CDT Summer School

- Reorganization (10 min)
  - No email/surfing web/texting
  - Find Github
    - Sit with breakout group

- Designing Oralytics: RL for Real Life
  - Lecture 45 min
  - Breakout in groups + Discussion (20 min)

1

https://github.com/StatisticalReinforcementLearningLab/Stat-ML-CDT-2023/tree/main

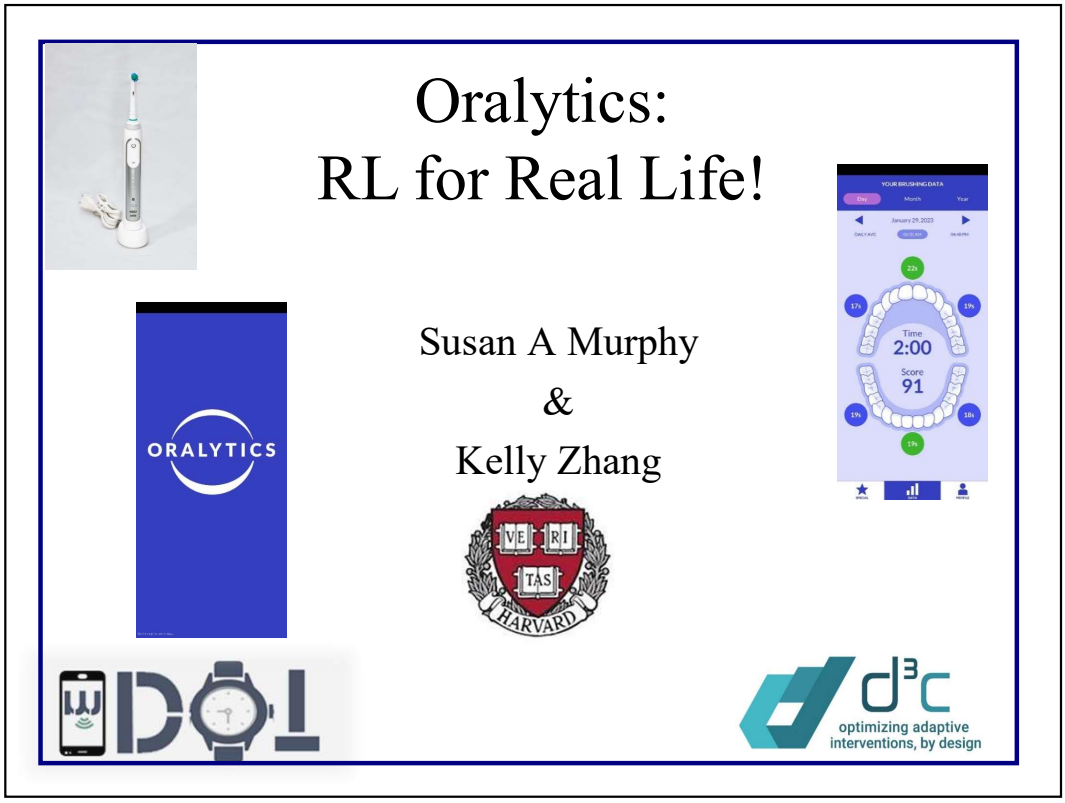
Oralytics; RL for Real Life 3 Hours, 15 min. 09.15 to 12.30 Tues morning

CDT Summer School

- Theory for data analyses when the RL algorithm pools data across users in order to select actions
  - Lecture 35 min
  - Breakout in groups + Discussion (20 min)
  - Lecture & Coding 35 min
- Breakout in groups + Discussion (30 min)

2

https://github.com/StatisticalReinforcementLearningLab/Stat-ML-CDT-2023/tree/main/precourse



Abstract:
Dental disease continues to be one of the most prevalent chronic diseases in the United States. Recent advances in digital technology, electric toothbrushes and smartphones, offer much potential for promoting quality tooth-brushing in real-time, real-world settings. Sensor data from electronic toothbrushes and smartphones provide data & matching mobile apps deliver on-demand feedback and educational information. Behavioral scientists have developed a variety of prompts to encourage engagement in the app and the associated toothbrushing behaviors. Here we describe the multiple stages of development of an online reinforcement learning algorithm for learning which type of prompt/ no prompt, in which state, is most effective in promoting quality tooth-brushing.

To Think About

- Where might you trade bias with variance as you design an online RL algorithm?
- What are delayed effects?
- What are some of the challenges in developing an RL algorithm for clinical trials?

4

Outline

- RL Re-Cap
- Oralytics & Clinical Trials
- Rocky Road to Clinical Trial
  - Testbed Development
  - RL Variants
- Data Collection for Causal Inference & to Assess Performance

5

RL Re-Cap Data

- Data at time t : $\{S_t, A_t, R_{t+1}\}$
- History at time t: $H_{t-1} = \{S_j, A_j, R_{j+1}\}_{j=1}^{t-1}$

- t: time
- S_t: State accrued after $t-1$ and up to/including time t
- A_t: Randomized treatment/action at t^{th} time
- R_{t+1}: Reward (e.g., utility, cost) accrued after time t and up to time $t+1$

$$E[R_{t+1}|S_t, A_t] = r(S_t, A_t)$$

Online RL Algorithm

We take a "regression" perspective to constructing an online decision-making algorithm, $\pi^{\mathcal{L}} = \{\pi_t^{\mathcal{L}}\}_{t \geq 1}$

- Two Elements
  - Learning Algorithm: An algorithm that estimates parameters in a model for (parts of) $\{S_j, A_j, R_{j+1}\}_{j=1}^{t-1}$
  - Optimization Algorithm: Uses output of learning algorithm to construct $\pi_t^{\mathcal{L}}$ and select A_t

7

In our class we focus on the "regression" perspective.

The other common perspective is a classification perspective (e.g. policy search)

Many methods combine the two perspectives. Actor-Critic algorithms do this.

Example
Simple Contextual Bandit

Learning Algorithm: An algorithm that estimates parameters in a model for (parts of) $\{S_j, A_j, R_{j+1}\}_{j=1}^{t-1}$

- Algorithm incrementally estimates parameters in a model for $r(s,a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$
- R_{t+1} is the time t regression target!
- What is the regression target in MDP setting?!

8

Regression examples:

Regression model for $r(s,a)$

Regression model for $Q(s,a)$

Simple Contextual Bandit

Optimization Algorithm: Uses output of learning algorithm to construct $\pi_t^{\mathcal{L}}$ and select A_t

– Uses estimators, $\hat{r}(s,1), \hat{r}(s,0)$, and some measure of confidence (estimated standard error, length of estimated confidence interval, posterior variance) to construct $\pi_t^{\mathcal{L}}$

9

What was the optimization algorithm in Raaz's Monday afternoon lecture?!

Re-Cap: Online RL Algorithm

- For $t = 1$ to T do:
  - Algorithm receives state S_t
  - Algorithm selects action A_t -- using $\pi_t^{\mathcal{L}}(\cdot\,|S_t, H_{t-1})$
  - Algorithm receives reward R_{t+1}
  - If t is an update time do:
    - Update model; update $\pi_t^{\mathcal{L}} \rightarrow \pi_{t+1}^{\mathcal{L}}$
  - End If
- End For

10

A statistician might write $\hat{\pi}_t^{\mathcal{L}}(\cdot\,|S_t)$ instead of $\pi_t^{\mathcal{L}}(\cdot\,|S_t, H_{t-1})$ Why?

OR

A statistician might write $\pi_t^{\mathcal{L}}(\cdot\,|S_t;\, \hat{\beta}_t)$ instead of $\pi_t^{\mathcal{L}}(\cdot\,|S_t, H_{t-1})$ Why?

Re-Cap: Online RL Algorithm

- For $t = 1$ to T do:
  - Receive state S_t
  - Algorithm selects action A_t -- using $\pi_t^{\mathcal{L}}(\cdot\,|S_t, H_{t-1})$
  - Receive reward R_{t+1}
  - If t is an update time do:
    - Update model; update $\pi_t^{\mathcal{L}} \rightarrow \pi_{t+1}^{\mathcal{L}}$
  - End If
- End For

Learning Algorithm

11

In our case the algorithm should be an "anytime" algorithm; the algorithm should not use knowledge of T

Re-Cap: Online RL Algorithm

- For $t = 1$ to T do:
  - Receive state S_t
  - Algorithm selects action A_t -- using $\pi_t^{\mathcal{L}}(\cdot\,|S_t, H_{t-1})$
  - Receive reward R_{t+1}
  - If t is an update time do:
    - Update model; update $\pi_t^{\mathcal{L}} \rightarrow \pi_{t+1}^{\mathcal{L}}$
  - End If
- End For

Optimization Algorithm

12

In our case the algorithm should be an "anytime" algorithm; the algorithm should not use knowledge of T

Outline

- RL Re-Cap
- Oralytics & Clinical Trials
- Rocky Road to Clinical Trial
  - Testbed Development
  - RL Variants
- Data Collection for Causal Inference & to Assess Performance

13

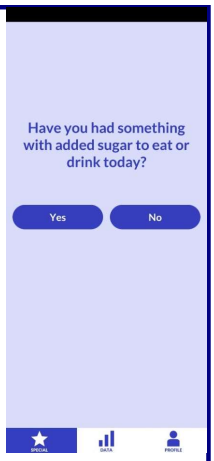
Papers

RL development for pilot study and using PCS principles: A.Trella, K. Zhang, I. Nahum-Shani, V. Shetty, F. Doshi-Velez, S. Murphy Designing Reinforcement Learning Algorithms for Digital Interventions: Pre-implementation Guidelines. Algorithms 2022, 15(8), 255; https://doi.org/10.3390/a15080255 (special issue on Algorithms in Decision Support Systems). NIHMSID: NIHMS 1825651. PMC9881427

Reward design paper: A. Trella, K. Zhang, I. Nahum-Shani, V. Shetty, F. Doshi-Velez, and S. Murphy Reward Design For An Online Reinforcement Learning Algorithm Supporting Oral Self-Care. *Accepted at IAAI 2023*

Oralytics

Goal: Develop an autonomous oral health coaching intervention for individuals at high risk of dental disease.



Oralytics V1, 10-week trial in Aug 2023 involving $n \approx 70$ users:

- App on smartphone + Bluetooth enabled toothbrush
- "Micro-randomization" via a reinforcement learning (RL) algorithm

14

We are currently in the middle of the second beta test. (following an initial beta test +pilot with 9 participants)

We're recruiting dental patients from UCLA, APLA, and Venice dental clinics, by sending recruitment SMS texts to patients via UCLA Qualtrics. All SMS texts will include an IRB-approved message along with a link to our recruitment website, where they can read more about the study and/or apply.

All dental patients will be 18+ years in age and have at least 20 teeth according to their perio chart data. Because all clinics are in West LA, this will be the general geographical area. We will then be ensuring through screening surveys that they have a home WiFi network, before confirming that they are eligible.

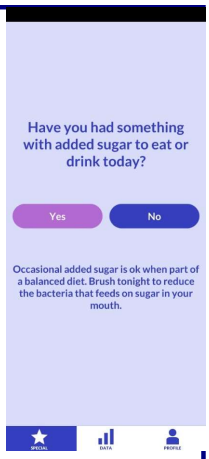
Oral B 8000 is toothbrush

Will take around 2 years to complete.

Oralytics

Oralytics V1, 10-week trial in Aug 2023 involving $n \approx 70$ users:

- App on smartphone + Bluetooth enabled toothbrush
- "Micro-randomization" via a reinforcement learning (RL) algorithm



Oralytics V2, 20-week, randomized-control trial (RCT) involving $n \approx 260$ users, early 2025

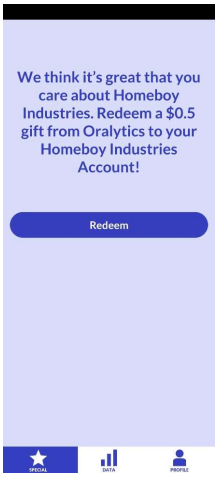
15

Both within study learning and between study learning
Need to be able to transfer to a potentially different population….

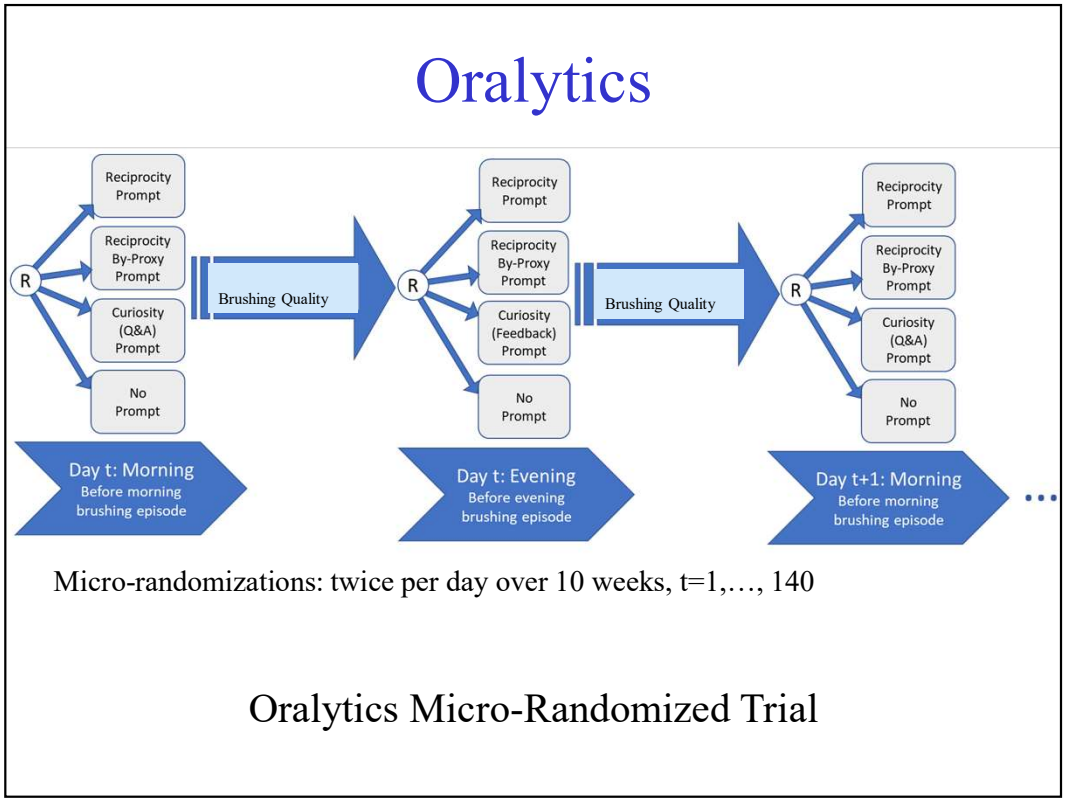
V2 will involve two groups. One with RL and one without RL…

Oralytics

- Decision Times, t: 2 time points per day (prior to the individual's usual brushing time)
- State, S_t: app engagement, prior brushing quality, time of day, weekend, prior # messages,…
- Action, A_t: An engagement message (deliver or not deliver)
- Reward, R_{t+1}: brushing quality score



Brushing Quality =min(180, brushing duration-over pressure)
The RL alg will use a surrogate reward –this afternoon talk.
ONLY 140 decision times per user 70 day study



Micro-randomization—aka RL policy, $\pi_t^{\mathcal{L}}$, is stochastic

These are clinical trials conducted for the purpose of optimizing a complex intervention prior to evaluation in an RCT

The morning engagement strategies include:

1. Standard reciprocity prompt: delivering non-contingent reward points as a "gift" from Oralytics to support goal achievement
2. Reciprocity by proxy prompt: delivering a message indicating that Oralytics donated to the person's selected charity
3. Q&A prompt: delivering a brief question relating to oral health; based on the response a tip/advice will be delivered.

The evening engagement strategies include:

1. Standard reciprocity prompt: delivering non-contingent reward points as a "gift" from Oralytics to support goal achievement

2. Reciprocity by proxy prompt: delivering a message indicating that Oralytics donated to the person's selected charity

3. Personalized feedback: delivering a message that contains feedback on brushing based on data from the past 24 hours

Oralytics RL Challenges

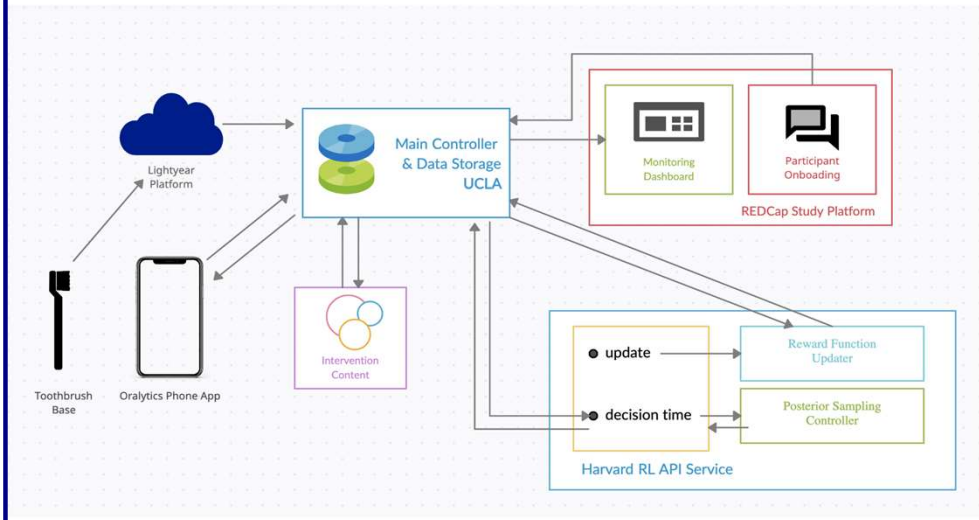
- Aim to select good actions even though high noise/low signal environment
  - RL alg for binary action—send an engagement message versus do not send
- Aim for both within study learning (personalization) and between study learning
  - RL alg uses a stochastic policy, $\pi_t^{\mathcal{L}}$, bounded away from 0, 1

18

Higher signal between do something vs do nothing than between different types of "do something's"
Both within study learning and between study learning
Need to be able to transfer to a potentially different population…. This is rationale for stochastic policy

We may do a decision rule in the two arm study for selecting between the 3 messages.

Oralytics System



Clinical Trials

- Conservative Traditions
  - Justify # User-Participants
  - Pre-Register Treatment Protocol
  - Real-Time Monitoring/Documentation of Treatment Protocol Fidelity

Replicable Science ⇒ One-way Door

20

Prespecify entire treatment protocol, justify sample size === enhance replicability

Conservative traditions due to money, reputations that are at stake thus need to prevent unethical as well as reduce sloppy practices..

Clinical Trials

- Conservative Traditions
  - Justify # User-Participants
  - Pre-Register Treatment Protocol
  - Real-Time Monitoring/Documentation of Treatment Protocol Fidelity

Oralytics RL Algorithm is part of the Treatment Protocol

21

Oralytics RL Algorithm must be pre-specified. Should be stabile and autonomous. NO on-the-fly alterations to algorithm during trial.

Outline

- RL Re-Cap
- Oralytics & Clinical Trials
- Rocky Road to Clinical Trial
  - Testbed Development
  - RL Variants
- Data Collection for Causal Inference & to Assess Performance

22

Testbed development based on impoverished data

Testbed used to assess different variants of the RL alg. User heterogeneity, impact of delayed effects, nonstationarity, model misspecification,

Testbed Development

- Critical for Comparing RL Variants
- Yet Existing Data is Usually Impoverished
  - On different population
  - Subset of treatment actions
  - Partial information on state, reward
- Use Rules of Thumb from Experimental Design Literature and Qualitative Expertise from Domain

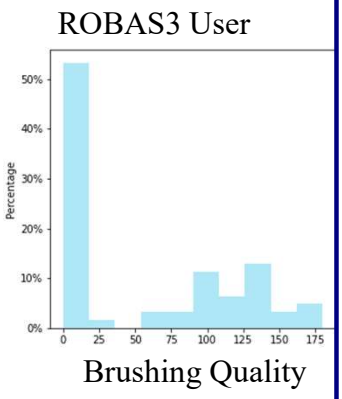
23

Selected RL variant must perform well across the set of realistic testbed variants.

We have access to three data sets on brushing behaviors, ROBAS 2 (Shetty et al. 2020) and ROBAS 3 and a pilot study of 9 participants from the target population. The first two data sets only have observations under no action / intervention; only the pilot study includes the treatment actions. ROBAS 3 has a more sophisticated sensory suite (the same suite that will be used in Oralytics) than ROBAS 2. In addition to brushing duration, ROBAS 3 was able to collect sensory information such as brushing pressure to inform brushing quality, whereas ROBAS 2 was only able to record brushing duration.

Testbed Development

- ROBAS 3: Similar Users, No Treatment Actions
  - Same sensors, 31 Users over 3 Months
- Reward based on brushing quality
  - Brushing quality = min(180, brushing duration minus over-pressure duration)



24

Zero-inflated reward outcome

We use the ROBAS 2 data set to design the prior used in the RL algorithm for the 9 user pilot study. Data from the pilot study of 9 users is used to design the prior for the MRT. We use the ROBAS 3 data set to construct a quality simulation environment that serves as a test-bed for evaluating the RL algorithm candidates.

Testbed Development

- ROBAS 3: Similar Users, No Treatment Actions
  - Same sensors, 31 Users over 3 Months

Step 1: Fit user-specific models for brushing quality under no action (31 model fits)

- Zero inflated generalized linear models each with 18-22 weights to learn
- Allow non-stationarity

25

We fit two types of models: Hurdle (using a Square Root for the continuous part) and Zero-Inflated Poisson

Below is the number of model classes for all users in the (31 users) ROBAS 3 study

Model Class Stationary Non-Stationary

Hurdle (Square Root) 14 12

Zero-Inflated Poisson 17 19

Table 9: Number of selected model classes for the Stationary and Non-Stationary environments.

Baseline features (action==0):

1. Bias / Intercept Term
2. Time of Day (Morning/Evening)
3. Prior Day Total Brushing Quality (Normalized)
4. Weekend Indicator (Weekday/Weekend)
5. Proportion of Non-zero Brushing Sessions Over Past 7 Days
6. Day in Study (Normalized)—only to provide non-stationarity; Non-stationarity is expected due to fact that state space is incomplete and disengagement is likely

We attempt to overfit the user-specific models used in forming the simulation testbed.

We don’t want to overfit too much as then the probabilities in the zero-inflated poisson (or the 0 value in hurdle model) become close to 1 or 0. Thoughts??

Testbed Development

Step 2: How to simulate effects of actions on reward?

Need to impute weights on action and on action×state features in the zero inflated generalized linear model

26

Since the ROBAS data does not contain treatment actions we have to impute effects of actions…

Use weights on baseline features to impute weights on action x state interaction features ("advantage features"):

1. Bias/Intercept Term
2. Time of Day (Morning/Evening)
3. Prior Day Total Brushing Quality (Normalized)
4. Weekend Indicator (Weekday/Weekend)
5. Day in Study (Normalized) only to provide non-stationarity: Non-stationarity is expected due to fact that state space is incomplete and disengagement is likely

Rules of Thumb
Treatment Effects

- Effect (magnitude of weight) of actions on reward is smaller than the weights on baseline features, (weights under no treatment)
- Heterogeneity in magnitude of user-specific weights likely on the order of the variance in the baseline feature weights across users
- Highest signal delayed effects likely negative due to habituation & treatment fatigue.

27

We design the treatment effects --weights on treatment x state interaction features ("treatment x state interaction features ==advantage features")

Further we expect that some features (e.g. day in study) will decrease the effect of sending a message as feature values increase and some features (e.g. prior day total brushing quality) will increase the effect of sending a message as feature values increase. So we are careful in specifying the sign of the weight on a feature.

Testbed Development

Heterogenous effects of action (send engagement message vs no message) on brushing quality:

- Draw user-specific action × state feature weights from a Gaussian with
  - Mean ∝ average magnitude of baseline feature weights (sign from expert/domain)
  - Variance=average variance in baseline feature weights

28

We first construct average effect sizes and then use the average effect size to sample unique effect sizes for each user. To set the average effect size per feature, we first take the absolute value of the weights (excluding that for the intercept term) of the user baseline models fitted using ROBAS 3 data and then average across users for each feature (e.g., the average absolute value of weight for time of day).

To produce small and moderate treatment effect sizes (to produce different testbed variants), we scale each averaged weight by a shrinkage value ζ. Notice that varying the value to ζ is a way to specify different simulation environment variants. We consider two values of ζ for experiments: 1/4 and 1/8 as treatment effects are likely to much smaller than baseline effects.

We expect the weight on the intercept term x action indicator should be larger than the weights on the other interaction features. To construct the mean effect size on the intercept x action term, we average the baseline effect size values across the interaction features and scale by 2. We did this procedure in order to ensure that the effect size on the intercept x action term is approximately 2 times the effect size of other interaction features.

The variance of the normal distribution per feature is found by again taking the absolute value of the weights of the baseline models fitted for each user, scaling

each value by ζ, then taking the empirical variance across users for each feature.

To generate user-specific effect sizes, for each user, we draw an effect size vector from a multivariate normal centered at the mean weight and a covariance matrix with variance values constructed above along the diagonal. Then we take the absolute value of the effect size vector and depending on the feature, we assign the effect size a positive or negative sign.

Overall immediate treatment effect in any state is truncated below by 0. WHY DO YOU THINK WE DO THIS?!!!

Testbed Development

Delayed effects of action (send engagement message vs no message) on brushing quality:

- Proportionally reduce user's action × state feature weights if
  - User is brushing better than average and is getting messages more than 50% of time recently OR
  - User is getting messages more than 80% of the time recently.

29

Some Testbed Variants

- Stationary vs Non-Stationary Environment
- Larger vs Smaller Delayed Effects of Treatment Actions
- Moderate vs Small Population Treatment Effect Sizes

30

Underlying generative model for brushing quality is distributed according to one of the two zero-inflated models (zero-inflated Poisson, hurdle model with square root transform) (which model depends on which model is the best fit for the ROBAS 3 user)

Non-stationary means that one of the features in the generative model is "day of study"

Selected RL variant must perform well across the set of realistic testbed variants.

Creating a Simulated User

- Select user, with replacement, from the 31 ROBAS 3 users
  - Retain this user's estimated baseline weights (AKA, regression coefficients) from the zero-inflated GLM
  - Impute this user's "treatment effect" weights.
  - Impute whether this user opens the app on each of 70 days.
- Above data is used to create this user's potential RL states & rewards….

31

ROBAS 3 also did not collect app engagement. But we will want to use this feature in the RL alg. In the testbed app opening is used to create an app engagement feature.

App opening Challenge

We simulate user app opening behavior to make the simulation environment as close to the study environment as possible. User app opening behavior is the only way a user obtains the most recent schedule of actions. We define the user opening the app as having the app in focus (not just in background). We model a binary 1 if app is in focus and 0 if not. For each day in the study, user i has a probability p^app of opening the app. We sample the user opening the app from Bern(p^app). This imputed data will be used by all RL alg. variants both in terms of state features as well as in creating action schedules. See notes on slide 33.

App Opening Probability, p^app: Since ROBAS 3 did not have viable app opening data, we simulate user app opening as follows. Every user has the same population-level probability of users' opening their apps: p^app = 0.7. This value is informed by Oralytics pilot data on 9 users as follows: For each user in the pilot study, we calculated p^app_i , the proportion of days that the user opened the app during the pilot study (i.e., number of days the user opened the app divided by 35, the number of days in the pilot study). p^app_i ranged from .27 to .98. Then p^app = average

of $\hat{p}_{app\_i} = 0.7$

Warning!!

- The states used to build the simulation testbed may only partially overlap with the RL states.
- The reward used in the simulation testbed may not be the reward used by the RL alg
- The model used for the reward in the simulation testbed may not be the model used by the RL alg

Generative model for user includes for each time point

1. prior day app engagement (related to app opening)
2. Time of Day (Morning/Evening)
3. Prior Day Total Brushing Quality (Normalized)
4. Weekend Indicator (Weekday/Weekend)
5. Proportion of Non-zero Brushing Sessions Over Past 7 Days
6. Day in Study (Normalized)—only to provide non-stationarity; Non-stationarity is expected due to fact that state space is incomplete and disengagement is likely

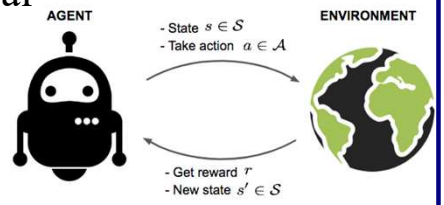
Action (RL alg will produce)

Brushing Quality =min(180, brushing duration-over pressure) generated using above features and that user's zero-inflated generalized linear model.

(including negative delayed effects)

Outline

- RL Re-Cap
- Oralytics & Clinical Trials
- Rocky Road to Clinical Trial
  - Testbed Development
  - RL Variants
- Data Collection for Causal Inference & to Assess Performance



33

1. whether to pool (oralytics –only 140 times per user, very noisy data)
2. update frequency for learning algorithm?

Choice of algorithm's reward (to be discussed this afternoon)

Here we do not discuss:

Oralytics and app opening issue. "in Focus": The app has to be in focus=open to schedule actions because we're using a Local Notification service. This means the mechanics of turning the message schedule into app actions/notifications is handled by Oralytics. We took this approach because there wasn't budget to implement Push (Remote) Notifications, which could be triggered from a cloud server.

This means that all RL variants must produce a schedule of assigned actions for the remaining part of the study on each evening. The first two time points in this schedule use the current state. The remaining time points must use a stale state. This schedule is updated every evening but schedule is only enacted if the user brings the app into focus. If the user brings the app into focus every day then the actions are always delivered using $\pi_t^{\mathcal{L}}$ with the current state.

Re-Cap: Online RL Algorithm

- For $t = 1$ to T do:
  - Receive state S_t
  - Algorithm selects action A_t -- using $\pi_t^{\mathcal{L}}(\cdot\,|S_t, H_{t-1})$
  - Receive reward R_{t+1}
  - If t is an update time do:
    - Update model; update $\pi_t^{\mathcal{L}} \rightarrow \pi_{t+1}^{\mathcal{L}}$
  - End If
- End For

34

RL Handicaps

- Commercial Sensors
  - App does not have direct access to sensor data (access is via commercial cloud)
  - Data accessible at decision times may be less complete that data accessible at the less frequent algorithm update times.
- Impacts choice of RL states

35

RL restricted to state data that can be reliably accessed at time policy selects action. Even then all RL algorithm variants must sometime act with stale state.

We wanted to use zone brushing in the algorithm's reward but we can't obtain this reliably. We also wanted to include in current state, most recent brushing quality but we can't obtain this reliably within 12 hours.

RL Algorithm

Bias versus Variance Tradeoffs:

- 4 State features constructed based on ability to obtain at decision times
  - Exponentially discounted prior brushing quality; Exponentially discounted prior number of messages; Time of day; Prior app engagement

36

These state features do not use prior day's evening brushing quality or current day's morning brushing quality as we are not sure we can obtain by evening brushing time.

Baseline and Advantage Features for RL alg

1. Bias / Intercept Term
2. Time of Day (Morning/Evening) $\in \{0, 1\}$
3. Exponential Average of Brushing Quality Over Past 7 Days (Normalized) $\in R$
4. Exponential Average of Messages Sent Over Past 7 Days
5. prior day app engagement $\in \{0, 1\}$

Testbed uses zero-inflated generalized linear model with features:

Baseline features (action==0):

1. Bias / Intercept Term
2. Time of Day (Morning/Evening)
3. Prior Day Total Brushing Quality (Normalized)
4. Weekend Indicator (Weekday/Weekend)

5. Proportion of Non-zero Brushing Sessions Over Past 7 Days

6. Day in Study (Normalized)—only to provide non-stationarity;

treatment x state interaction features (“advantage features”):

1. Bias/Intercept Term

2. Time of Day (Morning/Evening)

3. Prior Day Total Brushing Quality (Normalized)

4. Weekend Indicator (Weekday/Weekend)

5. Day in Study (Normalized) only to provide non-stationarity: Non-stationarity is expected due to fact that state space is incomplete and disengagement is likely

RL Algorithm

- Bias versus Variance Tradeoffs:
  - Discount rate, $\gamma = 0$
  - Model for $r(s,a) = \mathbb{E}[R_{t+1}(a)|S_t = s\,]$ is linear and stationary.
  - Learning algorithm is Bayesian with subjective prior

AKA

Thompson-Sampling

37

Misspecified reward model

This afternoon we discuss construction of the prior.

Oralytics: Linear Thompson-Sampling

- Learning Algorithm: Bayesian Algorithm
  - Inference for parameters in a linear model for $r(s,a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$
- Optimization Algorithm: Posterior Sampling
  - $\pi_t^{\mathcal{L}}(\cdot\,|s)$ is the posterior probability that
    $$r(s,1) - r(s,0) > 0$$
  - $A_t \sim \pi_t^{\mathcal{L}}(\cdot\,|S_t; H_{t-1})$

38

See Raaz's Monday afternoon lecture for posterior sampling

H_{t-1} has to be specified—based on prior collected data but from whom?

$\pi_t^{\mathcal{L}}(\cdot\,|s)$ is the posterior probability that
$$r(s,1) - r(s,0) > 0$$
Is not quite true, see this afternoon.

RL Variants

Two Clustering Variants

- All users in one cluster: algorithm uses all users' data up to time t to update policy at time t
- Each user in own cluster: user specific algorithm that uses only the user's prior data up to time t to update that user's policy at time t.

Bias due to user heterogeneity

VERSUS

Variance due to noisy rewards

39

So different definitions of H_{t-1}

Update Timing

Two Options

- Each Night (every 2 decision points)
- On weekends (every 10 decision points)

Learning Speed

VERSUS

Complexity in After Study Analyses due to Nuisance Parameters

40

Four Testbed Variants

- Stationary vs Non-Stationary Environment
  - If non-stationary then day in study is a feature in generative model
- High vs Low Delayed Effects of Treatment Actions on User

Here only show moderate population treatment effect sizes

41

Delayed effects are negative.

Comparison of Four RL Variants

	Average Brushing Quality			
ALG_CANDS	STAT HIGH-DELAY	STAT LOW-DELAY	NON_STAT HIGH-DELAY	NON_STAT LOW-DELAY
Weekly, Full Pooling	72 (0.5)	**75 (0.5)**	**69 (0.5)**	**72 (0.5)**
Weekly, No Pooling	71 (0.5)	74 (0.5)	68 (0.4)	71 (0.5)
Daily, Full Pooling	**72 (0.5)**	74 (0.5)	69 (0.5)	72 (0.5)
Daily, No Pooling	71 (0.5)	74 (0.5)	68 (0.5)	71 (0.5)
	25th Percentile Brushing Quality			
ALG_CANDS	STAT HIGH-DELAY	STAT LOW-DELAY	NON_STAT HIGH-DELAY	NON_STAT LOW-DELAY
Weekly, Full Pooling	34 (1.2)	35 (1.3)	36 (1.1)	38 (1.1)
Weekly, No Pooling	32 (1)	33 (1.3)	34 (1.1)	37 (1.0)
Daily, Full Pooling	**35 (1.3)**	**35 (1.3)**	**36 (1.0)**	**38 (1.0)**
Daily, No Pooling	32 (1.2)	34 (1.3)	35 (1.0)	37 (1.1)

Table 2: Value in each parenthesis is the standard error of the mean across 100 simulated trials.

42

Weekly means weekly update interval; Daily means update to model and thus policy each evening.

There is always heterogeneity in treatment effect sizes across users.

Underlying generative model for brushing quality is distributed according to one of the two zero-inflated models (zero-inflated Poisson, hurdle model with square root transform) (which model depends on which model is the best fit for the ROBAS 3 user)

Non-stationary means that one of the features in the generative model is “day of study”

RL Design Decisions

- Weekly Updates
  - Learning Algorithm updates policy each Sunday evening
- Full Pooling
  - One Learning Algorithm for all users.
  - At update time algorithm uses all users' data to update policy.

43

Outline

- RL Re-Cap
- Oralytics & Clinical Trials
- Rocky Road to Clinical Trial
  - Testbed Development
  - RL Variants

- Data Collection for Causal Inference & to Assess Performance

44

Data Collection

- Save "action selection probabilities, $\pi_t^{\mathcal{L}}$'s" from RL policy.
  - off-policy evaluation;
  - estimating causal excursion effects; estimating mediational effects;
- Save algorithm components needed to estimate standard errors of estimators formed in after-study analysis

45

Data Collection

Between study analyses

- Clustering by RL algorithm impacts confidence in between-study analyses
  - Elevated standard errors
  - Kelly will discuss next!

46

20 min for Discussion!

Break & Discussion

- Most implementations of RL in practice do not store the policy. Why might this be the case?
- What should the Oralytics RL algorithm do to handle missing data?
  - In learning alg?
  - In optimization alg?
- How might you design an RL algorithm to deal with delayed effects of the actions?

48