Ансамблирование алгортимов машинного обучения

Рыжков Александр

3 декабря 2014 года

Содержание

Содержание

Введение

- Что такое композиции и какие они бывают?
- Зачем их использовать?
- Как правильно строить композиции?
- Как выбрать тип композиции?

Содержание

Простое голосование

Простое голосование — усреднение ответов алгоритма:

$$Y_{answer} = C\left(\frac{1}{T}\sum_{i=1}^{T} b_i(x)\right)$$

Примеры алгоритмов:

- Random Forest
- Пользовательская композиция

Взвешенное голосование

Взвешенное голосование — усреднение ответов алгоритма с весами:

$$Y_{answer} = C\Big(\sum_{i=1}^{T} w_i b_i(x)\Big)$$

$$\sum_{i=1}^{T} w_i = 1,\ w_i \geqslant 0$$

Примеры алгоритмов:

- AdaBoost
- Пользовательская весовая схема

Смеси экспертов

Смесь экспертов — задание областей «уверенности» алгоритмов:

$$Y_{answer} = C\left(\sum_{i=1}^{T} w_i(x) b_i(x)\right)$$

$$\sum_{i=1}^{T} w_i(x) = 1,\ \forall x \in X$$

В случае выпуклой функции потерь настраивается алгоритмом, похожим на EM-алгоритм

Функции компетентности

Функция компетентности может определяться:

- признаком f(x)

$$w_i(x;\alpha,\beta) = \sigma(\alpha f(x) + \beta),\ \alpha,\beta \in \mathbb{R}$$

- направлением $\alpha \in \mathbb{R}^n$:

$$w_i(x;\alpha,\beta) = \sigma(x^T\alpha + \beta),\ \alpha \in \mathbb{R}^n, \beta \in \mathbb{R}$$

- расстоянием до $\alpha \in \mathbb{R}^n$

$$w_i(x;\alpha,\beta) = \exp(-\beta||x-\alpha||^2),\ \alpha \in \mathbb{R}^n, \beta \in \mathbb{R}$$

- Более сложными способами (density prediction)

Алгоритм

Итерационный процесс обучения, аналогичный EM-алгоритму:

1: Начальное приближение функций компетентности w_t
2: **пока** значения $w_t(x_i)$ не стабилизируются
3: **M-шаг:** при фиксированных w_t обучить все b_t
4: **E-шаг:** при фиксированных b_t оценить все w_t

В данном случае мы задаем число алгоритмов T, хотя его можно и подбирать автоматически

Мета-композиции

Мета-композиция — использование одного или нескольких алгоритма в качестве базового уровня для других алгоритмов. Для обучения классификатора верхнего уровня можно использовать:

- только ответы алгоритмов на всей выборке
- ответы алгоритмов на всей выборке и исходные признаки
- только ответы алгоритмов на кросс-валидации
- ответы алгоритмов на кросс-валидации и исходные признаки
- часть выборки и ответы алгоритмов, обученные по другой части
- ...

Примеры мета-композиций

Примеры достаточно успешных мета-композиций:

- двухслойный (и более) случайный лес — TradeShift Text Classification
- логистическая регрессия над ответами алгоритмов — Seizure Prediction Challenge
- vowpal wabbit над ответами различных онлайн алгоритмов — Criteo CTR
- ...

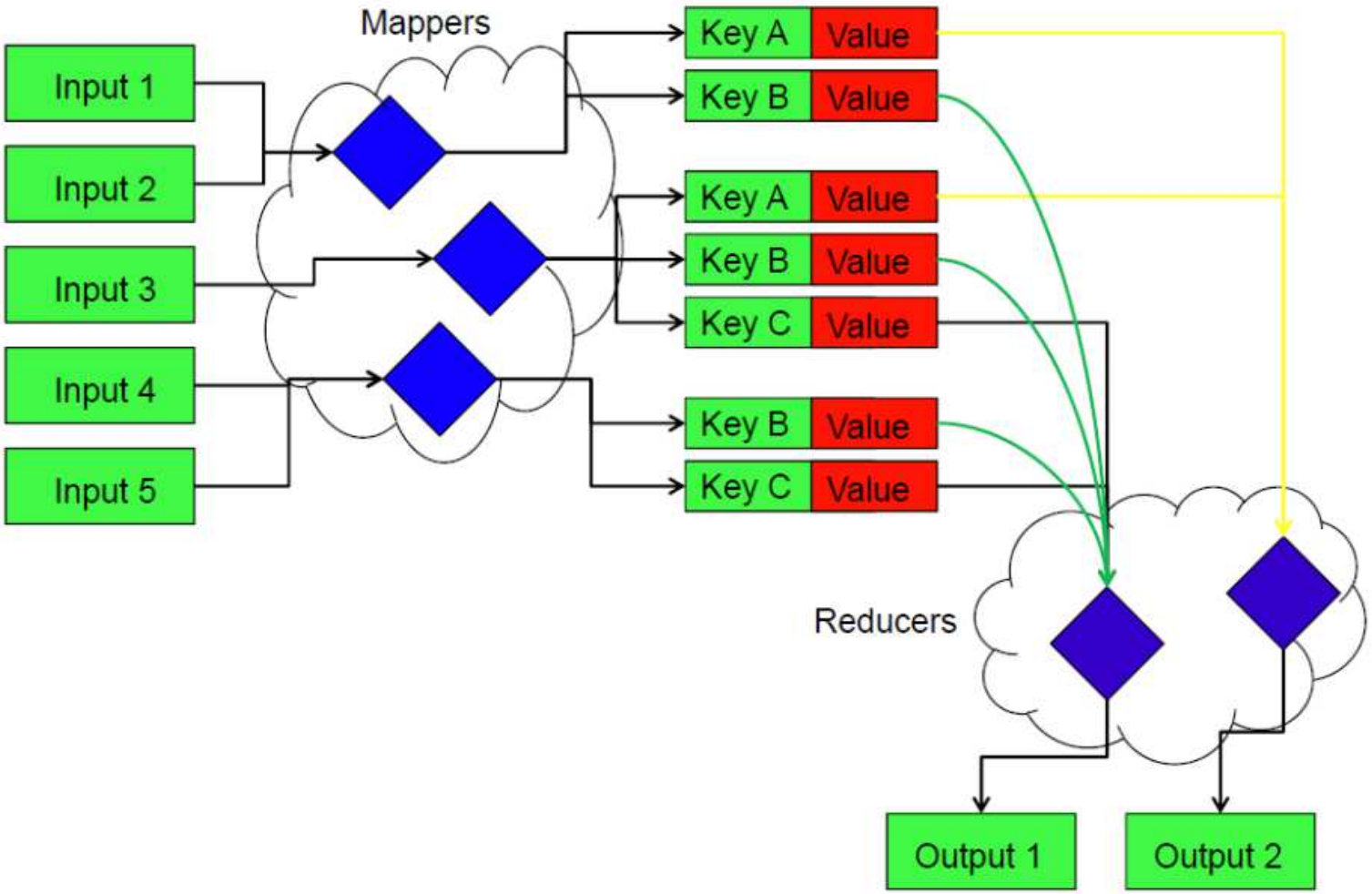
Содержание

Технология PLANET

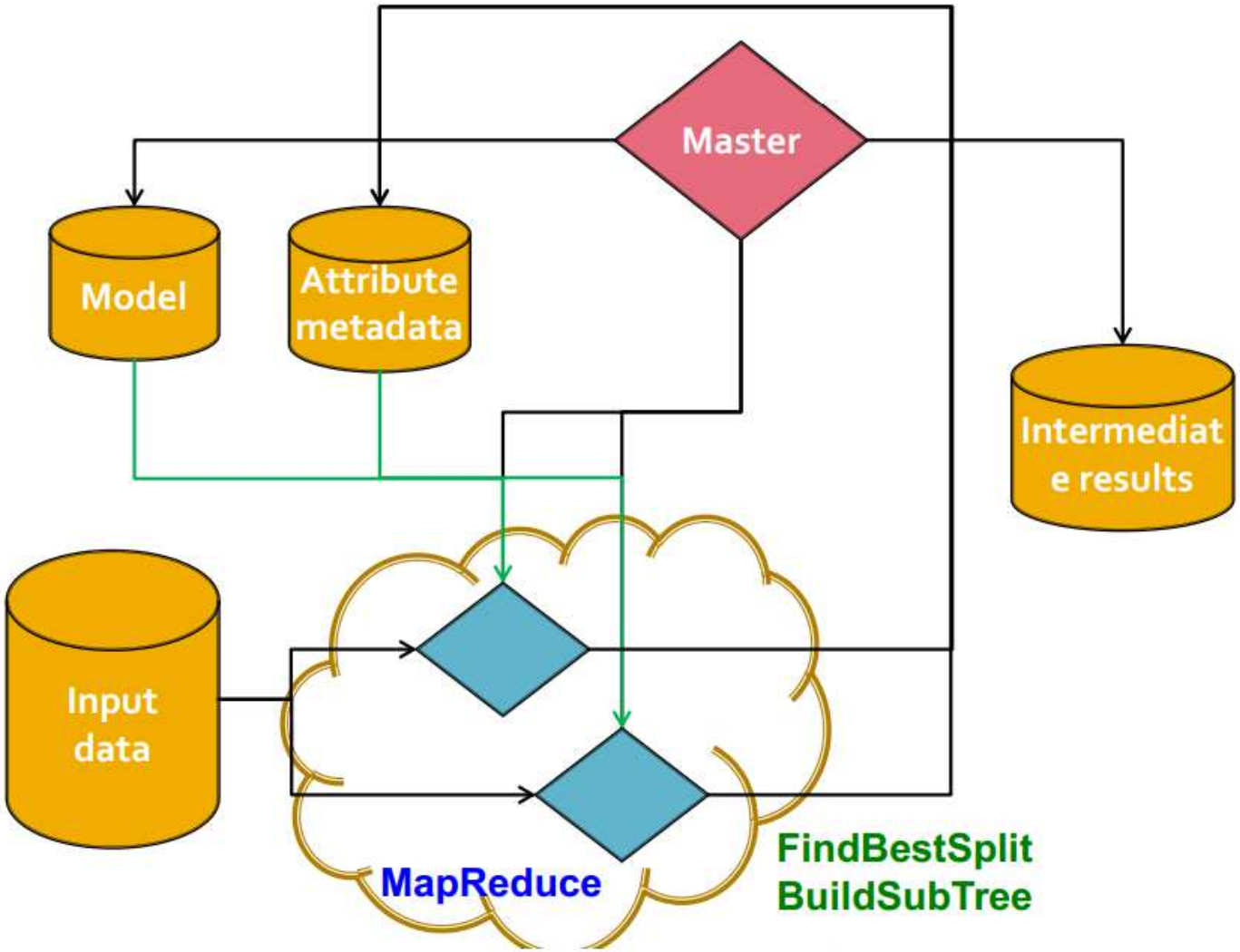
PLANET — Parallel Learner for Assembling Numerous Ensemble Trees — технология построения решающего дерева при помощи нескольких последовательных задач MapReduce.
Ограничения эксперимента:

- Только числовые признаки (вещественные/дискретные)
- Задача регрессии
- Только бнарные разделения: $X_j < v$
- Деревья помещаются в памяти
- Данные не помещаются в памяти

Архитектура обычного MapReduce



Архитектура PLANET



Компоненты

- Строится сразу целый уровень дерева (за один MapReduce)
- Map - сбор статистики по разделению элементов chunk-а для уровня признака V
- Reduce - обработка полученных статистик для выявления лучшего порога V^*
- Master - достраивает новый уровень дерева и сохраняет промежуточные результаты

Компоненты

1. Map - загружает модель и информацию о возможных уровнях разбиения и:
   - видит только часть данных D^*
   - пропускает объект от корня, чтобы найти подходящий лист
   - для каждого листа помнит, кто дошел до него и какие объекты пошли налево/направо
2. Reduce - агрегирует статистику от Map и выявляет лучший порог
3. Master - мониторинг, сохранение результатов и запуск новых MapReduce

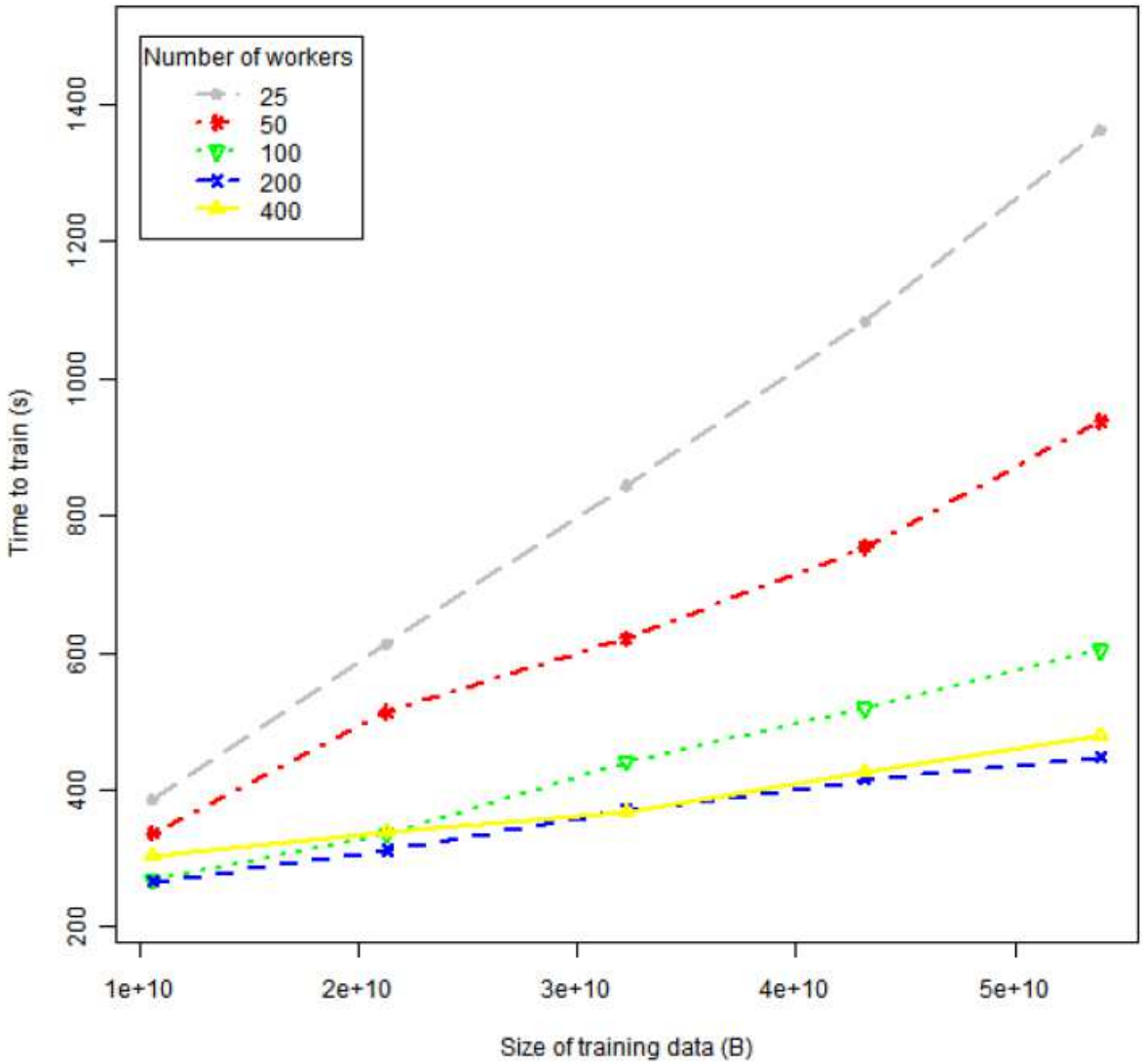
Три типа используемых MapReduce

1. MapReduce Initialization (1 раз) — для каждого признака определяет уровни для разбиения
2. MapReduce FindBestSplit (много раз)
3. MapReduce InMemoryBuild (1 раз) — строит поддерево для данных, которые помещаются в память

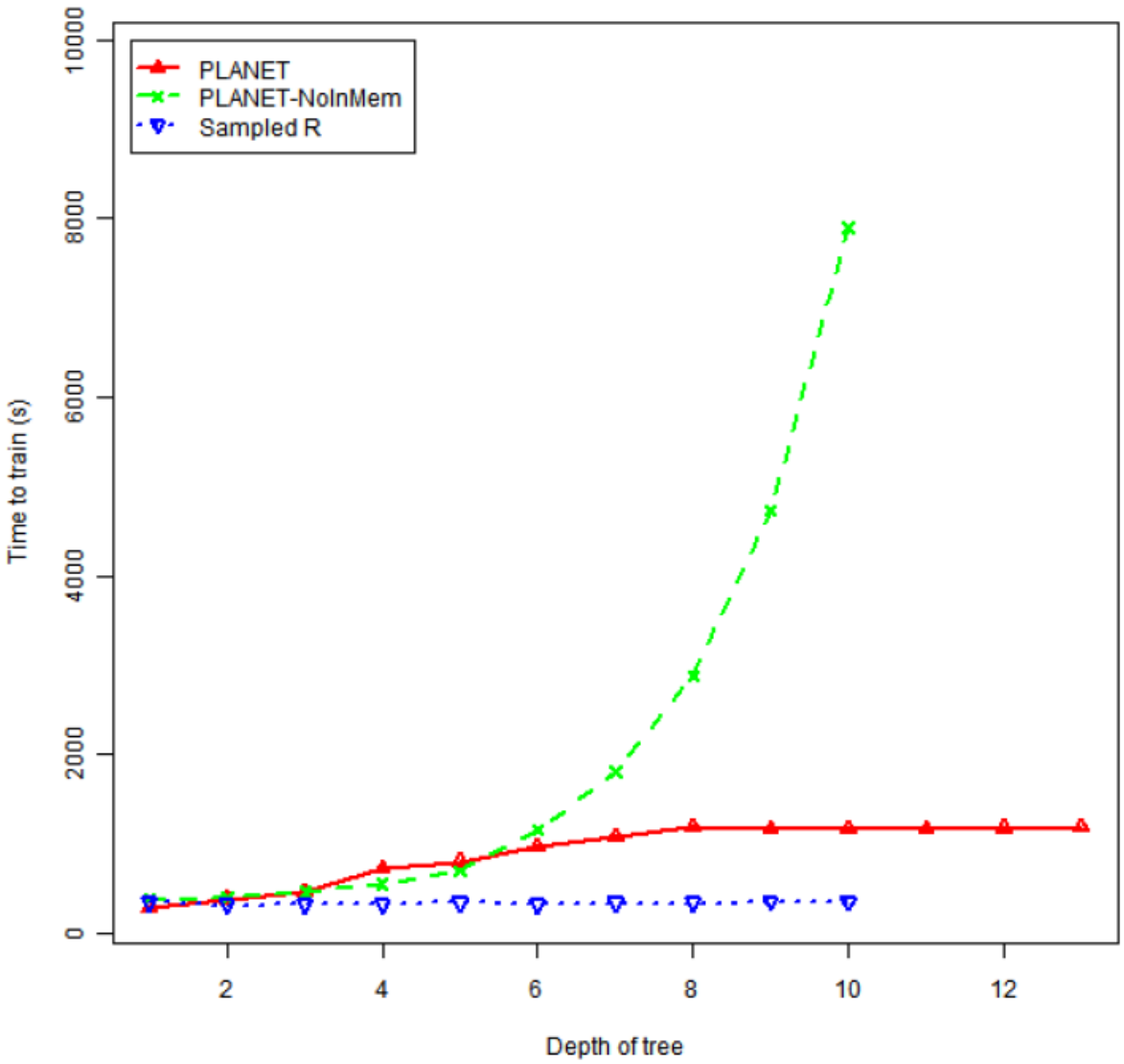
Эксперимент

1. Задача: предсказание Bounce Rate (CTR)
2. Данные: AdCorpus от Google
3. Признаки: 6 категориальных (уровни 1-500) и 4 вещественных
4. Размерность задачи: 314 млн. объектов 64Гб
5. Вычислительный кластер:
   - 400 машин
   - 1ГБ жесткий диск
   - 768Мб оперативной памяти

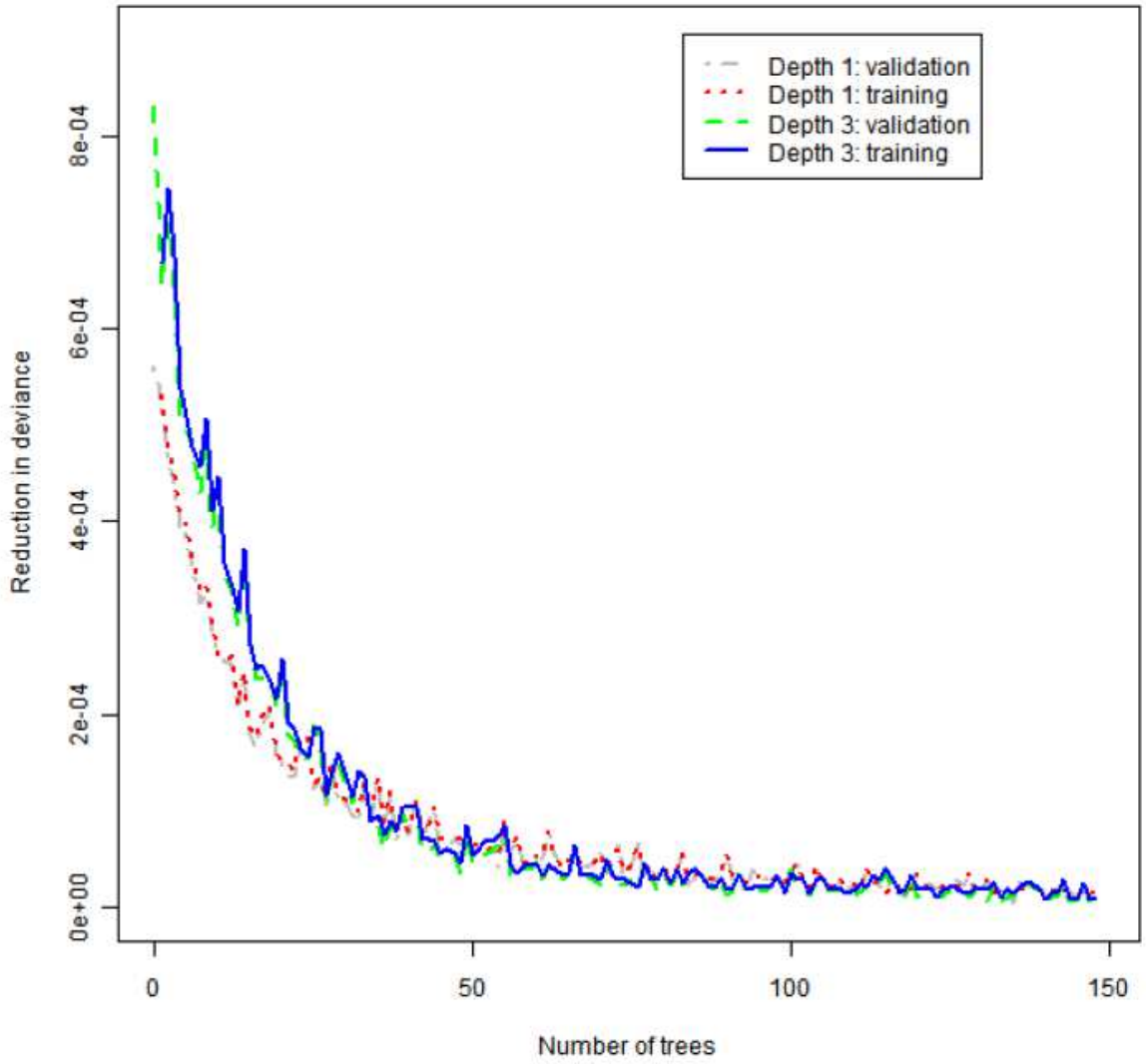
Результаты эксперимента



Результаты эксперимента

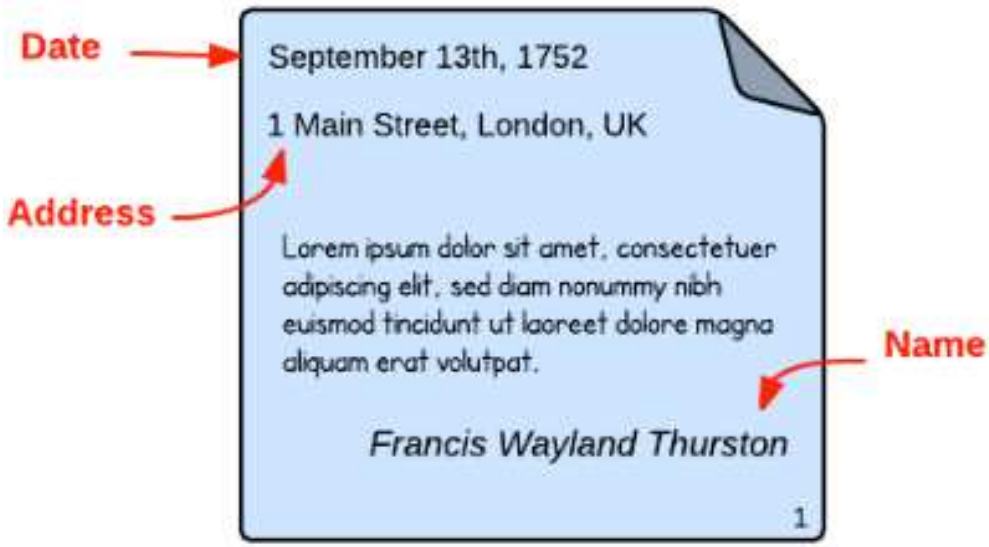


Результаты эксперимента

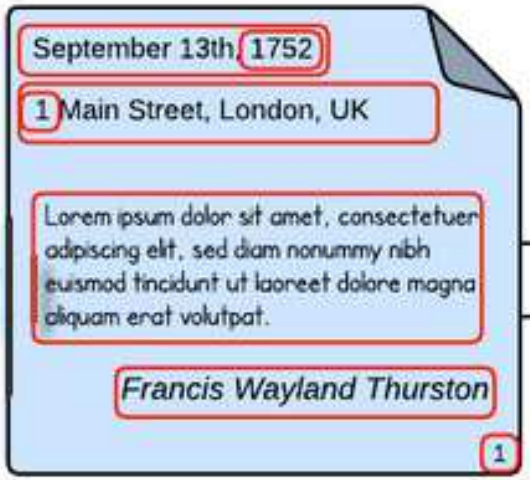


Содержание

TradeShift Text Classification



Данные TradeShift Text Classification



Box nr.	Box text
1	September 13th, 1752
2	1752
3	1
4	1 Main Street, London, UK
5	Lorem ipsum dolor sit amet, consectetuer adipiscing ...
6	Francis Wayland Thurston
7	1

Данные TradeShift Text Classification

- Число объектов (N): 2.1M (80% training, 20% testing)
- Число признаков (M): 145
- Число меток (K): 33

Опытным путем в данных выявлено, что $y33$ соответствует метке «Ничто другое не верно», то есть:

$$y_3 3 = \left[\sum_{i=1}^{32} y_i = 0\right]$$

Данные TradeShift Text Classification

train.csv

id	x1	x2	x3	x4	x5	x6
1	m268i97y	0	NO	105.4	14	
2	j0gheu6	1	YES	25.631	12	
3	26fmap6u	1	NO	12.0	12	m268i97y
4	of64nasl	0	NO	140.12	14	m268i97y
5	13e5dbzp	0	NO	150.92	40	of64nasl
6	8n4t73wy	0	YES	135.01	14	13e5dbzp
7	26fmap6u	1	YES	10.53	10	8n4t73wy

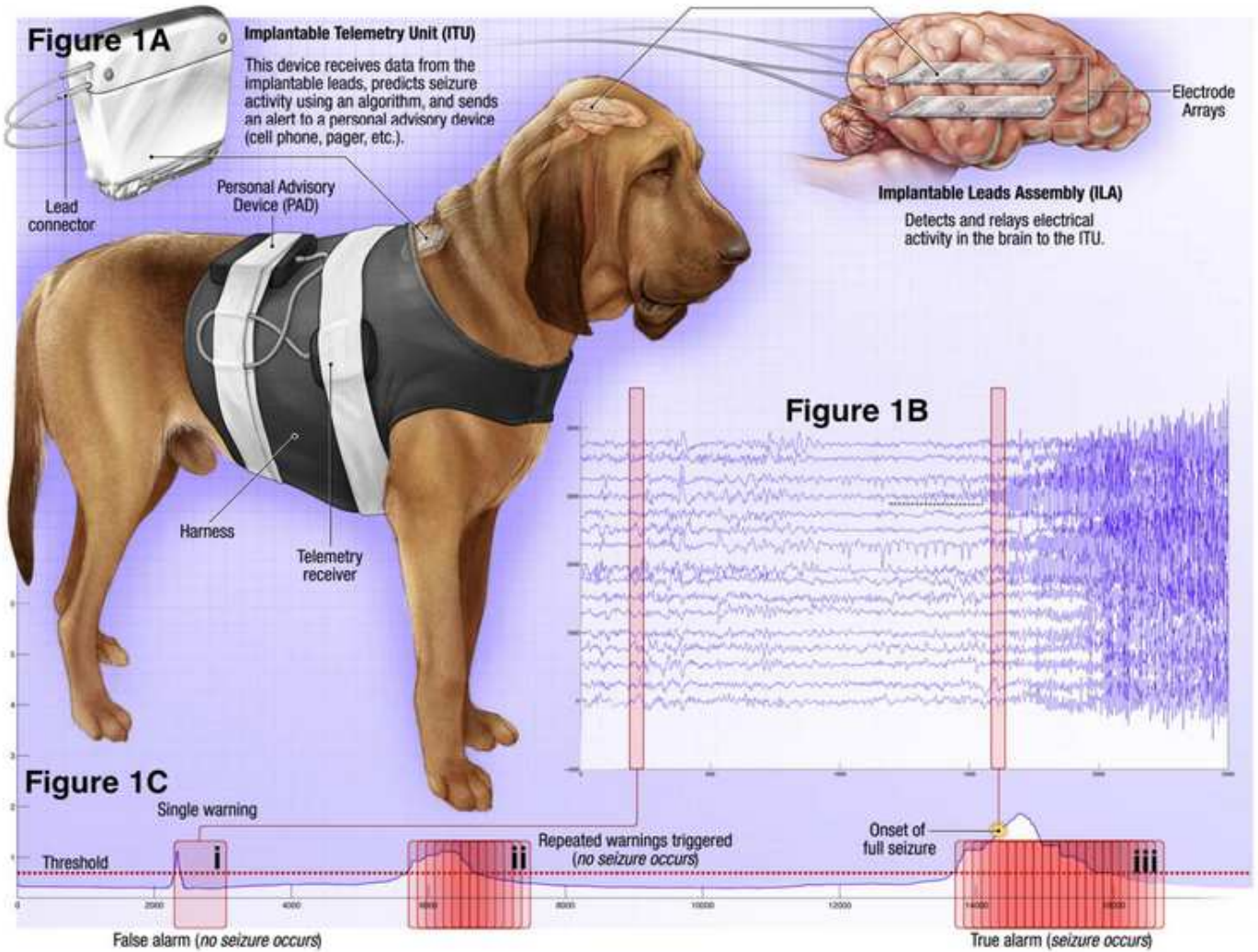
trainLabels.csv

id	y1	y2	y3	y4
1	1	0	0	0
2	0	1	0	1
3	0	0	0	1
4	0	1	0	0
5	0	0	1	1
6	0	0	1	0
7	0	0	0	1

Методы решения TradeShift Text Classification

- Число объектов достаточно велико
- Делим выборку в отношении 60% / 40%
- Обучаем первый слой RF и LinearSVC
- Добавляем полученные ответы для второго уровня к исходным признакам
- Обучаем второй слой RF
- Обучаем супер-мощный алгоритм для $y33$
- Корректируем полученный прогноз
- Submit!

Seizure Prediction Challenge



Данные Seizure Prediction Challenge

- Число объектов: 5 собак + 2 человека
- Число файлов для каждого объекта: от 200 до 2000
- Число каналов для каждого объекта: 15, 16, 24
- Общий объем данных после разархивирования: 105Гб

Возникающие проблемы

1. Алгоритмические ограничения:
   - Разное число каналов для объектов
   - Взаимодействия между каналами
2. Программные ограничения:
   - Запрещено использовать конструкции вида:

```
1: Names = [Dog_1, Dog_2, ..., Patient_2];
2: для каждого объекта obj из Names
3:     ...
```

В этом соревновании официально разрешили подстраивать итоговые алгоритмы по тестовым данным

Методы решения Seizure Prediction Challenge

- Объем данных более чем велик, поэтому его необходимо редуцировать
- Выделяем признаки из данных: abs(fft) и многое другое
- Делим выборку на 10 фолдов
- Генерируем ответы на каждый фолд при помощи различных алгоритмов: RF (2 варианта), ExtraTrees (2 варианта), GBM (3 варианта)
- Обучаем логистическую регрессию на ответах классификаторов базового уровня
- (Корректируем полученный прогноз)
- Submit!

Корректировка прогноза Seizure Prediction Challenge

- Напоминание:

$$logit(p) = log(p) - log(1 - p)$$

$$inv.logit(p) = \frac{e^p}{e^p + 1}$$

- Делаем преобразования ответов, полученных при помощи алгоритмов:

$$inv.logit(0.0 + 1.0 * logit(data[, out]))$$

- Пробуем еще 2 варианта:

$$inv.logit(0.5 + 1.0 * logit(data[, out]))$$

$$inv.logit(-0.5 + 1.0 * logit(data[, out]))$$

- Предполагая, что это парабола, находим точку максимума
- Отправляем в LeaderBoard и корректируем, если нужно

Содержание

Выводы

Композиции алгоритмов позволяют:

- повышать устойчивость итоговых алгоритмов
- преобразовывать признаковое пространство
- выявлять более сложные закономерности в данных
- не переобучаться с увеличением сложности модели (рост обобщающей способности)

Список литературы

- B. Panda, J. S. Herbach, S. Basu, and R. J. Bayardo. PLANET: Massively parallel learning of tree ensembles with MapReduce.
- *http://infolab.stanford.edu/~ullman/mmds/book.pdf*
- *http://www.kaggle.com/c/seizure-prediction*
- *http://www.kaggle.com/c/tradeshift-text-classification*

Спасибо за внимание!!