Подбор гиперпараметров и AutoML

Задачи:

- Определиться с местом и ролью процесса подбора гиперпараметров в рамках задачи построения интеллектуальной системы.

- Определиться с целью и методами подбора гиперпараметров.

- Рассмотреть различные подходы к подбору гиперпараметров.

Задачи DS:

1. Алгоритм подбирает параметры модели
2. DS подбирает алгоритмы и параметры модели

Что делает DS:

1. Алгоритм подбирает параметры модели
2. DS подбирает алгоритмы и параметры модели
3. AutoML подбирает алгоритмы и

AutoML заменит DS?

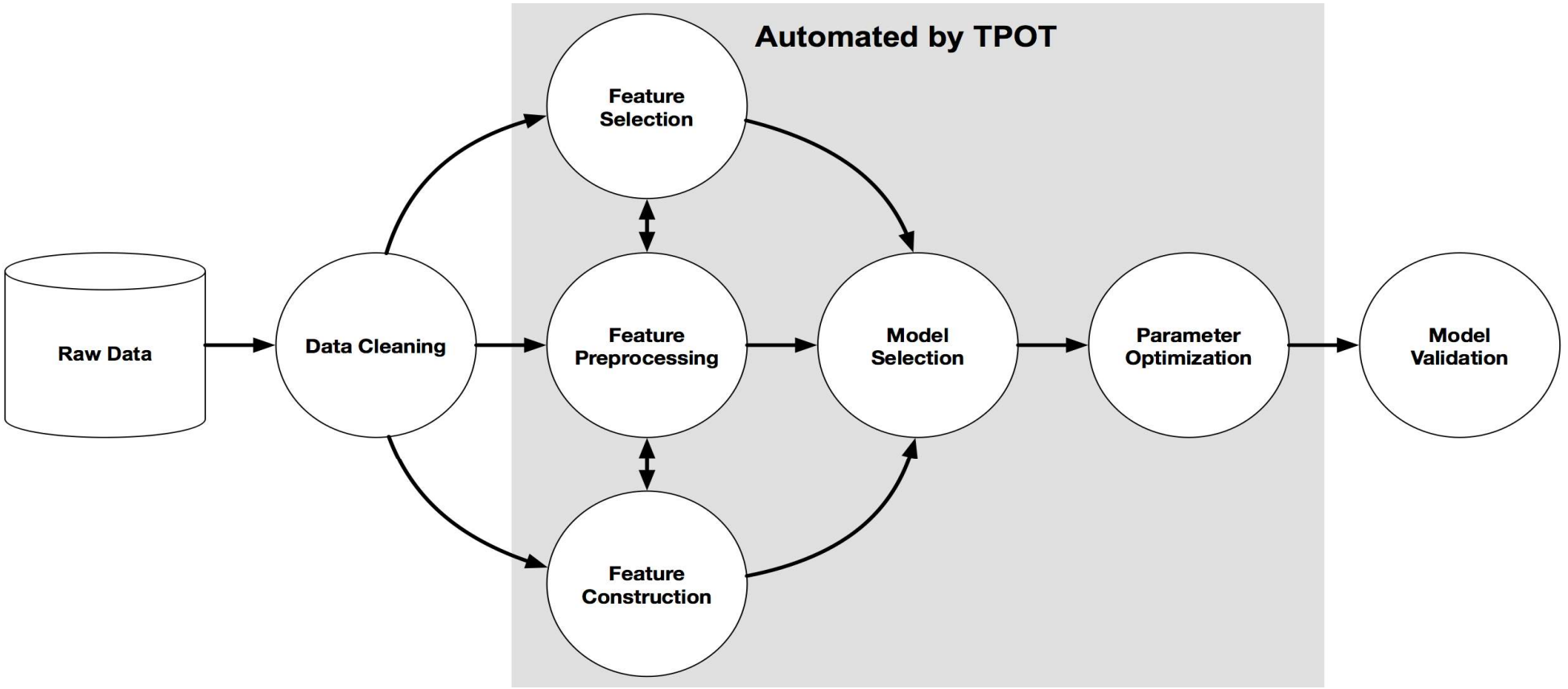
AutoML

- Робот, который полностью заменяет DS-специалиста?
- Инструмент для работы с ML без знания ML?
- Инструменты / подходы, которые автоматизируют или делают более эффективными ключевые процессы в ML?

Инструменты / библиотеки / фреймворки

- H_2O AutoML
- LAMA
- HYPERPORT
- AutoPilot
- AutoKeras
- AutoScikitLearn
- TPOT
- Spark MLLib

Демократизация ML



Data Scientist

AutoML

Data Scientist

AutoML

- Робот, который полностью заменяет DS-специалиста?
- Инструмент для работы с ML без знания ML?
- **Инструменты / подходы, которые автоматизируют или делают более эффективными ключевые процессы в ML**

Возможности AutoML

- Определяем класс
- Выбираем алгоритм
- Готовим данные
- Оптимизируем HP

Параметры и гиперпараметры: сходства, различия, поиск оптимальных значений

Параметры

- **Очень много (миллионы)**
- **Известна зависимость ответа модели от значений параметров**
- **Можно быстро проверить, насколько удачны**
- **?**

Гиперпараметры

- **Мало (десятки)**
- **Как отреагирует результат на изменение, неизвестно**
- **Проверять качество долго**
- **?**

Параметры и гиперпараметры: сходства, различия, поиск оптимальных значений

Параметры

- **Очень много (миллионы)**
- **Известна зависимость ответа модели от значений параметров**
- **Можно быстро проверить, насколько удачны**
- **Сложно подбирать распределенно**

Гиперпараметры

- **Мало (десятки)**
- **Как отреагирует результат на изменение, неизвестно**
- **Проверять качество долго**
- **Поиск распределяется практически равномерно**

Гиперпараметры: проблемы и решения

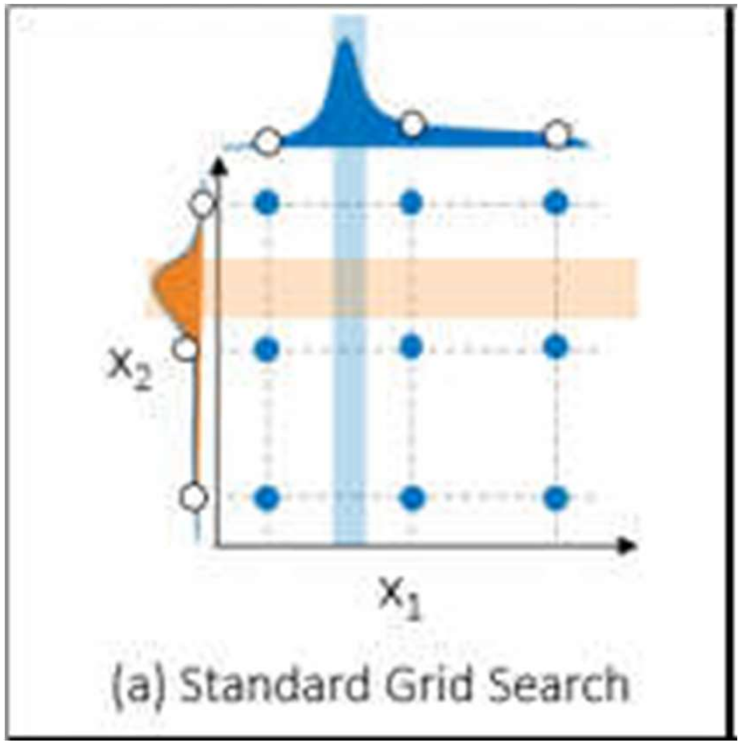
Проблемы:

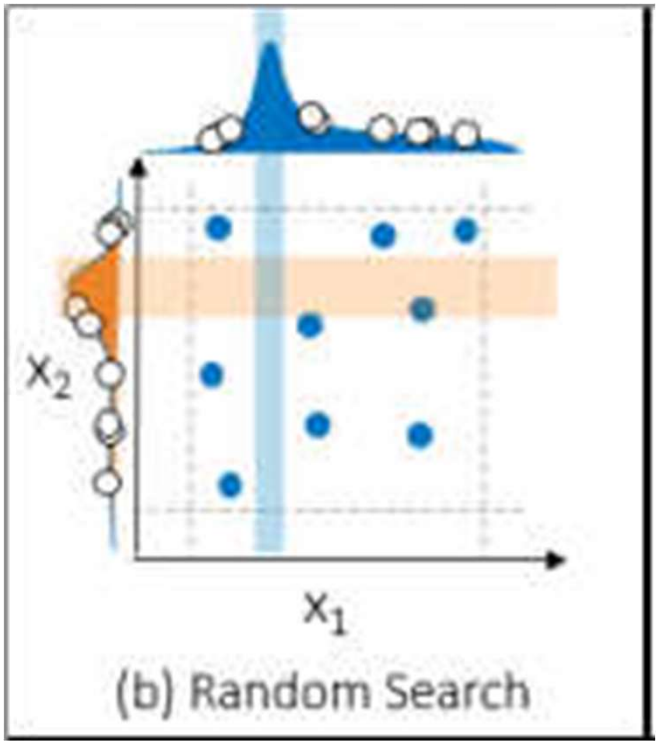
- **Комбинаторный взрыв**
- **Сложность вычислений**

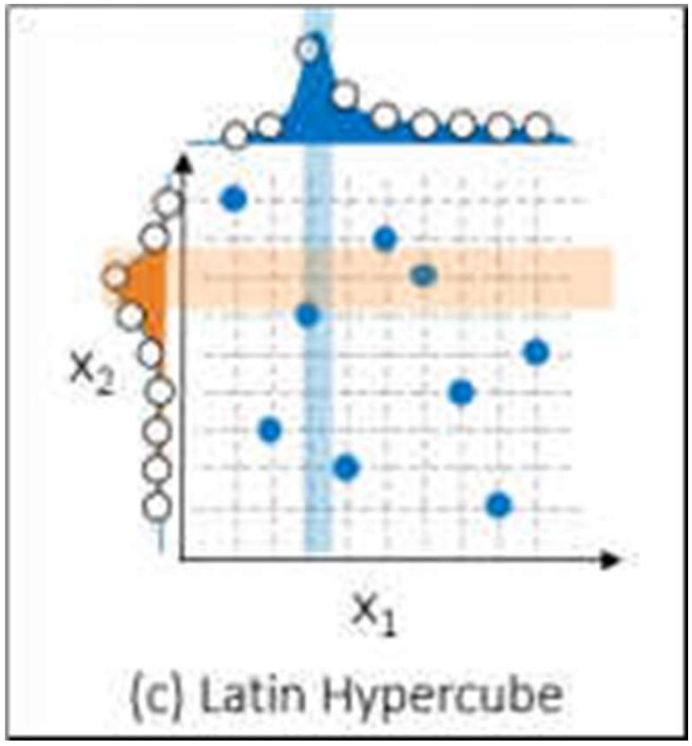
Решения:

- **Выбираем, какие гиперпараметры важны**
- **Выбираем пространство поиска гиперпараметров**
- **Используем распределенные вычисления**

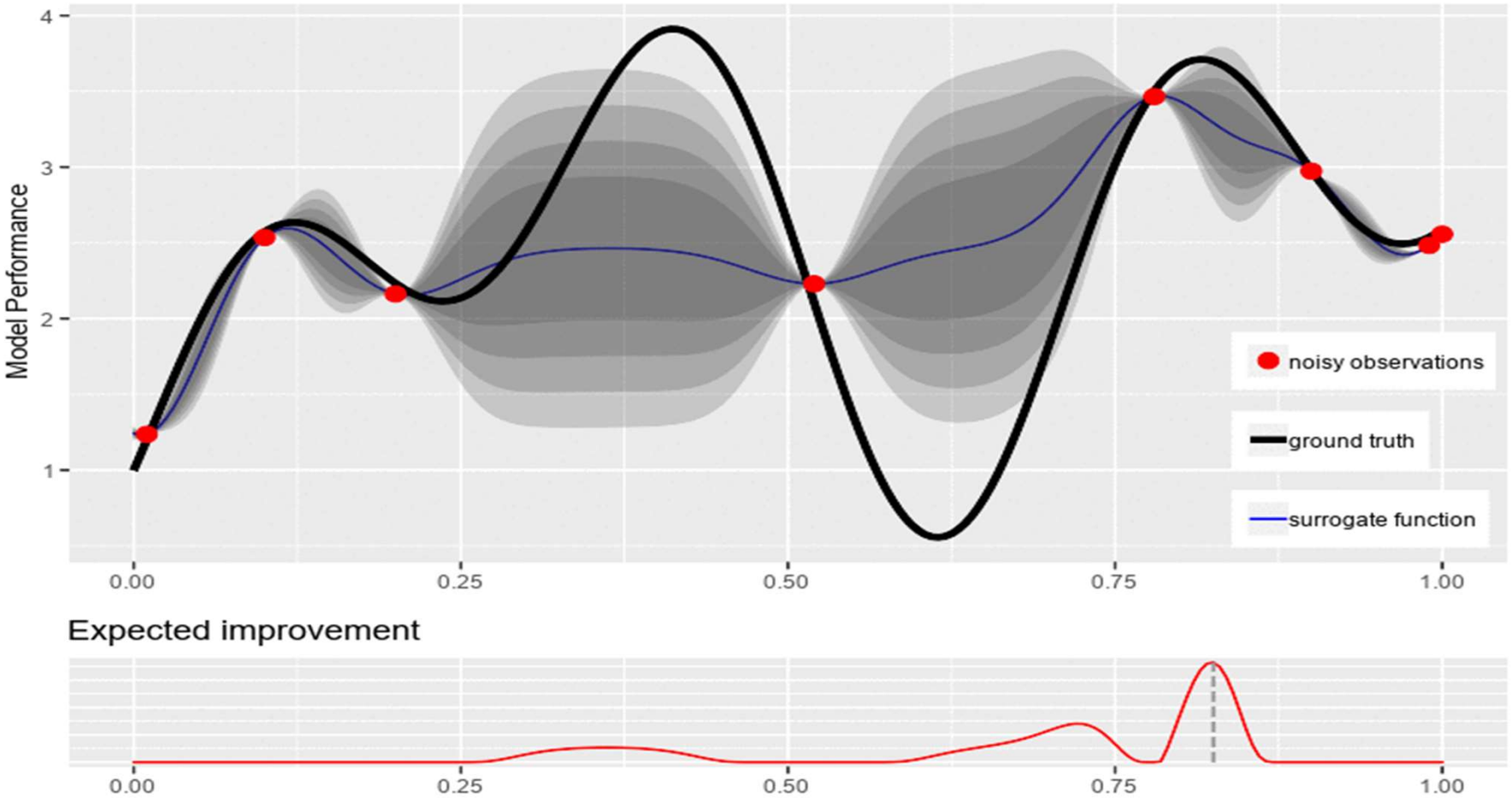
Методы поиска гиперпараметров: Grid Search







Методы поиска гиперпараметров: Biasian Approach



Фреймворки AutoML

1. Scikit Learn и joblib
    - GridSearchCV
    - RandomizedSearch SV
    - joblib
    - joblibspark
2. Spark MLLib
    - Cross Validation
    - Train-Validation Split

Ограничения SparkML

- Один способ подбора гиперпараметров – Grid Search
- Теряется большая часть истории
- Падает в процессе на сложных моделях и приходится начинать с нуля

Hyperopt

Hyperopt - популярная python-библиотека для подбора гиперпараметров. Hyperopt может работать с разными типами гиперпараметров: непрерывными, дискретными, категориальными и т.д, что является важным преимуществом этой библиотеки.

Поддерживает алгоритмы:

- Random Search
- Tree of Parzen Estimators
- Adaptive TPE

Можно распараллелить работу с помощью Spark

Pravda ML

- Вся информация о метриках, весах, параметрах и т.д. сохраняется вместе с моделью в parquet
- Параллельное вычисление блоков с восстановлением после падений
- Выделенные абстракции для тест-трейн разбиения, оценки качества, поиска гиперпараметров
- Гибкое управление параллелизмом
- Прокаченные распределенные ML-алгоритмы

Краткий итог

- Процесс оптимизации гиперпараметров может быть сложным и трудоемким
- Распараллеливание процессов – один из подходов для решения задачи
- Также могут помочь «продвинутые» методы поиска
- Нужно соблюдать баланс между параллелизмом и спец. подходами
- Учитываем условия задачи и бюджет
- Не забываем о логировании результатов

Альтернативные подходы

- Всегда ли нужен параллелизм (на уровне кластера)?
- Что если параллелизм уже заложен во фреймворк (Tensorflow + Horovod, etc.)
- А если еще проще? Hydra Sweeps + MLFlow
- А если сложнее? Из Спарка можно запускать Спарк
- Облачные решения на примере AutoPilot

Big Data и «букет» проблем

- Разные оптимальные конфигурации кластеры на разных этапах
- Высокий уровень параллелизма может приводить к проблемам с очисткой контекста
- При работе с нативным ML под капотом иногда невозможно в одном процессе учить несколько моделей