Шпаргалка по деревьям решений / Концепции

Cheatsheet (XeLaTeX)

Краткий справочник

April 2, 2025

Contents

Деревья Решений: Базовый Блок Ансамблей

Дерево решений — это простой, но мощный алгоритм, который лежит в основе многих продвинутых методов, таких как Случайный Лес (Random Forest) и Градиентный Бустинг. Понимание деревьев — ключ к пониманию ансамблей. Представь его как блок-схему или игру в "20 вопросов", где на каждом шаге мы задаем вопрос о данных, чтобы прийти к финальному ответу (прогнозу).

1 Как Дерево Принимает Решение?

Аналогия: Игра в "Угадай Животное"

Представь, что ты пытаешься угадать животное, задавая вопросы типа "У него есть перья?", "Он умеет летать?", "Он живет в воде?". Дерево решений работает похожим образом:

- **Узлы (Nodes):** Каждый узел — это вопрос (проверка условия) по одному из признаков (например, "Возраст > 30?").
- **Ветви (Edges/Branches):** Ответы на вопрос ("Да" / "Нет"), ведущие к следующему узлу.
- **Листья (Leaves):** Конечные узлы, где содержится прогноз (например, класс "Кликнет" / "Не кликнет" или среднее значение для регрессии).

Объект (например, пользователь) "проходит" по дереву от корня вниз, отвечая на вопросы в узлах, пока не достигнет листа. Прогноз в этом листе и будет результатом для данного объекта.

2 Выбор Лучшего Вопроса (Разделения)

Цель: Сделать Группы Чище

На каждом шаге дерево ищет **лучший вопрос** (признак и пороговое значение), который разделит текущие данные на две максимально "чистые" группы по целевой переменной.

- **Аналогия:** Представь, что у тебя корзина с яблоками и грушами. Хороший вопрос ("Это круглое?") поможет разделить их лучше, чем плохой ("Это тяжелее 100г?").
- **Чистота (Purity):** Группа считается чистой, если в ней преобладают объекты одного класса (в задаче классификации).
- **Меры Нечистоты (Impurity Measures):** Для оценки "качества" разделения используются специальные метрики. Чем ниже значение метрики после разделения, тем лучше. Основные:
  - **Критерий Джини (Gini Impurity):** Измеряет вероятность того, что случайно выбранный элемент из набора будет неправильно классифицирован, если его класс случайно выбирается в соответствии с распределением классов в наборе. Формула: $G = 1 - \sum_{k=1}^{K} p_k^2$, где p_k — доля объектов класса k. *Интуиция: ниже Gini — чище узел.*
  - **Энтропия (Entropy):** Мера хаоса или неопределенности в узле. Используется в алгоритмах ID3, C4.5. Формула: $E = -\sum_{k=1}^{K} p_k \log_2(p_k)$. *Интуиция: ниже энтропия — меньше хаоса, чище узел.*
- **Information Gain (Прирост информации):** Мера того, насколько разделение **уменьшает нечистоту** (измеренную с помощью Gini или Entropy). Рассчитывается как нечистота родителя минус средневзвешенная нечистота дочерних узлов. Дерево ищет разделение, которое дает **максимальный Information Gain**.

Процесс повторяется рекурсивно для каждого нового узла, пока не будет выполнен критерий остановки. В задачах **регрессии** вместо мер нечистоты используются критерии, основанные на уменьшении дисперсии (Variance Reduction), например, среднеквадратичная ошибка (MSE).

3 Переобучение: Проклятие Деревьев

Почему Деревья Легко Переобучаются?

Деревья по своей природе **очень гибкие** и могут строить очень сложные структуры. Если не ограничивать их рост, они будут продолжать делиться, пока в каждом листе не останется минимальное количество объектов (в идеале — один).

- **Аналогия:** Представь студента, который не выучил общие правила, а просто **зазубрил ответы** на все вопросы из тренировочного билета. На экзамене с новыми вопросами он провалится.
- **Результат:** Дерево идеально "подгоняется" под обучающие данные, запоминая даже шум и выбросы. Оно показывает отличные метрики на обучении, но **плохо обобщает** знания на новые, невиданные ранее данные (тестовый набор).

Такое поведение — классический пример **высокой дисперсии (high variance)** модели при потенциально низкой предвзятости (low bias) на обучающих данных.

4 Методы Регуляризации (Ограничения Роста)

Контроль Сложности Во Время Роста (Pre-pruning)

Чтобы дерево не "зубрило", а "учило", его рост ограничивают с помощью гиперпараметров (задаются ДО обучения):

- `max_depth`: **Максимальная глубина дерева.** Ограничивает количество "вопросов" на пути от корня к листу. Меньшая глубина — проще дерево. *Аналогия: Ограничить количество вопросов в игре "Угадай животное".*
- `min_samples_split`: **Минимальное количество объектов в узле**, необходимое для его дальнейшего разделения. Если объектов меньше — узел становится листом. Предотвращает деление на очень маленьких, возможно, шумовых группах. *Аналогия: Не делить группу людей, если их меньше 5 человек.*
- `min_samples_leaf`: **Минимальное количество объектов в листовом узле.** Гарантирует, что каждый прогноз (лист) основан на достаточном количестве примеров. *Аналогия: Каждый финальный ответ должен подтверждаться мнением как минимум 3 экспертов.*
- `max_features`: Максимальное количество признаков, рассматриваемых при поиске лучшего разделения. Вносит случайность, что **уменьшает корреляцию** между деревьями в ансамблях (например, в Случайном Лесу).

Подбор оптимальных значений этих параметров — задача **кросс-валидации**.

Идея Прунинга (Pruning - "Подрезка")

Это метод **упрощения дерева после построения (Post-pruning)**. Прунинг применяется **ПОСЛЕ** того, как дерево уже построено (часто до максимальной глубины).

- **Идея:** Упростить дерево, удаляя ("срезая") ветви или узлы, которые вносят малый вклад в точность на валидационном наборе данных или слишком сильно усложняют модель.
- **Типы (основной):** **Cost Complexity Pruning (CCP)** / Pruning по минимальной стоимости-сложности. Ищет баланс между точностью и сложностью дерева. Параметр `ccp_alpha` в scikit-learn контролирует степень прунинга. Значение $\alpha = 0$ означает отсутствие прунинга, а увеличение α увеличивает "штраф" за сложность, приводя к более сильной подрезке.
- **Преимущество:** Иногда позволяет найти более оптимальную структуру, чем простое ограничение роста гиперпараметрами (pre-pruning).