Recursive Estimation of User Intent from Noninvasive Electroencephalography using Discriminative Models

Niklas Smedemark-Margulies[1], Basak Celik[2], Tales Imbiriba[2], Aziz Kocanaogullari[2,3], Deniz Erdogmus[2]

[1]Khoury College of Computer Science, Northeastern University, [2]Department of Electrical and Computer Engineering, Northeastern University, [3]Analog Devices

Paper Code Dataset

Goal

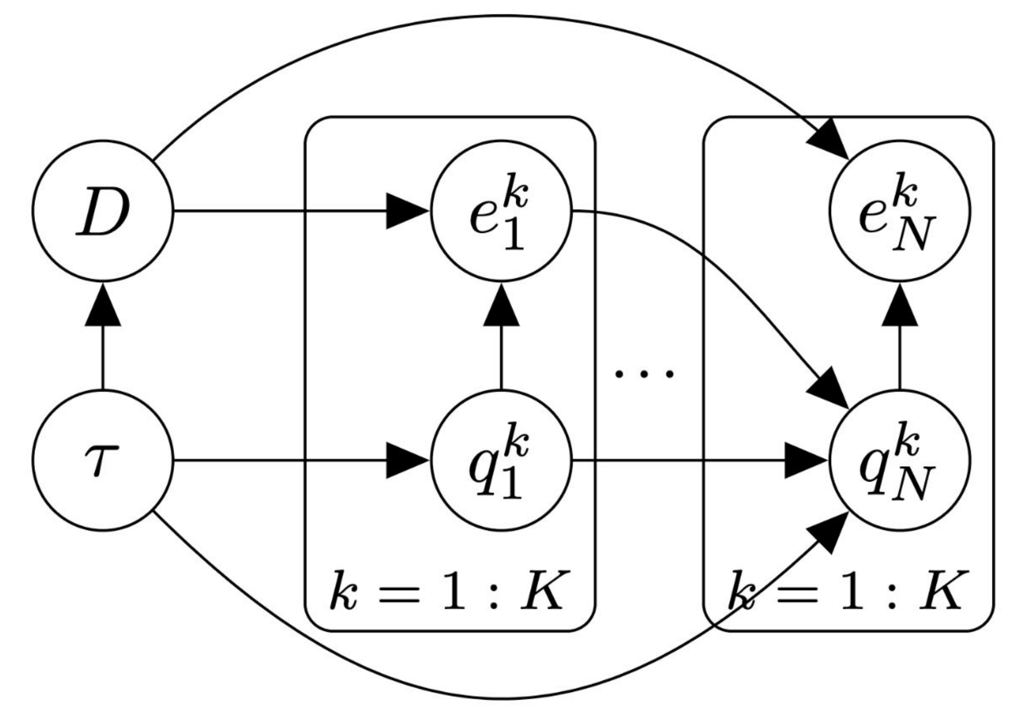
We seek to improve methods for communicating using non-invasive EEG.

- We focus on **Rapid serial visual presentation (RSVP) paradigm**, in which a subject thinks of a target symbol, then we query them with a series of quickly flashed possible symbols while measuring their EEG responses, and we try to update the estimated probability of each symbol.
- We design a **recursive Bayesian update** that uses **discriminative (classifier) models**.
- Discriminative models are generally **easier to train** when compared to generative models.
- Our method **enables the use of new families of EEG signal models** for the RSVP typing task.

Key Results

- We design an effective **typing task simulator** using a large RSVP benchmark dataset.
- The typing simulator **enables error metrics that encompass the whole typing task** such as **information transfer rate (ITR)** for candidate models.
- The proposed methodology led to **higher ITR** and **balanced accuracy** when compared with benchmark **generative models,** even when using small discriminative models.
- Among discriminative models, **1D and 2D CNNs led to highest ITR and balanced accuracy.**
- To calculate balanced accuracy of generative models, we used Bayes Theorem with uniform or empirical priors to compute the posterior over labels.

1. Probabilistic Graphical Model



τ - Previously typed text (known).
D - Subject's target symbol (unknown).
q - Query symbols we present to subject.
e - EEG evidence we measure during query.
K - Each query contains K symbols.
N - To type one symbol, we allow ourselves at most N queries.

RSVP Typing Procedure:

1. To type a symbol, we begin by examining the previously typed text τ, giving us a prior probability distribution over the alphabet.
2. The user's target symbol D depends only on what they want to type and what has been typed so far (e.g. imagine that τ = 'PIZZ' and D = 'A').
3. Before each query, we sample K query symbols q^1 thru q^K according to their current probability.
4. We present each query symbol, and measure the corresponding EEG responses e^1 thru e^K.
5. After updating our symbol probabilities, we can repeat the process for the next query.
6. If any symbol passes a decision threshold, it is typed. If we perform N queries without passing the threshold, we simply type the current highest probability symbol.

Modeling assumptions:

- EEG responses are binary: "**target**" and "**non-target**".
- EEG responses are conditionally independent, given the queried symbol: each response depends *only* on the current query symbol and the desired symbol.

2. Recursive Bayesian Update

The main task in RSVP typing is to update our posterior estimate for the alphabet. In the k^{th} symbol of the N^{th} query, let our displayed query symbol q be some particular symbol α; let β be any other symbol. We collect the corresponding EEG response e, apply our pre-trained classifier to estimate the label probabilities $p(l|e)$, compute a label prior $p(l)$, and then we can apply our recursive update rule. Let Q be all previous query symbols, and E be all previous EEG responses.

We derive our update rule; π represents a normalized posterior, and γ represents an unnormalized posterior. We begin with Baye's rule, including all current and previous observations, and apply conditional independences to describe the presented symbol α:

$$\pi_N^k(D{=}\alpha) := p(D{=}\alpha|\tau, Q, E, q{=}\alpha, e) \quad (1)$$

$$= \frac{p(e|D{=}\alpha, \tau, Q, E, q{=}\alpha)p(D|\tau, Q, E, q{=}\alpha)}{p(e|\tau, Q, E, q{=}\alpha)} \quad (2)$$

$$\propto p(e|D{=}\alpha, \cancel{\tau, Q, E}, q{=}\alpha)p(D{=}\alpha|\tau, Q, E, \cancel{q{=}\alpha}) \quad (3)$$

$$= p(e|D{=}\alpha, q{=}\alpha)p(D{=}\alpha|\tau, Q, E). \quad (4)$$

Next, we observe a recursion, and introduce the binary label l:

$$\pi_N^k(D{=}\alpha) \propto p(e|D{=}\alpha, q{=}\alpha)\pi_N^{k-1}(D{=}\alpha) \quad (5)$$

$$= \left[\sum_\ell p(e|\ell)\underbrace{p(\ell|D{=}\alpha, q{=}\alpha)}_{p(\ell=+)=1}\right]\pi_N^{k-1}(D{=}\alpha) \quad (6)$$

$$= p(e|\ell{=}+)\pi_N^{k-1}(D{=}\alpha) \quad (7)$$

$$= \frac{p(\ell{=}+|e)p(e)}{p(\ell{=}+)}\pi_N^{k-1}(D{=}\alpha) \quad (8)$$

$$\propto \frac{p(\ell{=}+|e)}{p(\ell{=}+)}\pi_N^{k-1}(D{=}\alpha) := \gamma_N^k(D{=}\alpha). \quad (9)$$

An analogous derivation for any other symbol β gives a similar recursion:

$$\gamma_N^k(D{=}\beta) = \frac{p(\ell{=}-|e)}{p(\ell{=}-)}\pi_N^{k-1}(D{=}\beta). \quad (10)$$

After computing γ for all symbols, the final step is to normalize the alphabet:

$$\pi_N^k(D{=}\alpha) = \frac{\gamma_N^k(D{=}\alpha)}{\sum_{d\in(D)}\gamma_N^k(d)}, \quad \pi_N^k(D{=}\beta) = \frac{\gamma_N^k(D{=}\beta)}{\sum_{d\in(D)}\gamma_N^k(d)}. \quad (11)$$

3. Dataset

We use the RSVP Benchmark Dataset from Zhang et al, 2020 (https://doi.org/10.3389/fnins.2020.568000). This dataset contains 64 subjects and a total of over 1M binary EEG trials.

Preprocessing: Data is pre-processed using a notch filter for AC line noise (50 Hz), a bandpass filter (1-20 Hz), downsampled in time by 2x, and then segmented into trials containing 500ms of data beginning at each stimulus onset.

Data-split: In all experiments, data is **pooled across subjects**. **80%** of each subject's data is used **for train**, and **20% for test**. We repeat all experiments using **5 random splits of the dataset.**

4. Experiments

We evaluate model performance using **balanced accuracy** and **information transfer rate (ITR)**:

$$\mathrm{ITR}(A, P) := \log_2(A) + P\log_2 P + (1{-}P)\log_2\frac{1{-}P}{A{-}1}.$$

Balanced accuracy is the average of accuracy on each class, and is computed using the entire test set. To estimate ITR, we follow the simulated typing procedure in Algorithm 1, also described below.

To type one symbol:

- Start with a uniform alphabet prior.
- Select an arbitrary target symbol.
- Repeatedly:
  - Sample K query symbols according to their current estimated probabilities.
  - For each symbol, fetch a random EEG trial from the test dataset. If the symbol matches the target, fetch a **target** trial. Otherwise fetch a **non-target** trial.
  - Feed these sampled EEG trials to the model, and perform recursive updates.
  - If a symbol exceeds the decision threshold, type it and break. After N repetitions without success, give up.

This procedure is repeated for T symbols

Finally, we compute an ITR based on the size of the alphabet, the number of symbols attempted, and the number of symbols that were correctly typed.

Algorithm 1: Estimating ITR via simulated typing. Note that the likelihood L predicted by the model at each step can be either $p(e|\ell)$ or $p(\ell|e)/p(\ell)$, as described in Sec. 2.3.

Input: Trained model $f(\cdot)$, Pos. and Neg. Test Data $\mathcal{X}^+, \mathcal{X}^-$, Iterations T, Symbols per query K, Attempts per symbol N, Alphabet size A, Decision threshold δ,
Output: ITR

$C \leftarrow 0$ // correct count
for $t \leftarrow 1 : T$ **do** // target symbols
  $\pi_0 \leftarrow (\frac{1}{A}, \ldots, \frac{1}{A})$ // unif symbol prior
  for $n \leftarrow 1 : N$ **do** // chances to update
    // sample query symbols
    $\{q_i\}_{i=1}^K \sim \pi_{n-1}$
    // sample matching data
    $\{x_i \sim \mathcal{X}^+$ **if** $q_i = t$ **else** $x_i \sim \mathcal{X}^-\}_{i=1}^K$
    $L \leftarrow f(\{x_i, q_i\})$ // model likelihoods
    Calc. π_n from π_{n-1} and L // Eq. 9-11
    // see if target was typed
    ind, val $\leftarrow \arg\max(\pi_n), \max(\pi_n)$
    if ind=t and val $\geq \delta$ **then** $C \leftarrow C + 1$ and **break**
return ITR$(A, C/T)$

Note that to evaluate balanced accuracy of generative models, we must convert their likelihood output $p(e|l)$ into a label posterior $p(l|e)$ using Bayes' rule. This requires choosing a prior over labels; we consider both a uniform prior (50:50) and an empirical prior (the class fraction observed in the training set).

5. Results

We found that the proposed methodology, which enables the use of discriminative classifiers, led to:

- **improvements** in both **ITR** and **balanced accuracy.**
- Specifically, all **discriminative models outperformed** the **baseline generative models** in both balanced accuracy and ITR.
- The **greatest benefit** was observed when using **discriminative neural network models** to perform updates.

In Table 1, we show the balanced accuracy and ITR of each model. In Figure 2, we show the ITR of various models as a function of the number of trainable parameters, to show that the proposed method offers a strong benefit for both large and small models.

Table 1: Balanced Accuracy and Information Transfer Rate (ITR) for Discriminative (Disc) and Generative (Gen) Models. The discriminative strategy yield models with higher balanced accuracy and information transfer rates. Entries show mean and standard deviation across 5 random train/test splits. Control models use the discriminative strategy but always assign high probability to a fixed class. See Sec. 3.6 for ITR calculation.

Strategy	Model	Balanced Acc	ITR
Disc	LogR	0.730 ± 0.001	0.817 ± 0.047
Disc	EEGNet	0.745 ± 0.003	0.930 ± 0.050
Disc	1D CNN	**0.782 ± 0.005**	**1.103 ± 0.047**
Disc	2D CNN	**0.779 ± 0.004**	**1.153 ± 0.068**
Gen	LDA (Emp Prior)	0.509 ± 0.000	0.678 ± 0.077
Gen	LDA (Unif Prior)	0.687 ± 0.003	
Gen	LogR (Emp Prior)	0.500 ± 0.000	0.218 ± 0.022
Gen	LogR (Unif Prior)	0.694 ± 0.002	
Control	Always Class 0	0.500 ± 0.000	0.000 ± 0.000
Control	Always Class 1	0.500 ± 0.000	0.000 ± 0.000

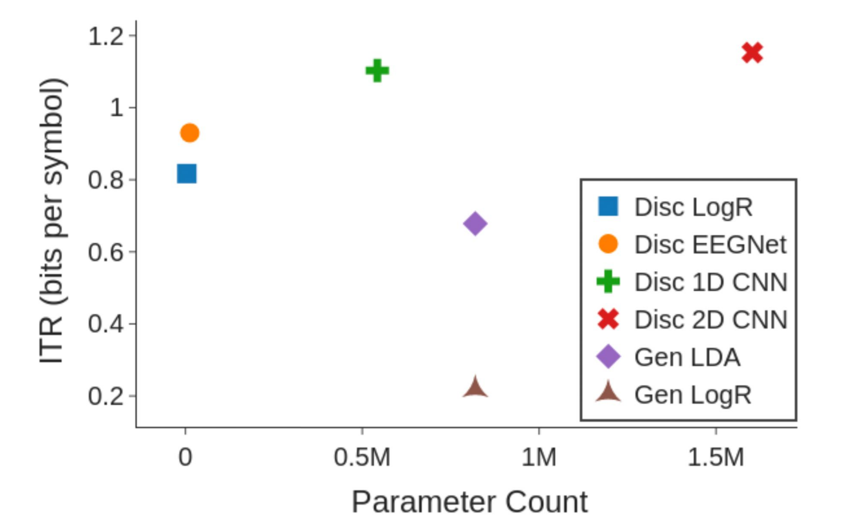


Fig. 2: Information Transfer Rate vs Model Size. Discriminative (Disc) models outperform generative (Gen) models across a wide range of sizes. Among Disc models, performance increases with model size.