Poster Presentation

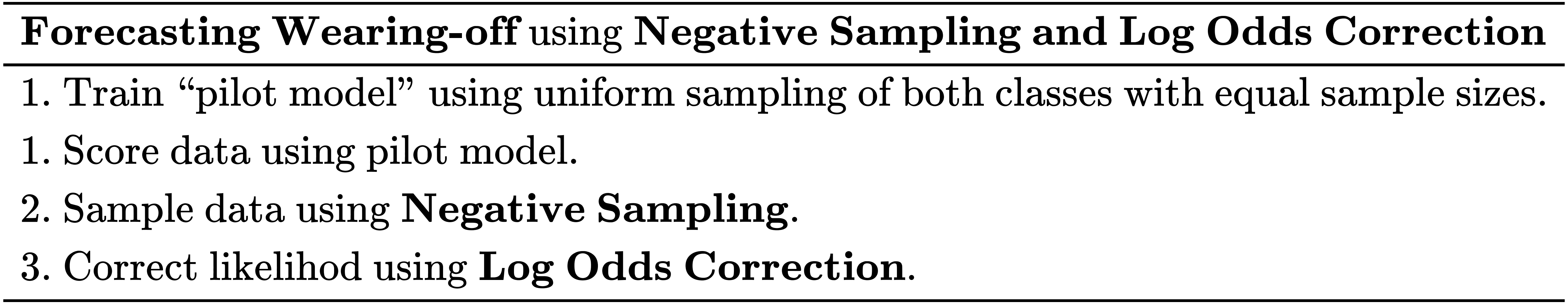
May 2023

VICTORINO John Noel

Overview for May

- **ABC 2023 Paper:** Handling Class Imbalance with Non-Uniform Negative Sampling, Log Odds Correction, and Oversampling in Forecasting Parkinson's Disease Wearing-off with Fitness Tracker Dataset
    - Paper: In Progress
    - Code & Results: In Progress
    - Presentation: for May 2023 Poster Presentation

Proposed Method



Forecasting Wearing-off using **Negative Sampling and Log Odds Correction**

1. Train “pilot model” using uniform sampling of both classes with equal sample sizes.
1. Score data using pilot model.
2. Sample data using **Negative Sampling**.
3. Correct likelihod using **Log Odds Correction**.

- **Contributions**
  - Application of Negative Sampling and Log Odds Correction to assign larger probabilities to more informative instances, i.e., wearing-off labels and informative normal labels.
  - Improves forecasting of wearing-off by *X%* as compared to ABC 2022 paper.
- **Paper category**: Application-based paper

Forecasting Wearing-off Task

- **Wearing-off** is a phenomenon where the effect of a drug wears off before the next dose is taken.
- Let $\{(X_t, y_t)\}_{t=1}^{N}$ be the training data that satisfies

$$Pr(y_{t+1} = 1|M) = p(M;\theta), M = \begin{bmatrix} X_t, & y_t \\ \vdots & \vdots \\ X_{t-1\text{hour}}, & y_{t-1\text{hour}} \end{bmatrix}$$

- $y \in \{0, 1\}$ is wearing-off (1) or not (0),
- M is the feature matrix of X and y from the last hour until time t,
- θ is the model parameters

Forecasting Wearing-off Task

For example, if the dataset is in 15-minute interval.

- Forecast: y_{t+4}
- Input:

$$M = \begin{bmatrix} X_t & y_t \\ X_{t-1} & y_{t-1} \\ X_{t-2} & y_{t-2} \\ X_{t-3} & y_{t-3} \\ X_{t-4} & y_{t-4} \end{bmatrix}$$

Imbalance in Wearing-off Dataset

Wearing-off Statistics

- Mean: 8.276%
- Median: 7.682%
- Standard Deviation: ± 3.624
- Participants Below Median: 1, 2, 7, 9, 12, 13
- Participants Equal / Above Median: 3, 4, 5, 6, 8, 10, 11

Participant	Normal ($y = 0$)	Wearing-off ($y = 1$)
1	92.361	7.639
2	97.348	2.652
3	86.771	13.229
4	90.492	9.508
5	92.318	7.682
6	89.815	10.185
7	98.821	1.179
8	87.973	12.027
9	92.992	7.008
10	86.771	13.229
11	90.705	9.295
12	92.519	7.481
13	93.528	6.472

Imbalanced Data Definition

- Let N_1 wearing-off cases, N_0 normal cases, and $N = N_1 + N_0$ total cases.
- When N_1 much smaller than N_0, appropriate to assume that N_1 increases in slower rate compared with N_0, i.e.,

$$\frac{N_1}{N_0} \xrightarrow{P} 0 \text{ and } N_1 \xrightarrow{P} \infty \text{ as } N \to \infty$$

- This requires that $Pr(y_{t+1} = 1) \to 0$ as $N \to \infty$ on model side.

Imbalanced Data Definition

Assume that $\theta = (\alpha, \beta^T)^T$, log odds is as follows:

$$g(M;\theta) := \log\{\frac{p(M;\theta)}{1-p(M;\theta)}\} = \alpha + f(M;\beta)$$

- $f(M;\beta)$ is the smooth function of β, e.g., neural network or logistic regression.
- The true parameter $\theta^* = (\alpha*, \beta^{*T})^T$ with assumption that
  - $\alpha^* \to -\infty$ as $N \to \infty$,
  - β^* is essentially fixed, as long as β^* has a finite limit
- For both assumptions indicate that marginal & conditional probabilities of positive instance are **small**.
- This means that changes in covariates do not convert small probability event to large probability event.

Number of Positive Instance Matter

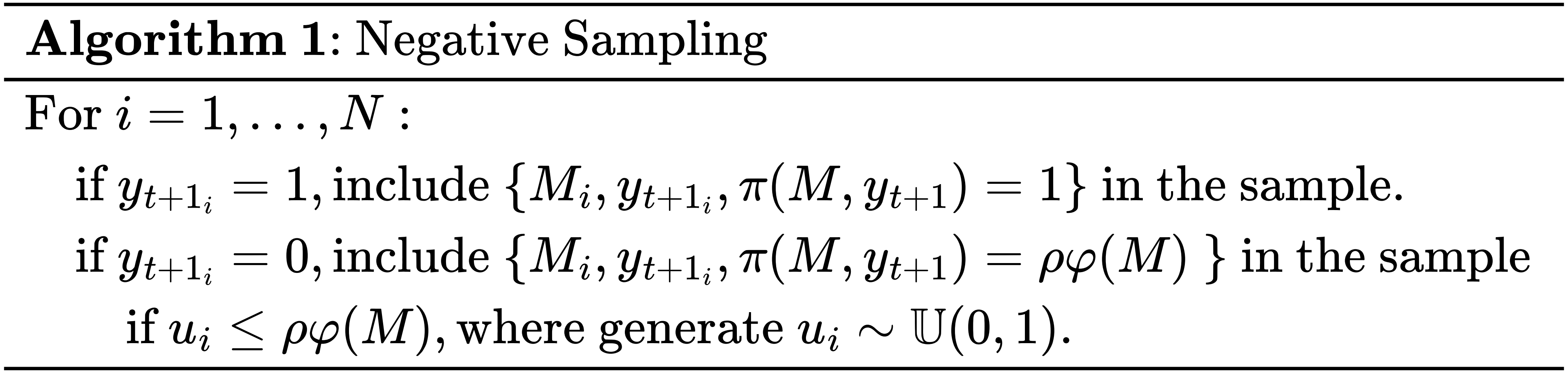
- For rare event, estimation error rate for θ is related to N_1 instead of N.
- Using the full training data, maximum likelihood estimator (MLE) of θ is

$$\hat{\theta}_{MLE} := \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \{ y_{t+1_i} \cdot g(M_i;\theta) - \log(1 + e^{g(M_i;\theta)}) \}$$

(1) Negative Sampling

- **Keep all wearing-off instances while significantly subsampling negative instances**, since available information ties with wearing-off instances instead of full training data.
  - **Uniform sampling**: keeping the ratio of positive to negative samples.
  - Reduce the number of negative samples using **nonuniform sampling**.

(1) Negative Sampling



Algorithm 1: Negative Sampling

```
For $i = 1, \ldots, N$ :
    if $y_{t+1_i} = 1$, include $\{M_i, y_{t+1_i}, \pi(M, y_{t+1}) = 1\}$ in the sample.
    if $y_{t+1_i} = 0$, include $\{M_i, y_{t+1_i}, \pi(M, y_{t+1}) = \rho\varphi(M)\ \}$ in the sample
        if $u_i \leq \rho\varphi(M)$, where generate $u_i \sim \mathbb{U}(0, 1)$.
```

- ρ: sampling rate on negative instances.
- $\pi(M, y_{t+1})$: sampling probability for given (M, y_{t+1}).
  - $\pi(M)$: sampling probability for negative cases, as shorthand.
- $\varphi(M)$: "pilot model" from uniform sampling of both classes with equal sample sizes.
- $\delta = 1$ if data point is selected; otherwise, $\delta = 0$.

(2) Log Odds Correction

Recap...

$$Pr(y_{t+1} = 1|M) = p(M;\theta) = \frac{1}{1+e^{-g(M;\theta)}},$$

$$\hat{\theta}_{MLE} := \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \{y_{t+1_i} \cdot g(M_i;\theta) - \log(1+e^{g(M_i;\theta)})\}$$

For the **Log Odds Correction**, for the data included in the subsample (where $\delta = 1$), the conditional probability of $y_{t+1} = 1$

$$Pr(y_{t+1} = 1|M, \delta = 1) = \frac{1}{1+e^{-g(M;\theta)-l}}, \text{where } l = -\log\{\pi(M)\}$$

$$\hat{\theta}_{lik} := \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \delta_i [y_{t+1_i} \cdot g(M_i;\theta) - \log\{1+e^{g(M_i;\theta)+l_i}\}]$$

This is done to

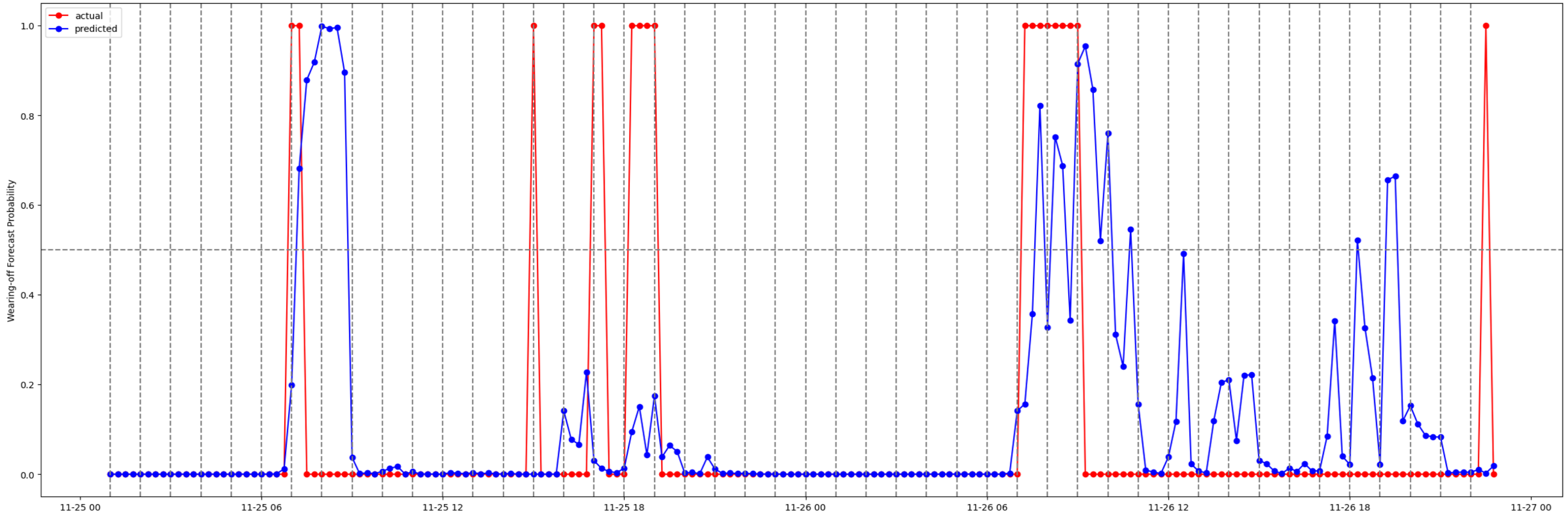
- avoid assigning smaller weights to more informative instances
- avoid biased estimator due to Algorithm 1

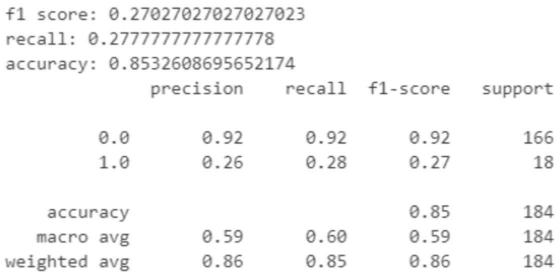
Summary of Method

```
Forecasting Wearing-off using Negative Sampling and Log Odds Correction
1. Train "pilot model" using uniform sampling of both classes with equal sample sizes.
1. Score data using pilot model.
2. Sample data using Algorithm 1: Negative Sampling.
3. Correct likelihod using Log Odds Correction.
```

Initial Results - Base XGBoost Estimator

- Base XGBoost



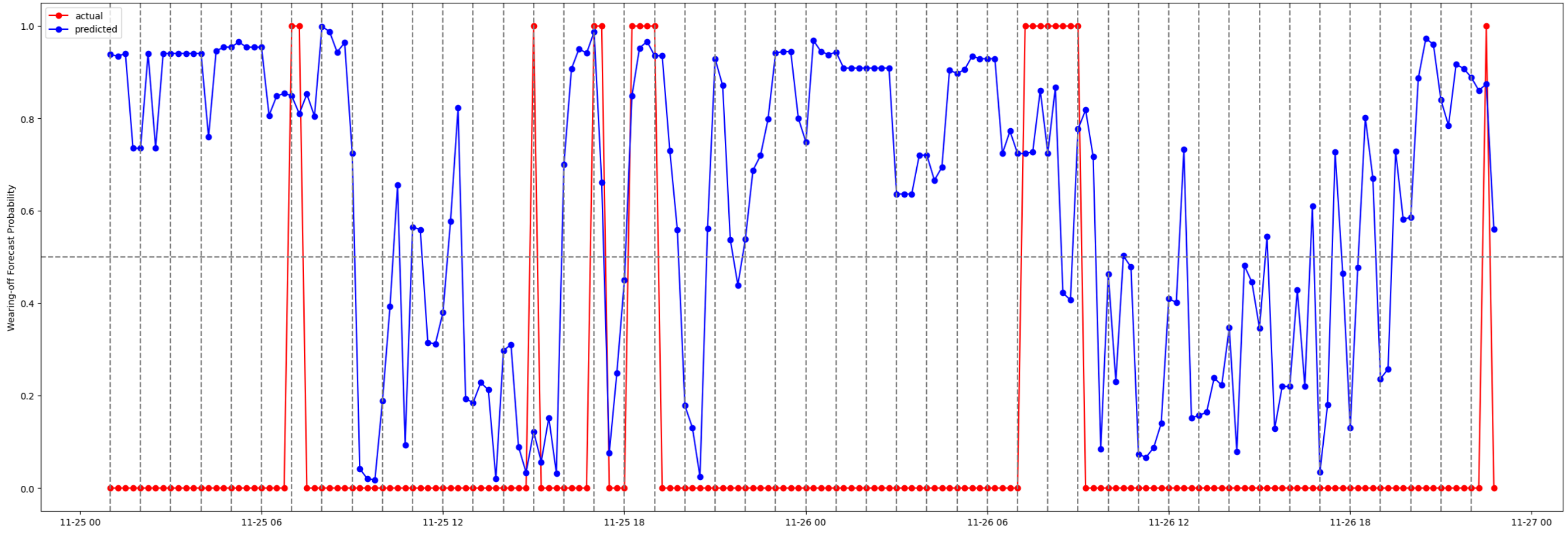


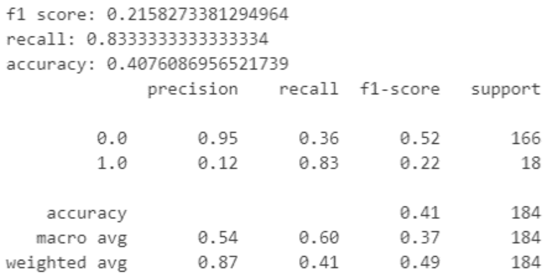
```
f1 score: 0.27027027027027023
recall: 0.2777777777777778
accuracy: 0.8532608695652174
              precision    recall  f1-score   support

         0.0       0.92      0.92      0.92       166
         1.0       0.26      0.28      0.27        18

    accuracy                           0.85       184
   macro avg       0.59      0.60      0.59       184
weighted avg       0.86      0.85      0.86       184
```

- Proposed Method should:
  - Reduce negative samples and as a result improve the performance of the model.
  - Boost probabilities from positive samples.

Initial Results - Proposed Negative Sampling with Log Odds

- Proposed Negative Sampling with Log Odds Correction





```
f1 score: 0.2158273381294964
recall: 0.8333333333333334
accuracy: 0.4076086956521739
              precision    recall  f1-score   support

         0.0       0.95      0.36      0.52       166
         1.0       0.12      0.83      0.22        18

    accuracy                           0.41       184
   macro avg       0.54      0.60      0.37       184
weighted avg       0.87      0.41      0.49       184
```

- However, results have not improved the model on F1 score and forecasts:
    - There might be discrepancy between the reference papers vs implementation.

Comments from Inoue-sensei's Poster Presentation

- Is the test data resampled or kept the same? (latter is realistic)
    - **ANS**: Test data is kept the same.
- Why do you want to apply the proposed method (negative sampling and log odds correction)?
    - **ANS**: Because under binary model with rare event data, the difference between **full data estimator** & **true parameter** converge to a normal distribution with similar rate to the number of **rare positive cases, i.e., wearing-off**. As such, subsampling the negative instances, i.e., normal cases, is justified.
    - If uniform sampling (equal number of classes) is used, there is **information loss**. This is the reason for the **log odds correction** as it gives more weight to samples with higher probability of being positive, keeping samples that confuses the model, and learns to separate true positive from negative samples that are similar to positive samples.
- Aren't there other class imbalance techniques to try? If so, which should be baseline?
    - **ANS**: Looking into this area as part of the RRL.

Next Steps

- Finish ABC paper (if forecast results have improved.)
- Otherwise, try to improve forecast results using other methods previously studied.
    - Inoue-sensei's suggestion: look into **adaptive resampling**.
    - Then, submit to journal.
- Prepare data collection experiment using new Garmin App Collector to include
    - Accelerometer, HRV
    - Active feedback of patients from forecasts
- Development work on Pose Estimation using Jetson Nano.
- Work on Additional tasks due to ABC 2023 challenge.