

CROWD-SOURCED AI

Ярослав Чернышев

Научный руководитель:

Daniel Kudenko

21 февраля 2018 г.

Мотивация проекта

Последние разработки в области игрового искусственного интеллекта (e.g. AlphaGo^[1]) используют сочетание алгоритмов машинного обучения и анализа экспертных действий профессиональных игроков.

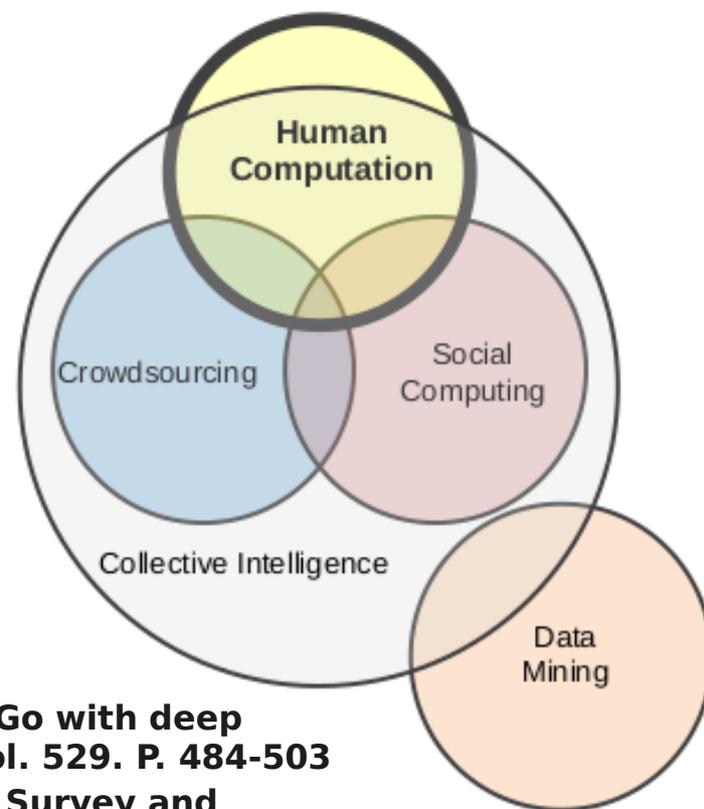
***The current state of human computation has a tendency to represent workers as faceless computational resources*^[2].**

1. D.Silver, A.Huang, C.Maddison and others. Mastering the game of Go with deep neural networks and tree search // Google DeepMind Nature. 2016 Vol. 529. P. 484-503

1. Alexander J. Quinn, Benjamin B. Bederson. Human Computation: A Survey and Taxonomy of a Growing Field // CHI 2011, May 7-12, 2011, Vancouver, BC, Canada



AlphaGo



Постановка цели

Глобальная цель: проанализировать применимость подхода *human computation for solving computational problems* на примере игры *Dota 2*



Задачи на семестр:

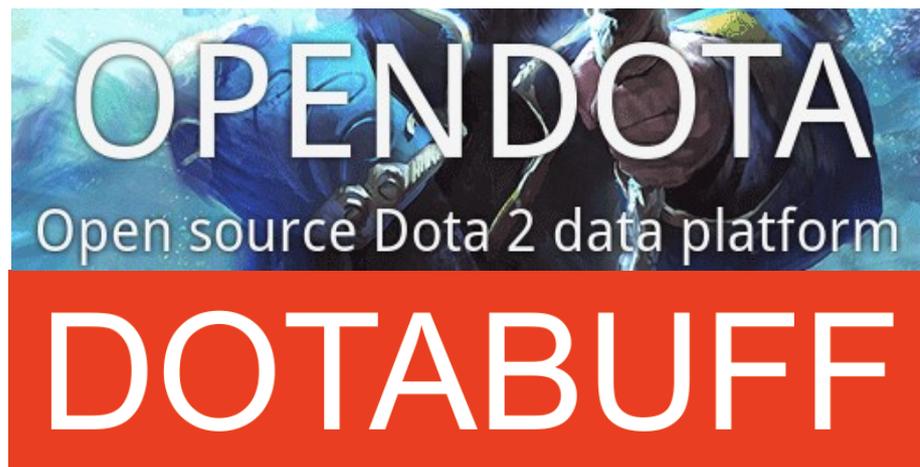
- ▶ **Data Mining** - найти способ загружать большое количество (тысячи) записей игр с серверов Valve.
- ▶ **Data Mining** - Извлекать данные о действиях игроков из записей игр, формировать дата-сет данных по каждому матчу (вектора состояния во времени)
- ▶ **Crowdsourcing** - применить к полученным данным алгоритмы машинного обучения для аналитики и выявления первичных метастратегий игры.

Имеющиеся решения в области



- ✗ Для data-mining привлекли opendota.com, которым пожертвовали \$12000^[1]
- ✗ Проанализировано 5.8 миллионов записей игр.
- ✗ Proprietary project команды Илона Маска

1. <https://blog.openai.com/more-on-dota-2/>



- ✗ Предоставляют игровую статистику по отдельным матчам (в основном о результатах матча)
- ✗ Имеют значительные баги API
- ✗ Непригодны для аналитики

Результаты - Data Mining

I. Steam WebAPI закрыл доступ к инструментам, используемым для фильтрации матчей

II. Для формирования прямой ссылки на загрузку реплея необходимо использовать клиент Steam или сервис opendota.com - в обоих случаях нет возможности автоматизации

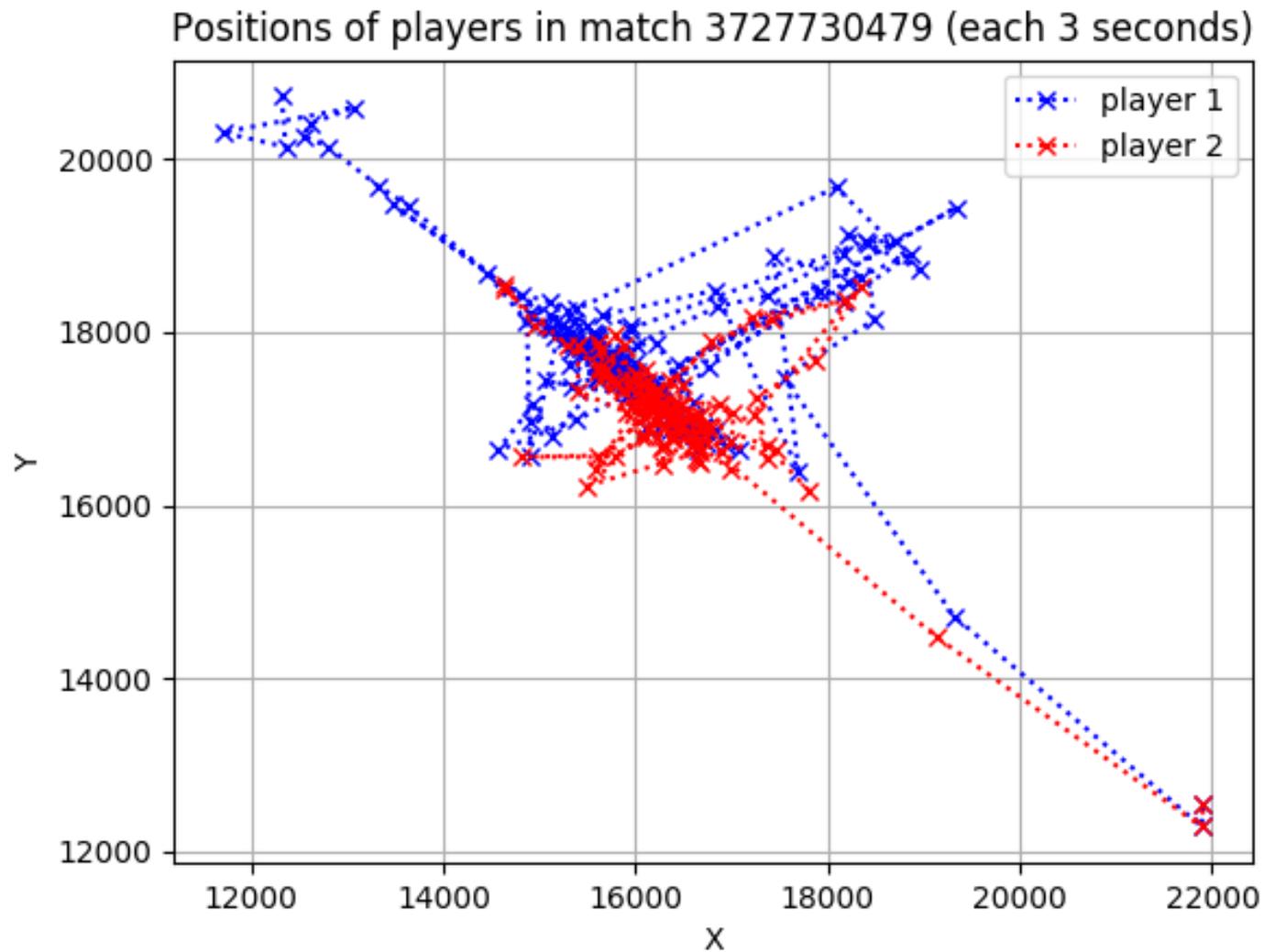
III. Имеем большое количество игровой информации, упакованной в бинарники. Как извлечь и правильно сформировать дата-сет?

I. Используем Python для скраппинга данных с сайта dotabuff.com

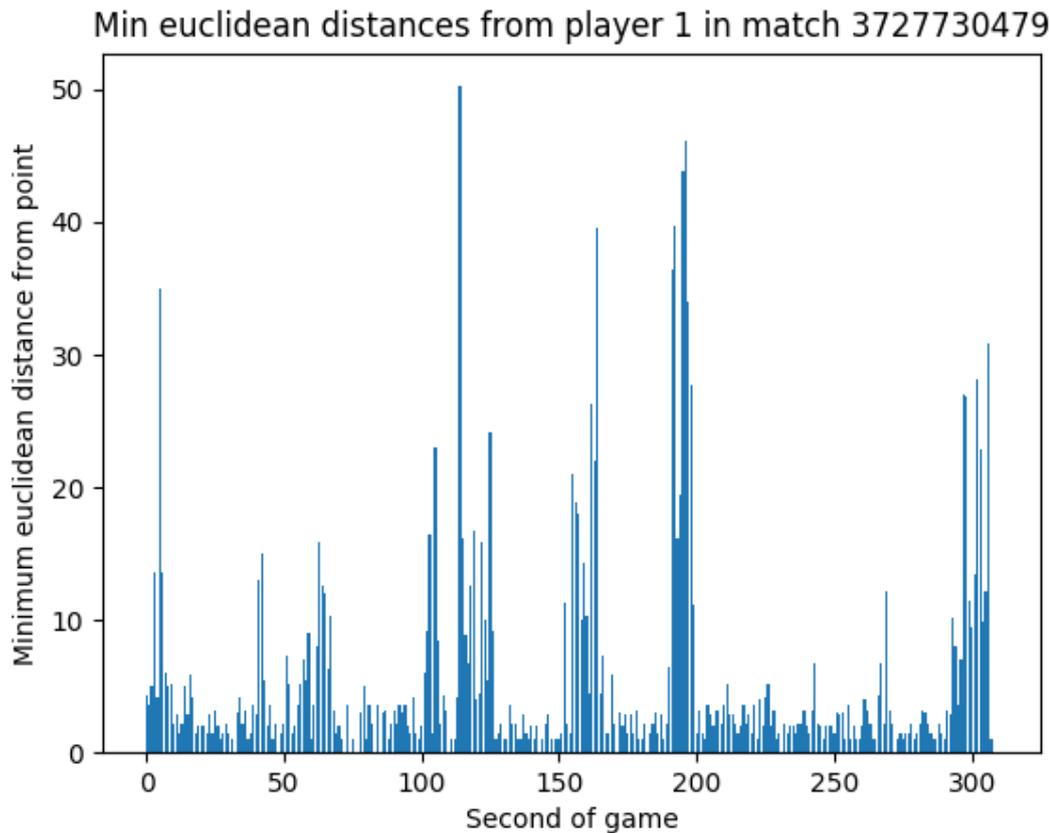
II. Эмулируем клиент Steam с помощью библиотеки `node-dota2` на JavaScript. Ограничение 100 обращений к серверу на 1 аккаунт в сутки и 500 на 1 IP-адрес (можно обойти мультиаккаунтингом и прокси-серверами).

III. Для извлечения данных используем open-source парсер Clarity (написан на Java, с которой я никогда не работал) Экспериментируем с Clarity, получаем логи и с помощью NumPy и SciPy анализируем.

Результаты - Data Mining



Результаты работы - Crowdsourcing



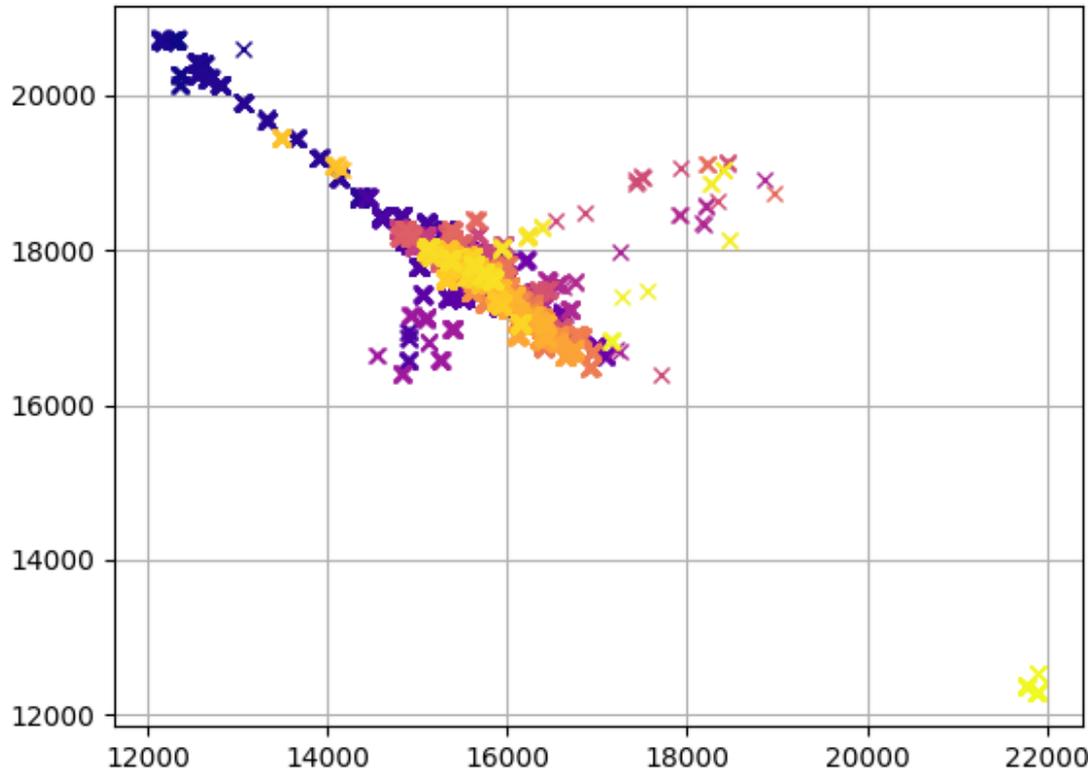
Вычисляем евклидово расстояние по вектору состояния в текущий момент времени по базе из 1000 реплеев.

Использование:

Получили вектор → нашли наиболее похожий по базе → выполнили действие, соответствующее найденному вектору

Результаты работы - Crowdsourcing

kNN clusterisation of player 1 coordinates in match 3727730479



Считать евклидовы расстояния - дорого и долго. Можно хранить предсчет, а можно кластеризовать.

Построили для базы из 1000 реплеев k-мерное дерево координат в момент времени.

Для вектора состояния каждого игрока во времени с помощью данного дерева можем провести кластеризацию по k ближайшим соседям.

Дальнейшее развитие

- 1) Прикрутить СУБД для хранения результатов вычисления**
- 2) Расширять еще кучей характеристик (e.g. hp, gold, вектор противника и крипов)**
- 3) по вектору находить соответствующее состоянию действие через Евклидово расстояние или kNN**
- 4) заставить этого Франкенштейна играть.**

