План лекции

Мотивация

Обзор методов построения алгоритмических композиций

Стохастические методы построения алгоритмических композиций

Random Forest

Выбираем ноутбук

Параметры ноутбука

- Камень
- Память
- Видеокарта
- Диагональ монитора
- и т.д

Воспользуемся экспертным мнением

- Консультант в магазине
- Сосед “программист”
- Друзья
- Форумы

Простое голосование

Simple Voting

$$h(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} a_i(\mathbf{x})$$

Взвешенное голосование

Weighted Voting

$$h(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} b_i a_i(\mathbf{x}), \;\; b_i \in \mathcal{R}$$

Смесь экспертов

Mixture of Experts

$$h(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} b_i(\mathbf{x})a_i(\mathbf{x}),\ \ b_i(\mathbf{x}) : \mathcal{X} \to \mathcal{R}$$

Топологические композиции

Модельные деревья решений (Model decision trees)

- Поместим в листья деревьев какие-нибудь алгоритмы вместо констант

Нейронные сети

Neural networks

$h(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$ - композиция линейных моделей

Текущее положение дел

Decision Trees Compositions vs Neural Nets

Neural nets

Где нейронные сети побеждают?

- Computer Vision (ImageNet)
- Speech recognition

Decision trees compositions

Где побеждают ансамбли деревьев решений?

- Learning to rank (Yahoo Learning to rank challenge 2010)
- Text categorization (NIPS 2006)
- Везде

Взгляд с точки зрения функционального анализа

Гладкие функции

$$h(\mathbf{x}) = \sum \sigma(\ldots \sum \sigma(\mathbf{w}^T \mathbf{x}))$$

Кусочно-постоянные функции

$$h(\mathbf{x}) = \sum_d c_d I\{\mathbf{x} \in R_d\}$$

Методы построения алгоритмических композиций

Стохастические методы

Бустинг

Мета алгоритмы. Stacking

Stacking

Мета-алгоритм

Netflix Challenge

- Задача предсказания оценки фильму
- ПФ - 1000000$
- Победил Stacking на “зоопарке” алгоритмов

Немного теории

$\theta_0 + \theta_1 x$

$\theta_0 + \theta_1 x + \theta_2 x^2$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Недообучение - сложность модели недостаточна. Данные имеют более сложную природу.
Переобучение - сложность модели избыточна. Модель слишком настроена на выборку. Обобщающая способность алгоритма низка.

The bias-variance Decomposition

- Задача регресии
- $y = f(x) + \epsilon$
- $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$
- Ищем $h(\mathbf{x})$ аппроксимирующую $f(\mathbf{x})$ Таким образом, мы можем оценить ожидание среднеквадратичной ошибки для некоторой точки $\mathbf{x}_0$

$$Err(\mathbf{x}_0) = E[(y - h(\mathbf{x}_0))^2]$$

Это можно переписать в виде

$$Err(\mathbf{x}_0) = (E[h(\mathbf{x}_0)] - f(\mathbf{x}_0))^2 + E[h(\mathbf{x}_0) - E(h(\mathbf{x}_0))]^2 + \sigma_\epsilon^2$$

$$Err(\mathbf{x}_0) = Bias^2 + Variance + Noise(Irreducible\ Error)$$

Bias and variance tradeoff

Bias and variance tradeoff

Стохастические методы

Стратегии

- **Bagging = Bootstrap aggregation**. Обучаем алгоритмы по случайным подвыборкам размера N, полученным с помощью выбора с возвращением.
- **RSM = Random Subspace Method**. Метод случайных подпространств. Обучаем алгоритмы по случайным подмножествам признаков.

Благодаря описанным стратегиям добиваемся максимального различия между базовыми алгоритмами.

Bagging & RSM

```
function BaggingRSM(X, F, poi, pof, T):
    # X - point from dataset
    # F - original set of features
    # poi - percent of items
    # pof - percent of features
    # T - number of algorithms
    for j in 1..T:
        fsubset = sample_features(F, pof)
        xsubset = sample_items(X, poi)
        a_j = build_base_model(xsubset, fsubset)

    return a_1 ... a_T
```

Bagging & RSM

Геометрическая интерпретация

Bias-Variance for Bagging & RSM

Модель простого голосования

$$h(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} a(\mathbf{x})$$

Смещение и разброс

$$Bias^2 = (E[h(\mathbf{x}_0)] - f(\mathbf{x}_0))^2 = (E[a(\mathbf{x}_0)] - f(\mathbf{x}_0))^2$$

$$Variance = E[h(\mathbf{x}_0) - E(h(\mathbf{x}_0))]^2 =$$

$$= \frac{1}{T}\sum_{i=1}^{T} E[a_i(\mathbf{x}_0) - E[a_i(\mathbf{x}_0)]]^2 +$$

$$+\frac{1}{T}\sum_{i\neq j} E[(a_i(\mathbf{x}_0) - E[a_i(\mathbf{x}_0)])(a_j(\mathbf{x}_0) - E[a_j(\mathbf{x}_0)])] =$$

$$= \frac{1}{T}\sum_{i=1}^{T} E[a_i(\mathbf{x}_0) - E[a_i(\mathbf{x}_0)]]^2 + \frac{1}{T}\sum_{i\neq j} cov(a_i(\mathbf{x}_0), a_j(\mathbf{x}_0))$$

Random Forest

Случайный лес (random forest) - bagging & rsm над решающими деревьями.

“Неустойчивость” деревьев решений

- Незначительные изменения в данных приводят к значительным изменениям в топологии дерева

Out of bag samples

Random Forest & overfitting

- Не переобучается с ростом числа алгоритмов

Feature impotance

Random Forest Clustering

Концепция

- В один кластер попадают те примеры, которые чаще всего вместе попадали в листья деревьев

Random Forest. Итоги

- ✔ Алгоритм прост
- ✔ Не переобучается
- ✔ Хорошо параллелится
- ✔ Не требует сложной настройки параметров
- ✔ Не требует нормализации данных (деревья решений)
- ✖ Модели получаются большие и неинтерпретируемые
- ✖ Плохо работает с полиномиальными зависимостями.
- ✖ “Медленно” работает для большого объема данных.