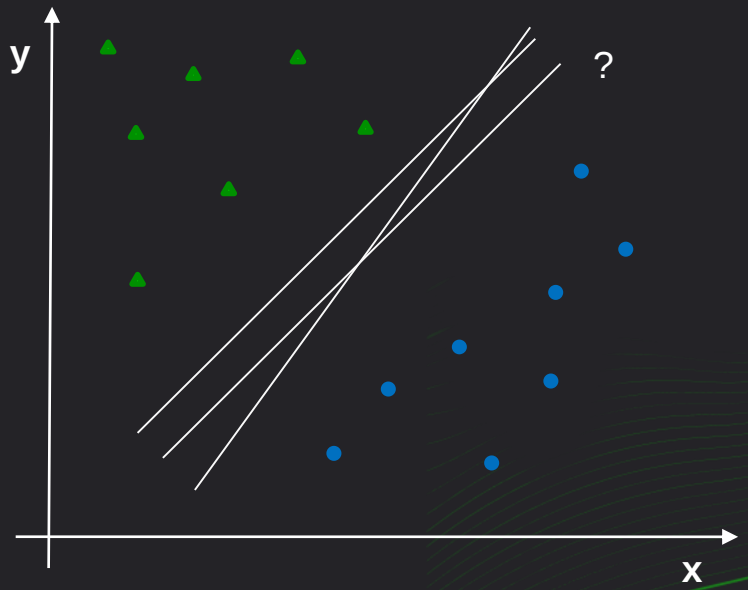
- ML – обширная тема: существует множество задач и изобретено множество моделей.
  - Соотношение «многие ко многим»:
    - Класс задач может решаться несколькими моделями.
    - Модель может решать несколько классов задач.
- Решение конкретной задачи – творческий поиск модели или их комбинации.
- Нужно знать, какие изобретены модели, их сильные и слабые стороны.

Задача классификации

- Задача классификации: есть пространство объектов, каждый объект обладает какими-то свойствами и относится к одному из N классов. Задача: обучившись на выборке объектов, уметь сказать для произвольного объекта, к какому классу он относится.
- Вы уже рассмотрели многослойный персептрон.
- Задача классификации может так же решаться другими моделями (необязательно глубокими нейронными сетями):
  - Support Vector Machine (SVM)
  - K Nearest Neighbours (KNN)
  - Decision Tree
  - Random Forest
  - …
- Для широты кругозора их полезно знать, могут пригодиться. ;)

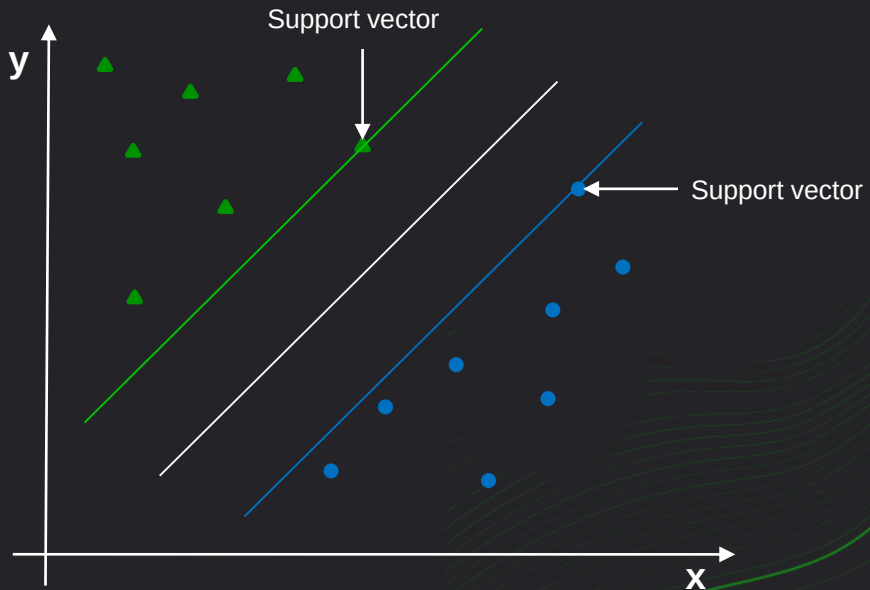
SVM. Гиперплоскость

- Основная идея – построить разграничительную прямую (или, в общем случае, гиперплоскость), отделяющую один класс от другого.
- Таких прямых бесконечно много. Какая из них оптимальная?



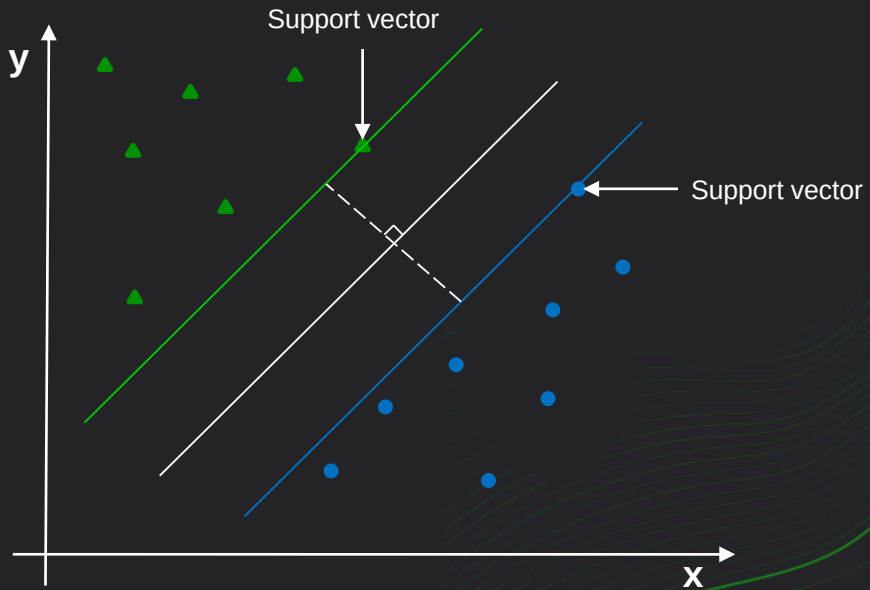
SVM. Support Vector

- Есть ближайший объект каждого класса, через который можно провести прямую, параллельную разграничительной прямой.
- Такие объекты называются опорными векторами (support vector).



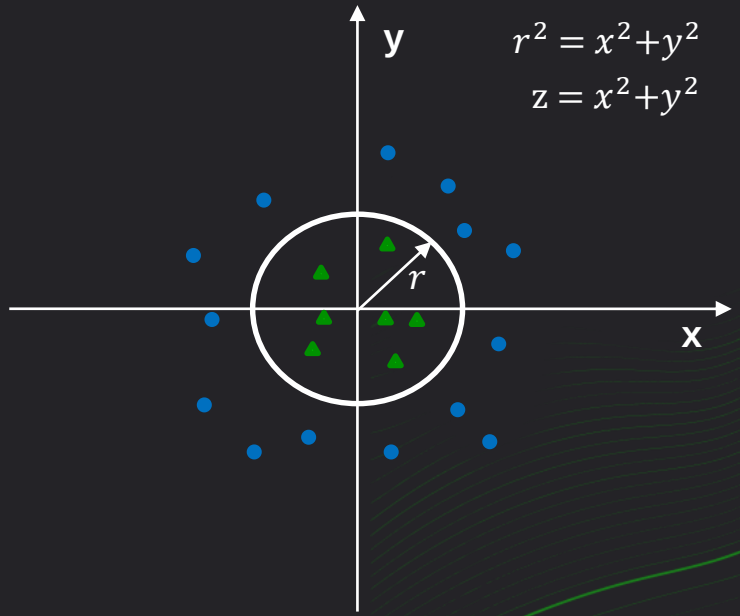
SVM. Критерий оптимальности

- Оптимальная прямая – с максимальным расстоянием до параллельных прямых, проведённых через опорные векторы.



SVM. Kernel

- Функция, описывающая разделяющую гиперплоскость, не обязательно линейная.
- Эта функция называется kernel.



SVM. Реализация в sklearn

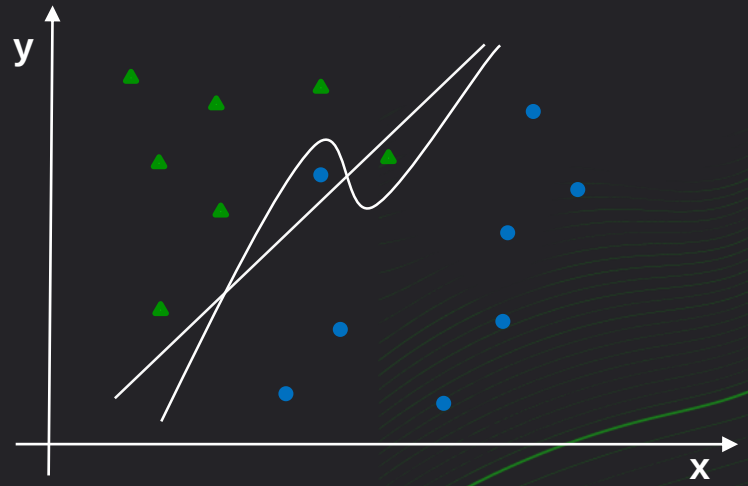
- from sklearn.svm import SVC
- Support Vector Classifier
- Синоним. То, что мы сейчас рассмотрели – и есть классификатор на основе support vectors.

SVM. Параметры. kernel

- kernel – функция разделяющей гиперплоскости
  - linear – линейная
  - poly – полином
  - rbf – радиально-базисная функция
  - sigmoid – сигмоид
  - precomputed – произвольный Python callable
- По умолчанию – rbf.
- Радиально-базисная функция – функция, которая зависит только от расстояния от точек до начала координат, либо до другой фиксированной точки (центра).

SVM. Параметры. C

- C – параметр регуляризации. Положительное вещественное число.
  - Маленькое значение – менее «петляющая» гиперплоскость. Но есть риск неправильно классифицировать объекты, близкие к разделяющей гиперплоскости.
  - Большое значение – лучше классификация, но есть риск переобучения (overfitting), гиперплоскость может получится недостаточно обобщающей.



SVM. Параметры. Гамма

- Гамма – коэффициент kernel функции.
- Определяет насколько далеко распространяется влияние каждой точки на разделяющую гиперплоскость.
- Большое значение – более «петляющая» гиперплоскость. Близкие к ней точки сильно искривляют её.
- Маленькое значение – более «выпрямленная» линия. Дальние точки тоже получают большое влияние и «выпрямляют» её.

SVM. Параметры реализации sklearn. Probability

- SVM не позволяет получить вероятностные оценки результатов в явном виде.
- probability = true позволяет оценить их с помощью различных статистических методов.
- По умолчанию false.

SVM. Плюсы и минусы

- + Хорошо справляется с задачами высокой размерности.
- - Не предоставляет вероятностных оценок результатов в явном виде.

K Nearest Neighbors (KNN)

- К ближайших соседей.
- «Ленивая» модель. Не пытается вывести общую закономерность. Просто хранит набор расстояний.
- К ближайших соседей «голосуют», к какому классу принадлежит объект.
- Победа простым большинством.

KNN. sklearn

- from sklearn.neighbors import KNeighborsClassifier
- Параметры:
  - n_neighbors – количество соседей, которые «голосуют».

KNN. Плюсы и минусы

- + Прост в реализации.
- + Устойчив к шумам во входных данных.
- + Эффективен на больших датасетах.
- - Большие вычислительные затраты на подбор K и расчёт расстояний.

Decision Tree

- Дерево решений.
- Набор правил.
- В соответствии с каждым правилом выбор ветки.
- Условия имеют вероятностные характеристики:
  - «С вероятностью 67%, нам нужно пойти по этой ветке.»
- Обучение модели подбирает эти вероятности.

Decision Tree. Плюсы и минусы

- + Простое в понимании и визуализации.
- + Не требуется трудоёмкой предварительной подготовки датасета.
- + Может использоваться как для числовых данных, так и просто для аннотированных объектов (аннотированных принадлежностью к классу).
- - Плохо обобщает закономерность.
- - Нестабильность: небольшое изменение в датасете может генерировать принципиально другую структуру.

Random Forest

- Случайный лес.
- Группа деревьев решений для различных изменений обучающей выборки.
- Из исходной обучающей выборки генерируется несколько обучающих выборок, на каждой обучается дерево решений.
- Каждое дерево решений говорит свой ответ.
- Результат усредняется.
- Случайный лес – способ борьбы с переобучением, свойственным деревьям решений, и способ борьбы с их нестабильностью.
    - «Не баг, а фича.» ☺ Изменение входного датасета сильно влияет на выход? Так давайте используем это – нагенерируем много разных деревьев, будем смотреть, какой ответ говорит каждое, и усреднять ответы в итоговый ответ.

Random Forest. Плюсы и минусы

- + Меньше склонен к переобучению, по сравнению с одним деревом решений.
- + Меньше склонен к нестабильности, по сравнению с одним деревом решений.
- - Сложный алгоритм.
- - Долгое время работы. Не подходит для задач в реальном времени (мягком реальном времени, естественно).

Резюме

- Рассмотрена задача классификации.
- Рассмотрено решение с помощью SVM.
- Рассмотрено решение с помощью KNN.
- Рассмотрено решение с помощью Decision Tree.
- Рассмотрено решение с помощью Random Forest.

Вопросы для самоконтроля

- Что такое классификация?
- Как устроен SVM?
- Как устроен KNN?
- В чем подход Decision Tree?
- В чем подход Random Forest?
- В чем улучшения при подходе Random Forest?

Спасибо!