Шпаргалка по Random forest / Концепции

Cheatsheet (XeLaTeX)

Краткий справочник

April 2, 2025

Contents

Случайный Лес: Введение

Случайный Лес (Random Forest, RF) — это ансамблевый метод машинного обучения, который строит множество деревьев решений во время обучения и выводит класс, который является модой классов (классификация) или средним предсказанием (регрессия) отдельных деревьев. Это один из самых популярных и эффективных "из коробки" алгоритмов.
Аналогия: Представь, что тебе нужно принять важное решение. Вместо того чтобы спросить одного эксперта (одно дерево), ты собираешь **комитет разных экспертов** (много деревьев), каждый из которых смотрит на проблему немного под своим углом, а затем принимаешь решение на основе их коллективного мнения. Случайный лес делает то же самое, но с деревьями решений.

1 Идея Бэггинга (Bagging)

Bagging

Бэггинг — это основной принцип, лежащий в основе Случайного Леса. Он состоит из двух шагов:

- **Bootstrap (Бутстрэп):** Создается множество (N) подвыборок из исходного обучающего датасета. Каждая подвыборка формируется путем случайного выбора объектов **с возвращением**. Это означает, что некоторые объекты могут попасть в одну подвыборку несколько раз, а некоторые — ни разу. Размер каждой подвыборки обычно равен размеру исходного датасета.
- **Aggregating (Агрегация):** На каждой подвыборке независимо обучается своя модель (в случае RF — дерево решений). Затем предсказания всех N моделей усредняются (для регрессии) или определяется самый популярный класс (для классификации — голосование большинством).

Цель бэггинга: Снизить **дисперсию (variance)** модели. Индивидуальные деревья могут сильно переобучаться (высокая дисперсия), но усреднение их предсказаний сглаживает ошибки и делает итоговую модель более устойчивой. Объекты, не попавшие в конкретную бутстрэп-выборку ($\approx$ 37%), называются **Out-of-Bag (OOB)** и могут использоваться для оценки качества модели (OOB-оценка) без необходимости отдельной валидационной выборки.

2 Случайный Выбор Признаков (Feature Subsampling)

Дополнительная Случайность

В отличие от простого бэггинга деревьев, Случайный Лес вносит **дополнительный элемент случайности** при построении каждого дерева:

- При поиске лучшего разбиения (split) в каждом узле дерева, алгоритм рассматривает не все доступные признаки, а только их **случайное подмножество** (размер подмножества, `max_features`, является гиперпараметром).
- Для задачи классификации обычно берут $\sqrt{p}$ признаков, для регрессии — $p/3$, где p — общее число признаков.

Зачем это нужно? Это делается для **декорреляции** деревьев. Если бы все деревья видели все признаки, и был бы один очень сильный признак, большинство деревьев использовали бы его для первого разбиения. В результате деревья были бы очень похожи (скоррелированы), и усреднение не дало бы такого сильного эффекта снижения дисперсии. Случайный выбор признаков заставляет деревья быть более разнообразными.
Аналогия: Возвращаясь к комитету экспертов. Чтобы они не пришли к одному и тому же выводу, опираясь на самый очевидный факт, ты просишь каждого эксперта при анализе сосредоточиться только на **случайном наборе аспектов** проблемы. Это побуждает их исследовать разные стороны вопроса.

3 Как Уменьшает Дисперсию

Борьба с Переобучением через Усреднение

Ключевая сила Случайного Леса — в его способности значительно **уменьшать дисперсию** по сравнению с одним деревом решений, не сильно увеличивая (или даже немного уменьшая) **смещение (bias)**.

- **Одно дерево решений:** Имеет низкое смещение (может хорошо подогнаться под обучающие данные), но высокую дисперсию (сильно меняется при небольшом изменении данных, легко переобучается).
- **Случайный Лес:**
  - **Бэггинг (усреднение):** Усреднение предсказаний N моделей, ошибки которых не полностью скоррелированы, приводит к снижению общей дисперсии ансамбля. Чем больше деревьев (N), тем ниже дисперсия (до определенного предела).
  - **Случайный выбор признаков (декорреляция):** Уменьшает корреляцию между деревьями, что делает усреднение еще более эффективным для снижения дисперсии.

В итоге, RF получает модель, которая все еще достаточно гибкая (относительно низкое смещение, унаследованное от деревьев), но гораздо более стабильная и устойчивая к переобучению (значительно сниженная дисперсия). Это классический пример улучшения модели через управление **компромиссом смещения-дисперсии (Bias-Variance Tradeoff)**.
Аналогия "Мудрость Толпы": Один человек может сильно ошибаться в оценке (высокая дисперсия), но если усреднить оценки большой группы людей (где ошибки случайны и не связаны), итоговая оценка будет гораздо ближе к истине (низкая дисперсия).

4 Важность Признаков (Feature Importance)

Оценка Влияния Признаков

Хотя Случайный Лес менее интерпретируем, чем одно дерево, он позволяет оценить **важность** каждого признака для предсказания. Основные подходы:

- **Mean Decrease in Impurity (MDI) / Gini Importance:**
  - *Как считается:* Для каждого признака суммируется уменьшение критерия неопределенности (например, индекса Джини для классификации или MSE для регрессии) по всем узлам всех деревьев, где этот признак использовался для разбиения. Затем эти суммы усредняются по всем деревьям и нормализуются.
  - *Плюсы:* Быстро считается (информация доступна после обучения).
  - *Минусы:* Склонен завышать важность числовых признаков и признаков с большим количеством категорий. Может давать неверные результаты для скоррелированных признаков.
- **Mean Decrease in Accuracy (MDA) / Permutation Importance:**
  - *Как считается:* 1. Оценивается качество модели (например, Accuracy, R2) на отложенной выборке (Out-of-Bag или отдельный validation set). 2. Значения одного признака случайно перемешиваются во всей отложенной выборке (нарушается связь между этим признаком и целевой переменной). 3. Качество модели повторно оценивается на перемешанных данных. 4. Уменьшение качества модели и есть важность этого признака. Повторяется для всех признаков.
  - *Плюсы:* Более надежен, чем MDI, особенно при наличии скоррелированных признаков. Показывает реальное влияние признака на **производительность модели** на новых данных. Идея метода часто **модель-агностична** (применима не только к RF).
  - *Минусы:* Требует дополнительных вычислений (может быть **медленным** на больших данных или при большом числе признаков). Результат может зависеть от конкретной отложенной выборки.

Что Спрашивают на Собеседованиях

Часто спрашивают разницу между MDI и Permutation Importance. Важно понимать:

- MDI измеряет, насколько признак **использовался** деревьями при построении (на основе обучающей выборки).
- Permutation Importance измеряет, насколько признак **влияет** на итоговое качество предсказания модели (на основе отложенной выборки). Permutation Importance обычно считается более надежным показателем реальной важности.

5 Ключевые Гиперпараметры

Основные параметры для настройки Случайного Леса:

- **`n_estimators`:** Количество деревьев в лесу. Чем больше, тем лучше (до некоторого плато), но дольше обучение. Обычно выбирают достаточно большим (100, 500, 1000+).
- **`max_features`:** Количество признаков, рассматриваемых при поиске лучшего сплита в каждом узле. Ключевой параметр для контроля корреляции деревьев и борьбы с переобучением. Значения по умолчанию ($\sqrt{p}$ / $p/3$) часто работают хорошо, но стоит подбирать.
- **Параметры деревьев:** Гиперпараметры базовых деревьев решений также влияют на лес (например, `max_depth`, `min_samples_split`, `min_samples_leaf`). Часто оставляют деревья достаточно глубокими в RF, полагаясь на усреднение для борьбы с переобучением, но иногда их ограничение тоже помогает.

6 Сравнение с Конкурентами

RF vs. Одно Дерево Решений

- **Плюсы RF:**
  - Значительно **меньше переобучается**, более **устойчив** (низкая дисперсия).
  - Обычно **выше точность** и обобщающая способность.
- **Минусы RF:**
  - **Менее интерпретируем** ("черный ящик" по сравнению с одним деревом).
  - Требует **больше ресурсов** для обучения и предсказания (память и время).

RF vs. Линейные Модели (Логистическая/Линейная Регрессия)

- **Плюсы RF:**
  - Легко улавливает **нелинейные зависимости** и **взаимодействия** между признаками без необходимости их явного добавления (как в линейных моделях).
  - **Не требует масштабирования** признаков (деревьям не важен масштаб).
  - Менее чувствителен к **выбросам** (решающие правила деревьев устойчивы).
  - Хорошо работает "из коробки" с минимальной настройкой.
- **Минусы RF:**
  - **Менее интерпретируем**, чем линейные модели (где можно смотреть на веса).
  - Может быть **медленнее** в обучении и предсказании на очень больших данных.
  - Плохо **экстраполирует** за пределы диапазона значений признаков, виденных в обучении (предсказание ограничено значениями в "листьях"). Линейные модели могут экстраполировать.
  - Может требовать **больше памяти**.