Crowdsourcing Management

Page • 1 backlink • uni

Crowdsourcing – это отдельный проект

Зачастую стартапы сажают людей на разметку, а уже потом, если проект выстреливает, делают честное решение

Задача: практически никогда в unsupervised модели не обучаются, нужна разметка

Популярные задачи:

- search relevance
- модерация/валидация данных
- sentiment analysis
- транскрибирование (дополнить медиаданные полезными метаданными)

Улучшаем качество модели *и получаем полезные признаки*

В краудсорсинге есть понятия задачи, пререквизитов для ее выполнения, платформа, работник (асессор) – иногда ML-инженер должен сам становиться асессором

Процесс:

- заказчики определяют задание и делают формочку
- поставка данных из дата пайплайна (см. предыдущее занятие)
- требуемое качество, требования по времени, **стоимость задания** (опирается на первые два)
- публикация задания, оплата и мониторинг скорости и качества

Ценообразование бывает динамическим (в зависимости от количества разметчиков и он срочности проекта)

Перекрытие – даем одно задание по несколько раз (есть динамическое перекрытие)

На сходимость всех крауд проектов влияют:

- затраты на разметку (чем больше денег забираем, тем выше шанс, что нас закроют)
- скорость разметки

- качество разметки

Далеко не всегда асессор делает ошибку специально, это нужно брать в расчет: нужно отличать непредвиденные ошибки от читерства

- тесты на квалификацию
- подмешивание голден сета в данные для разметки – размечаем сами или с помощью экспертов

Модели перекрытия:

- majority vote – легко интерпретировать и оценивать стоимость всей разметки, минус – не оцениваем уровень качества асессоров
- Dawid-Skene
- Whitehill
- многие другие

Unified Framework in Existing Works – EM Algorithm: имея оценки качества асессоров, оцениваем истину для каждого задания, основываясь на которой меняем оценки качества асессоров, пока оценки качества не сойдутся

Нужно разработать вероятностную модель: учесть тип задания, затем модель асессора, objective function для оптимизации

Dawid-Skene – модель, основывающаяся на confusion matrix (ground truth vs. assessor prediction для каждого асессора): EM-алгоритм между вероятностями получить оценку при условии ground truth и значениями ground truth

Item response theory: моделирование навыка асессора (считаем, что владение каждым навыком распределено нормально), но опять используем правильные ответы (если их нет, берем EM-алгоритм) – Whitehill

Прочие алгоритмы: GLA, CARD, Minimax и так далее

Cost control

Слагаемые стоимости проекта: зарплаты, электричество и крауд

Техники:

- task pruning
- вывод ответа на основе других (нужно условие транзитивности, имеем граф ответов с **вероятностными** ребрами) – подходят для разных типов хадачи, но не везде можно использовать + усиливается эффект систематических ошибок

- выбор задачи – какие задачи выгоднее всего отдать в крауд (получаем quality/cost tradeoff, но можем проиграть с задержкой выполнения)
    - active learning
    - разметка top-k (актуально в поиске, но нельзя забывать про хвост)
- сэмплирование – генерируем задачи из сэмпла, а на общие данные делаем экстраполяцию (контролируем вероятность правильного ответа, доверительные интервалы, а это не всегда применимо)
- частичная оплата труда (самым качественным сотрудникам) – законодательство может не позволить

Task design – пример с трейдоффов по переработке и потере фокуса

Latency control (определяется как время до завершенного последнего задания):

- recruitment time – для уменьшения этого и помогают крауд-платформы
- qualification + training time
- work time
- quality control time

Single-Batch Latency Control – контроль задержки "мгновенных" заданий

Контроль: раскидываем между несколькими асессорами, кто первый среагировал, тот и размечает

Creating Training Datasets

На практике данные не соотносятся с распределениями (собираются, откуда получается и в каких количествах получается), в выборку вносятся смещения + недостатки данных на определенных срезах + все равно есть некачественные ответы – с этим нужно уметь жить

Shuffle then split – помним про стратификацию, помимо этого помним про время, географию и определение этих двух вещей на домене задачи

IID (н.о.р.) в проде не бывает

В аудио зацикливание в 1% случаев – критическая проблема, но любой SBS покажет, что мы все равно выигрываем

Сколько данных достаточно? Можно построить модель зависимости качества от количества, но интуиция все равно понадобится (тем более, не всегда добирать данные хорошо, лучше разметить меньше, но качественнее)

Пример с именованными сущностями: неспециализированный асессор может выдать несколько означений, имеющих право на существование

Другой способ: авторазметка (ручная дорогая, имеет риски протечки, медленная, неадаптивная, но ограничены качеством – автоматическая решает все эти проблемы), можем обучить модель на разметке специалистов и оценивать ее качество специалистами (часть разметки заменять кодовыми эвристиками – task pruning)

Dawid-Skene, Whitehill и так далее также применяются к labeling function