Information Prevalence Inference using the i-th order statistic: i-test Toolbox

•Overview

i-test Toolbox is the MATLAB implementation of the second-level (group-level) statistical test for the decoding accuracy proposed by Hirose (https://www.biorxiv.org/content/10.1101/578930v2), which is an extension of "Permutation-based prevalence inference using the minimum statistic", proposed by Allefeld et al., 2016.

i-test evaluates whether the proportion of the population that has label information in the brain from experimental results (decoding accuracy; D-Acc) from multiple participants,

e.g. we can argue that “more than half of the population has label information in the brain activity.”

Basic usage

The implementation-level codes are fond in ***implementation*** directory. These can be applied to your experimental results (D-Acc’s from multiple participants) with known values (from your experimental results or predetermined parameters) with specific assumptions.

If you want to use your own assumptions, use the lower-level code **itest/itest.m**. For practice, use **GUI/itest_GUI**, which is a graphical user interface of **itest/itest.m**.

Feedback and bug report

Any feedback and bug reports are welcome. Please mail to satoshi.hirose [at] nict.go.jp (please replace the [at] with the ‘@’ symbol).

Copyright

i-test Toolbox is free but copyrighted software, distributed under the terms of the GNU General Public Licence. Further details on copyleft can be found at http:// www.gnu.org/copyleft/. No formal support or maintenance is provided or implied.

Developed and tested environment

MATLAB 2019b on MAC and MATLAB 2020a on Linux.
There is no toolbox dependencies,
except for the cross validation simulation for replication of the study (requires libsvm; https://www.csie.ntu.edu.tw/~cjlin/libsvm/).

Installation

To install i-test Toolbox, download files and add directory **itest**, **implementation**, **subfunctions** to MATLAB path.

Future Release

So far, the followings are not available (please wait future release).
1) GUI interface for the implementation-level algorithm, e.g. i-test-unif-bino.
2) i-test without the assumption of identical distribution across participant (i-test-woid).
3) Other implementation-level algorithm with particular assumptions.

For other requests please contact me.

Carsten Allefeld, Kai Görgen and John-Dylan Haynes, 'Valid population inference for information-based imaging: From the second-level t-test to prevalence inference', NeuroImage 2016, https://doi.org/10.1016/j.neuroimage.2016.07.040. https://github.com/ allefeld/prevalence-permutation/)

•Quick how to's

1.Do experiment and finish the decoding analysis for each participant (first level analysis).

*Try MVPC toolbox (https://github.com/satoshi-hirose/MVPC_toolbox) if you are a beginner to decoding analysis.

2. Decide the two threshold parameters, e.g. $\gamma_0 = 0.5, \alpha = 0.05$.

*Keep in mind that significant result of the i-test leads the conclusion that *"proportion larger than γ_0 of the population has label information in the brain activity" with false-positive probability less than α.*

3. Run the function itest_unif_bino as follows

```
[H, prob] = itest_unif_bino(SD,N_trial,Pcm,g_0,alpha)
```

SD: Decoding accuracies (Nx 1 vector)
N_trial: Number of trials (integer)
Pcm: Chance level (e.g. 0.5 for binary)
g_0 and alpha: threshold parameters (e.g. g_0=0.5, alpha=0.05)

4. If H=1, the result is significant (non-significant if H=0), the p-value is found as prob.

IMPORTANT NOTE: Please understand what are assumed in the algorithm before your formally use this implementation.

•Quick understanding of algorithm/codes

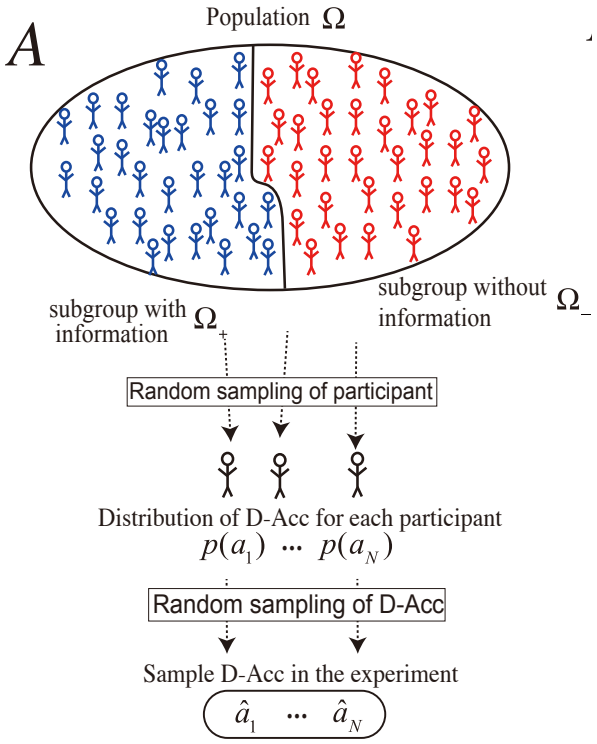
You love to see MATLAB codes?: Download the example experimental results from Empirical_Experiment_Results.mat, put it in **Replication_of_study/Results** directory (create, if not exist), run **Replication_of_study/Sec4_Empirical_Data_Analysis.m** and check the code. This code is programmed by using MATLAB default functions, without functions in itest toolbox, and without minor process (error handling and input parser etc.). So you can easily understand the algorithm without referring other source codes.

You love to see mathematical equations?: Please see my original paper (https://www.biorxiv.org/content/10.1101/578930v1). Particularly, Section 2 and Appendix A.

You love to see algorithm-level pseudo codes?: Please see below "Run-level functions" section.

•Theory

- Problem Definition



The population Ω can be partitioned into two subgroups, Ω_+ and $\Omega\_$. People in Ω_+ have label information in their brain activity and thus, the expectations of their D-Acc are higher than the chance level, while people in $\Omega\_$ do not have label information and the expectations are at chance level. When a person with index n is randomly chosen from the population, he/she belongs to $(n \in \Omega_+)$, with a probability γ, or otherwise belongs to $(n \in \Omega\_)$. The experiment is expressed as a two-step random sampling. First, N participants are randomly sampled from the population independently of each other (Random sampling of participant). Each sampled participant is associated with the probability distribution (probability mass function) of the D-Acc ($p(a_n)$). Second, the experimental results (sample D-Acc; $\hat{a}_n$) are randomly sampled from the distribution for each participant (Random sampling of D-Acc).

The objective of the i-test is to test whether γ ($= P(n \in \Omega_+)$) is larger than the predetermined threshold γ_0 with the significance threshold α from participants' experimental results ($\{\hat{a}_n : n = 1...N\}$). The test statistic of i-test is the i-th order statistic of the experimental results (i-th lowest D-Acc; $\hat{a}_{(i)}$)

- Notations, variable names, and variable types in the codes

Predetermined Constant and parameters

N (`N: integer`): number of participants

γ_0 (`g_0: probability`): prevalence threshold

α (`alpha: probability`): statistical threshold

i (`i: integer`): rank of order statistic

i_{opt} (`i_opt: integer`): optimized rank of order statistic

Experimental Results

$\hat{a}_n$ (`SD: column vector of probability`): observed D-Acc of participants in the experiment ($n = 1...N$)

Index

n (`n: integer`): index of participant

Population parameter

γ (`gamma: vector of probability`): true population information prevalence

Parameters for binomially distributed D-Acc

N_{trial} (`N_trial: integer`): number of trials (known)

$P_{correct+}$ (`Pcp: vector of probability`): probability of the correct decoding in a trial for a participant with label information

$P_{correct-}$ (`Pcm: probability`): probability of the correct decoding in a trial for a participant without label information (=chance level; known)

Probablistic variable

a_n (`a_n: probability`): D-Acc of a participant with index n

$a_{(i)}$(`order_stat: probability`):i-th order statistic of the D-Acc

Probability mass function (PMF)

$p(a_n \mid n \in \Omega_+)$ (`pp_true: pmf style`):PMF of decoding accuracy with label information.

$p(a_n \mid n \in \Omega\_)$ (`pm_true: pmf1 style`): PMF of decoding accuracy without label information.

$\tilde{p}(a_n \mid n \in \Omega\_)$ (`pm_est: pmf1 style`): PMF of decoding accuracy without label information.

Probability

$P(Significant)$ (`P_sig: matrix of probability`): Probability that itest report significant resutls

Function

$BPDF(k, K, \rho) =_K C_k \rho^k (1-\rho)^{(K-k)}$: binomial probability mass function

$BCDF(k, K, \rho) = \sum_{h=0}^{k} BPDF(h, K, \rho)$: binomial cumulative probability mass function

-variable types

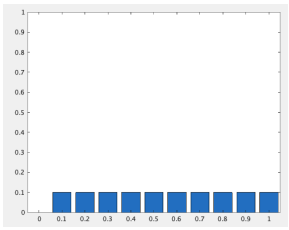
`probability`: real number between 0 and 1
`integer:` positive integer
`pmf1`: Probability mass function 2 x [Number of possible D-Acc] matrix)

first row : possible values of D-Acc (elements should be probability)

second row: Probability for each values in the first row (elements should be probability & the sum should be 1)

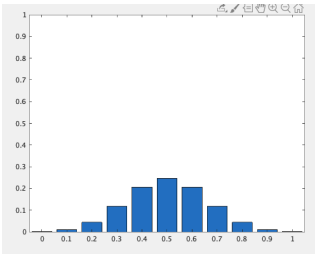
Example 1: Uniform distribution

```
p = [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1;
     0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1];
```



Example 2: binomial distribution

```
p = [0:0.1:1;
     binopdf(0:10,10,0.5)]
```



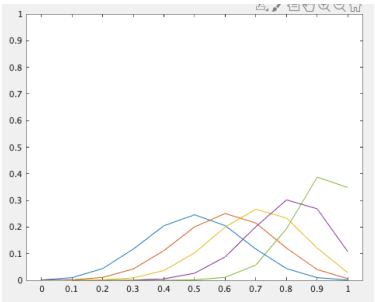
`pmf`: Probability mass functions ([Number of PMF]+1) x [Number of possible D-Acc] matrix)

first row : possible values of D-Acc (elements should be probability)

second, third... row: Probability for each values in the first row. Each row corresponds to one PMF (elements should be probability & sum of each row should be 1)

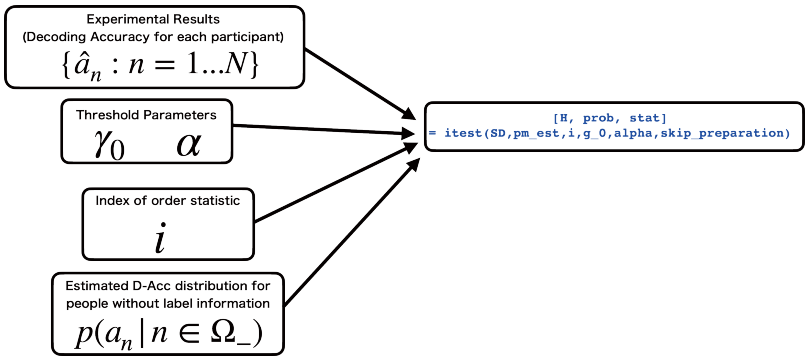
Example : binomial distributions

```
p = [0:0.1:1;
     binopdf(0:10,10,0.5);
     binopdf(0:10,10,0.6);
     binopdf(0:10,10,0.7);
     binopdf(0:10,10,0.8);
     binopdf(0:10,10,0.9)]
```



•Run-level functions

- *i*-test (itest/itest.m)



Usage

```
[H, prob, stat] = itest(SD,pm_est,i,g_0,alpha,skip_preparation)
```

Input

SD: Experimental results (D-Acc) (`column vector of probability`)
pm_est: estimated D-Acc distribution without label information (`pmf1 style` or `row vector of probability`)
i: rank of order statistic (`integer`)
g_0: Prevalence threshold (`probability`; default 0.5)
alpha: Statistical threshold (`probability`; default 0.05)
skip_preparation:Skip input parser if true (`boolean`; default False)

Output

H: Whether result is statistically significant (prob<alpha) or not [`boolean`]
prob: p-value (`probability`)
stat (`structure`)
.prob_min minimum p-value with identified parameters. (`probability`)
.param: parameters (**g_0,i,alpha**)
.order_stat :i-th order statistic (`probability`)

Input pm_est format

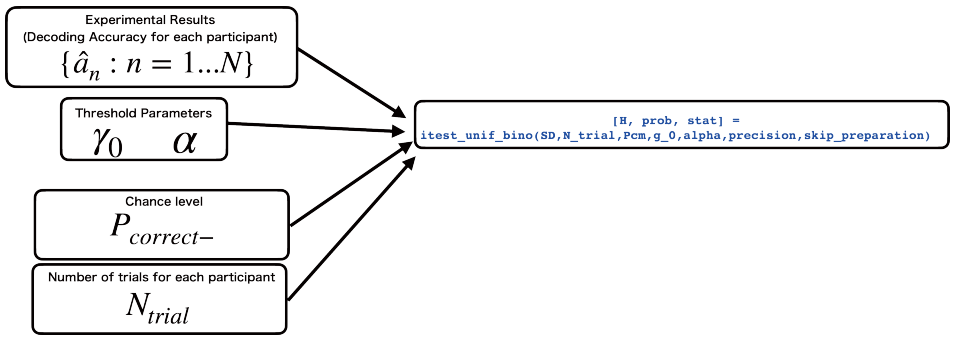
The decoding accuracy distribution under the null hypothesis "there is no label information in the brain."
The distribution can be specified with `pmf1 style` or `vector of probability` (list of results of empirical estimation).

1) `pmf1 style`
2 x [number of possible decoding accuracies] matrix,
Example: If there's 10 trials of binary decoding (chance = 50%) for each participants and if you assume that decoding accuracy follows binomial distribution,

```
pm_est = [0:0.1:1; binopdf(0:10,10,0.5)]
```

2) `vector of probability`
1 x [Number of repetition] matrix
Example: After permutation test with 1,000 repetition for each of 10 participants,
pm_est is 1 x 10,000 (10*1,000) vector.

- *i*-test-unif-bino (implementation/itest_unif_bino.m)



Usage

```
[H, prob, stat] =
itest_unif_bino(SD,N_trial,Pcm,g_0,alpha,precision,skip_preparation)
```

Input

SD: Experimental results (D-Acc) (`row vector of probability`)
N_trial: Number of trials for each participant (`integer`)
Pcm: Probability of the correct decoding in a trial for a participant without label information (= chance level) (`probability`)
g_0: Prevalence threshold (`probability`; default 0.5)
alpha: Statistical threshold (`probability`; default 0.05)
precision: Precision parameter for optimize i (`probability`; default 0.01)
skip_preparation:Skip input parser if true (`boolean`; default False)

Output

H: Whether result is statistically significant (prob<alpha) or not [`boolean`]
prob: p-value (`probability`)
stat (`structure`)
.prob_min minimum p-value with identified parameters. (`probability`)
.param: parameters (**g_0,i,alpha**)
.order_stat :i-th order statistic (`probability`)

Assumptions

1) Uniform prior distribution of γ and $P_{correct+}$.
2) True D-Acc with label information ($p(a_n \mid n \in \Omega_+)$) follows binomial distribution.
3) True D-Acc without label information ($p(a_n \mid n \in \Omega\_)$) follows binomial distribution.
4) Estimate that D-Acc without label information ($\tilde{p}(a_n \mid n \in \Omega\_)$) follows binomial distribution.

Note

The parameter i is optimized to maximize the expectation of the statistical power ($P(Significant \mid H_1)$) under the above assumptions. Smaller `precision` leads more precise optimization but requires time consuming calculation.

Pseudo-code

Input:

Experimental results: $\{a_n \; for \; n = 1 \ldots N\}$,

Number of trials: N_{trial}

Expectation of D-Acc without label information: $P_{\text{correct}-}$

Two predetermined threshold parameters: α and γ_0

Precision parameter: h

(find $i_{\max}$)

Set N as the number of the experimental results a_n

Set $i_{\max}$ as the largest i satisfies $BCDF(i-1, N, 1-\gamma_0) < \alpha$ in $i = 1,2,\ldots N$

(Find $i_{\text{unif-bino}}$)

Set $i_{\text{unif-bino}}$ as the value of i that maximize $Power(i)$ in $i = 1,2,\ldots i_{\max}$

where $Power(i)$ for each i is calculated as follows

(find T with fixed i)

Set T as the largest $a_{(i)}$ satisfies $L < \alpha$ in $a_{(i)} = 0,\ 1/N_{\text{trial}},\ 2/N_{\text{trial}} \ldots N_{\text{trial}}/N_{\text{trial}}$

where $Q = (1-\gamma_0)BCDF(\lfloor \hat{a}_{(i_{\text{unif-bino}})} \times N_{\text{trial}} \rfloor^*, N_{\text{trial}}, P_{\text{correct}-})$

$L = BCDF(i-1, N, Q)$

(Calculate marginal statistical power with fixed i)

Set $Power(i)$ as sum of $P(a_{(i)} > T | \gamma, P_{\text{correct}+})$

for γ from $\gamma_0 + h$ to 1 with h step and $P_{\text{correct}+}$ from $P_{\text{correct}-} + h$ to 1 with h step

where $P(a_n \le T) = \gamma BCDF(T \times N_{\text{trial}}, N_{\text{trial}}, P_{\text{correct}+}) + (1-\gamma)BCDF(T \times N_{\text{trial}}, N_{\text{trial}}, P_{\text{correct}-})$

$P(a_{(i)} > T | \gamma, P_{\text{correct}+}) = BCDF(i-1, N, P(a_n \le T))$

(perform i-test with $i_{\text{unif-bino}}$)

Set $\hat{a}_{(i_{\text{unif-bino}})}$ as the $i_{\text{unif-bino}}$-th order statistic of the experimental results $\{a_n \; for \; n = 1 \ldots N\}$

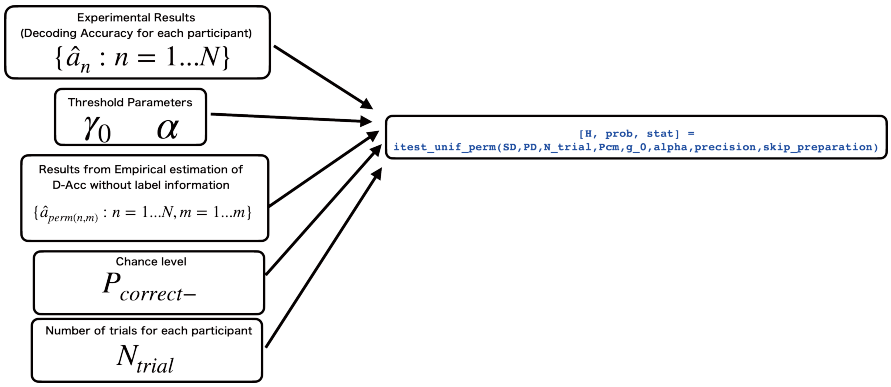
Set L with the following equations

$Q = (1-\gamma_0)BCDF(\lfloor \hat{a}_{(i_{\text{unif-bino}})} \times N_{\text{trial}} \rfloor^*, N_{\text{trial}}, P_{\text{correct}-})$

$L = BCDF(i-1, N, Q)$

If $\alpha > L$, i-test reports significant result

- i-test-unif-perm (implementation/itest_unif_perm.m)



Usage

```
[H, prob, stat] =
itest_unif_perm(SD,PD,N_trial,Pcm,g_0,alpha,precision,skip_preparation)
```

Input

SD: Experimental results (D-Acc) (`row vector of probability`)
PD: Results of the empirical estimation (e.g. permutation test) of D-Acc without label information (`matrix of probability`), with elements of each empirical estimation results.
N_trial: Number of trials for each participant (`integer`)
Pcm: Probability of the correct decoding in a trial for a participant without label information (= chance level) (`probability`)
g_0: Prevalence threshold (`probability`; default 0.5)
alpha: Statistical threshold (`probability`; default 0.05)
precision: Precision parameter for optimize i (`probability`; default 0.01)
skip_preparation:Skip input parser if true (`boolean`; default False)

Output

H: Whether result is statistically significant (prob<alpha) or not [`boolean`]
prob: p-value (`probability`)
stat (`structure`)
.prob_min minimum p-value with identified parameters. (`probability`)
.param: parameters (**g_0,i,alpha**)
.order_stat :i-th order statistic (`probability`)

Assumptions

1) Uniform prior distribution of γ and $P_{correct+}$.
2) True D-Acc with label information ($p(a_n \mid n \in \Omega_+)$) follows binomial distribution.
3) True D-Acc without label information ($p(a_n \mid n \in \Omega\_)$) follows the empirically estimated distribution.
4) Empirically estimate D-Acc without label information ($\tilde{p}(a_n \mid n \in \Omega\_)$).

Note

The parameter i is optimized to maximize the expectation of the statistical power ($P(Significant \mid H_1)$) under the above assumptions. Smaller `precision` leads more precise optimization but requires time consuming calculation.

Pseudo-code

Input:

Experimental results: $\{a_n \; for \; n = 1 \ldots N\}$,

Number of trials: N_{trial}

Expectation of D-Acc without label information: $P_{\text{correct}-}$

Two predetermined threshold parameters: α and γ_0

Precision parameter: h

Permutation results: $\{\hat{a}_{perm(n,m)} \; for \; n = 1 \ldots N, m = 1 \ldots M\}$

(find $i_{\max}$)

  Set N as the number of the experimental results a_n

  Set $i_{\max}$ as the largest i satisfies $BCDF(i-1, N, 1-\gamma_0) < \alpha$ in $i = 1,2, \ldots N$

(Find $i_{\text{unif}-\text{perm}}$)

  Set $i_{\text{unif}-\text{perm}}$ as the value of i that maximize $Power(i)$ in $i = 1,2, \ldots i_{\max}$

    where $Power(i)$ for each i is calculated as follows

    (find T with fixed i)

    Set T as the largest $a_{(i)}$ satisfies $L < \alpha$ in $a_{(i)} = 0, \; 1/N_{\text{trial}}, \; 2/N_{\text{trial}} \ldots N_{\text{trial}}/N_{\text{trial}}$

      where $Q = (1-\gamma_0)\frac{1}{M \times N}\sum_{n=1}^{N}\sum_{m=1}^{M}[\hat{a}_{\text{perm}(n,m)} < a_{(i)}]$

      $L = BCDF(i-1, N, Q)$

    (Calculate marginal statistical power with fixed i)

    Set $Power(i)$ as sum of $P(a_{(i)} > T | \gamma, P_{\text{correct}+})$

      for γ from $\gamma_0 + h$ to 1 with h step and $P_{\text{correct}+}$ from $P_{\text{correct}-} + h$ to 1 with h step

        where $P(a_n \le T) = \gamma BCDF(T \times N_{\text{trial}}, N_{\text{trial}}, P_{\text{correct}+}) + (1-\gamma)\frac{1}{M \times N}\sum_{n=1}^{N}\sum_{m=1}^{M}[\hat{a}_{\text{perm}(n,m)} \le T]$

        $P(a_{(i)} > T | \gamma, P_{\text{correct}+}) = BCDF(i-1, N, P(a_n \le T))$

(perform i-test with $i_{\text{unif}-\text{perm}}$)

  Set $\hat{a}_{(i_{\text{unif}-\text{perm}})}$ as the $i_{\text{unif}-\text{perm}}$-th order statistic of the experimental results $\{a_n \; for \; n = 1 \ldots N\}$

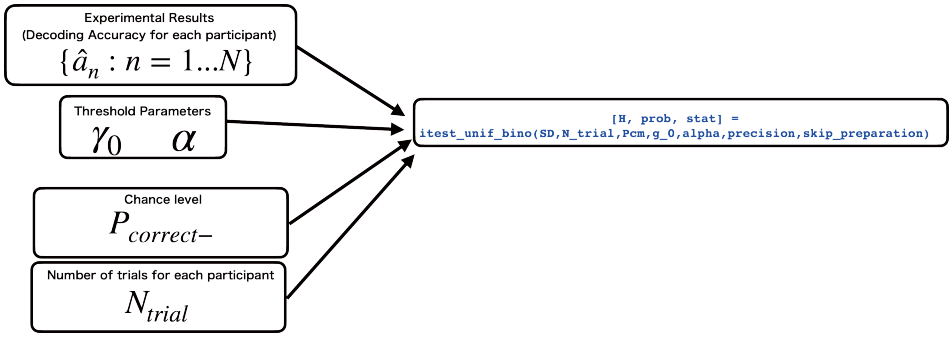
  Set L with the following equations

    $Q = (1-\gamma_0)\frac{1}{M \times N}\sum_{n=1}^{N}\sum_{m=1}^{M}[\hat{a}_{\text{perm}(n,m)} < \hat{a}_{(i)}]$

    $L = BCDF(i-1, N, Q)$

If $\alpha > L$, i-test reports significant result

- *i*-test-ml-bino (implementation/itest_ml_bino.m)



Usage

```
[H, prob, stat] =
itest_ml_bino(SD,N_trial,Pcm,g_0,alpha,precision,skip_preparation)
```

Input

SD: Experimental results (D-Acc) (`row vector of probability`)
N_trial: Number of trials for each participant (`integer`)
Pcm: Probability of the correct decoding in a trial for a participant without label information (= chance level) (`probability`)
g_0: Prevalence threshold (`probability`; default 0.5)
alpha: Statistical threshold (`probability`; default 0.05)
precision: Precision parameter for optimize i (`probability`; default 0.01)
skip_preparation:Skip input parser if true (`boolean`; default False)

Output

H: Whether result is statistically significant (prob<alpha) or not [`boolean`]
prob: p-value (`probability`)
stat (`structure`)
.prob_min minimum p-value with identified parameters. (`probability`)
.param: parameters (**g_0,i,alpha**)
.order_stat :i-th order statistic (`probability`)

Assumptions

1) Dirac delta prior distribution (maximum likelihood (ML) estimation) of γ and $P_{correct+}$.
2) True D-Acc with label information ($p(a_n \mid n \in \Omega_+)$) follows binomial distribution.
3) True D-Acc without label information ($p(a_n \mid n \in \Omega_-)$) follows binomial distribution.
4) Estimate that D-Acc without label information ($\tilde{p}(a_n \mid n \in \Omega_-)$) follows binomial distribution.

Note

The parameter i is optimized to maximize the expectation of the statistical power ($P(Significant \mid H_1)$) under the above assumptions. The ML estimation of γ and $P_{correct+}$ is first calculated with the dataset (`SD`). Then, parameter i is optimized. Smaller `precision` leads more precise but time consuming calculation.

Pseudo-code

Input:

Experimental results: $\{\hat{a}_n \; for \; n = 1 \ldots N\}$,

Number of trials: N_{trial}

Expectation of D-Acc without label information: $P_{\text{correct}-}$

Two predetermined threshold parameters: α and γ_0

Precision parameter: h

(find $i_{\max}$)

Set N as the number of the experimental results a_n

Set $i_{\max}$ as the largest i satisfies $BCDF(i-1, N, 1-\gamma_0) < \alpha$ in $i = 1,2,\ldots N$

(Find $i_{\text{ml-bino}}$)

Set γ_{ML} and $P_{\text{correct+ML}}$ as the values of γ and $P_{\text{correct+}}$ that maximize $Likelihood(\gamma, P_{\text{correct+}})$

for γ from $\gamma_0 + h$ to 1 with h step and $P_{\text{correct+}}$ from $P_{\text{correct}-} + h$ to 1 with h step

where $Likelihood(\gamma, P_{\text{correct+}}) = \prod_{n=1}^{N}\left((1-\gamma)\text{BPDF}(\lfloor \hat{a}_n \times N_{\text{trial}} \rfloor, N_{\text{trial}}, P_{\text{correct}-}) + \gamma \, \text{BPDF}(\lfloor \hat{a}_n \times N_{\text{trial}} \rfloor, N_{\text{trial}}, P_{\text{correct+}})\right)$

Set $i_{\text{ml-bino}}$ as the value of i that maximize $Power(i)$ in $i = 1,2,\ldots i_{\max}$

where $Power(i)$ for each i is calculated as follows

(find T with fixed i)

Set T as the largest $a_{(i)}$ satisfies $L < \alpha$ in $a_{(i)} = 0, \; 1/N_{\text{trial}}, \; 2/N_{\text{trial}} \ldots N_{\text{trial}}/N_{\text{trial}}$

where $Q = (1-\gamma_0)BCDF(\lfloor a_{(i)} \times N_{\text{trial}} \rfloor^{*}, N_{\text{trial}}, P_{\text{correct}-})$

$L = BCDF(i-1, N, Q)$

(Calculate marginal statistical power with fixed i)

Set $Power(i)$ as $P(a_{(i)} > T | \gamma_{\text{ML}}, P_{\text{correct+ML}})$

where $P(a_n \le T) = \gamma_{\text{ML}} BCDF\left(T \times N_{\text{trial}}, N_{\text{trial}}, P_{\text{correct+ML}}\right) + (1-\gamma_{\text{ML}})BCDF(T \times N_{\text{trial}}, N_{\text{trial}}, P_{\text{correct}-})$

$P(a_{(i)} > T | \gamma_{\text{ML}}, P_{\text{correct+ML}}) = BCDF(i-1, N, P(a_n \le T))$

(perform i-test with $i_{\text{ml-bino}}$)

Set $\hat{a}_{(i_{\text{ml-bino}})}$ as the $i_{\text{ml-bino}}$-th order statistic of the experimental results $\{\hat{a}_n \; for \; n = 1 \ldots N\}$

Set L with the following equations

$Q = (1-\gamma_0)BCDF(\lfloor \hat{a}_{(i_{\text{ml-bino}})} \times N_{\text{trial}} \rfloor^{*}, N_{\text{trial}}, P_{\text{correct}-})$

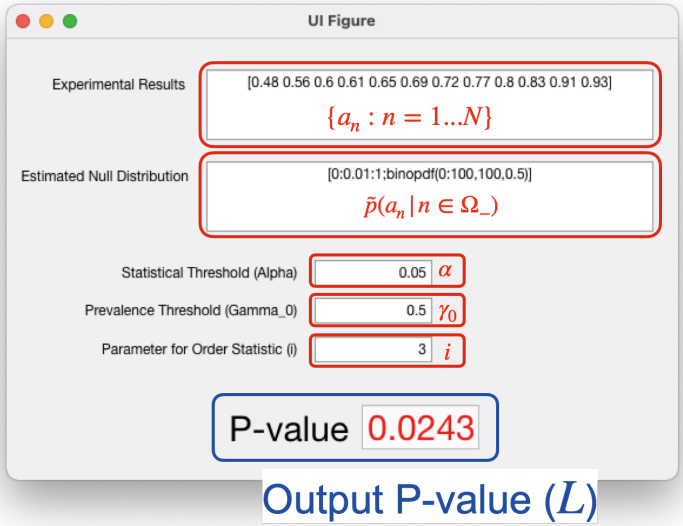
$L = BCDF(i_{\text{ml-bino}} - 1, N, Q)$

If $\alpha > L$, i-test reports significant result

•GUI

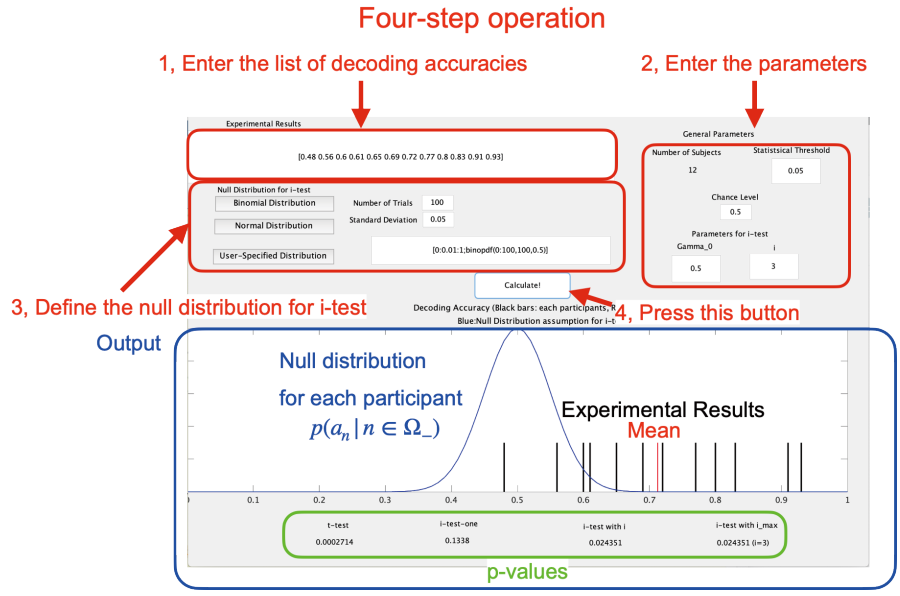
- itest_GUI (GUI/itest_GUI.mlapp)

  GUI interface to run itest.m.



- DEMO/Demo_Comparison.m, DEMO/Demo_Comparison.fig

  Comparison between the results from i-test (with assumption of binomial distribution) and Student t-test from the experimental result.



•Replication of the study

You can replicate all the analyses in the study with the following steps.

- Computation

A. Download the experimental result from (Empirical_Experiment_Results.mat),

B. Complete all the benchmark test, numerical calculation, simulation with the assumption of the binomial distribution, and (first and second level) simulation with the cross validated decoding accuracy.

This is done with the codes in the **Replication_of_study/Computation** directory and the results will be stored in the **Replication_of_study/Results** directory.

Caution!!: This is really, really time-consuming step. It requires > 1 month in total with a high-spec PC (~50GB system memory is required). The fastest workaround is to create **Replication_of_study/Results** directory, download the results from my release (https://github.com/satoshi-hirose/i-test/releases), and put them to the directory.

If you have multiple MATLAB licensed computers and you surely want to run the calculation, you can use codes for parallel computing with codes in **Replication_of_study/Computation/parallel_computing** directory to reduce the computational time. (NOTE: There's no parallel computing code for benchmark test).

> —How to use parallel computing—
>
> For each of (Numerical calculation, Simulation with binomial distribution, First level simulation with cross validation, Second level simulation with cross validation), do the following steps at **Replication_of_study/Computation/parallel_computing** directory.
>
> 1, Open master.m, run the 3 lines below the "`%% before the calcualtion of slaves,`" to initialize the parallel process, e.g.
>
> ```
> mkdir Numerical_Calculation_Computation
> my_number = 0;
> save Numerical_Calculation_Computation/number my_number
> ```
>
> 2, Run the **XX_slave.m** with as many computers as available.
>
> E.g. Open MATLAB with -r option (-r "cd('/[your itest directory]/Replication_of_study/Computation/parallel_computing'); Simulation_Binomial_Computation_slave"
>
> 3. After all the computation completed, open master.m and run the lines below the "`%% after the calcualtion of slaves`", to combine the results.

- Replicate the results

A. Confirm that there're the following 9 mat files in "Replication_of_study/Results" directory, after completed the computation step.

1,Benchmark_Test.mat
2,Empirical_Experiment_Results.mat
3,Numerical_Caliculation_Computation_g0_1.mat
4,Numerical_Caliculation_Computation_g0_3.mat
5,Numerical_Caliculation_Computation_g0_5.mat
6.Numerical_Caliculation_Computation_g0_7.mat
7,Simulation_Binomial_Computation.mat
8.Simulation_CV_Computation.mat
9,Simulation_CV_firstlevel.mat

B. Run the **Replication_of_study/Replication_of_all_resutls_in_paper.m** and you will find the replication of the figures on the paper. Also, in the command line, you will

see the values and descriptions reported in the text body of the paper. If you want to see each components, check the codes in **Replication_of_study** directory.

1,Sec3_Numerical_Calculation_Analysis.m
2,Sec4_Empirical_Data_Analysis.m
3,AppA_P_sig_calculation.m
4,Sup1_1_Simulation_Binomial_Analysis.m
5,Sup1_2_1_Simulation_CV_firstlevel_Analysis.m
6,Sup1_2_2_Simulation_CV_Analysis.m
7,Sup2_Numerical_Calculation_different_g0.m

•Sub-functions for i-test

- Calculate the maximum available i (itest/calc_imax.m)

Usage

```
[i_max,prob_min] = calc_imax(N,g_0,alpha)
```

Input

N: Number of subjects (`integer`)
g_0: Prevalence threshold (`probability`)
alpha: Statistical threshold (`probability`)

Output

i_max: maximum available i (`integer`)
prob_min minimum p-value with **i_max** (`probability`)

Descriptions

The parameters should satisfy the inequation $\alpha > BCDF(i-1,N,1-\gamma_0)$, otherwise i-test never report the significant results. This is because the p-value is bounded below with $BCDF(i-1,N,1-\gamma)$ (=**prob_min**).
From above inequation, we find the upper limit of i (**i_max**) with the specified inputs.

- Calculate the probability that itest report significant result (itest/calc_Psig.m)

Usage

```
[P_sig,param] =
calc_Psig(N,g_0,alpha,i,gamma,pp_true,pm_true,pm_est,skip_preparation)
```

Input

N: Number of subjects (`integer`)
g_0: Prevalence threshold (`probability; default 0.5`)
alpha: Statistical threshold (`probability; default 0.05`)
i: rank of order statistic(`vector of integer`, or 'all' (1:i_max))
gamma: list of true prevalence threshold (`vector of probability`)
pp_true: PMF of decoding accuracy with label information (`pmf style`)
pm_true: true PMF of decoding accuracy without label information. (`pmf1 style`)
pm_est: estimated PMF of decoding accuracy without label information (`pmf1 style`)
skip_preparation:Skip input parser if true [`boolean`; default False]

Output

P_sig: Probability that i-test reports significant resut. (3-dimensional ([number of **i**] x [number of gamma] x [number of pp_true]) `matrix of probability`)

param: parameters [structure] (**N, g_0, alpha, i, gamma, pp_true, pm_true, pm_est**)

Descriptions

The probability that itest report significant result is calculated with the parameters (**N**, **g_0**, **alpha**, **I**), the true distribution of D-Acc (**gamma, pp_true, pm_true**), and the estimated distribution of D-Acc without label information (**pm_est).** Multiple possibilities of **i, gamma, pp_true (**underlined inputs above**).** Can be specified. Output
Example:

```
I=1:3;
gamma=[0.6 0.7 0.8 0.9 1]
pp_true     =  [0      0.5      1        % 0:1/2:1
                0.25   0.5      0.25     %binopdf(0:2,2,0.5)
                0.16   0.48     0.36     %binopdf(0:2,2,0.6)
                0.09   0.42     0.49     %binopdf(0:2,2,0.7)
                0.04   0.32     0.64     %binopdf(0:2,2,0.8)
                0.01   0.18     0.81     %binopdf(0:2,2,0.9)
                0      0        1       ]%binopdf(0:2,2,1)
```

In this example, output **P_sig** will be 3 x 5 x 6 matrix.

- Binomial distribution version of calc_Psig (itest/calc_Psig_bino.m)

Usage

```
[P_sig,param] =
calc_Psig_bino(N,g_0,alpha,i,gamma,N_trial,Pcm,Pcp,skip_preparation)
```

Input

N: Number of subjects (`integer`)
g_0: Prevalence threshold (`probability;` default 0.5)
alpha: Statistical threshold (`probability;` default 0.05)
i: rank of order statistic(`vector of integer`, or 'all' (1:i_max))
gamma: list of true prevalence threshold (`vector of probability`)
N_trial: Number of trials (`integer`)
Pcm: Probability of the correct decoding in a trial for a participant without label information (= chance level) (`probability`)
Pcp: probability of the correct decoding in a trial for a participant with label information (vector of `probability`)
skip_preparation:Skip input parser if true [`boolean`; default False]

Output

P_sig: Probability that i-test reports significant result. (3-dimensional ([number of **i**] x [number of **gamma**] x [number of **Pcp**]) `matrix of probability`)

param: parameters [structure] (**N, g_0, alpha, i, gamma, N_trial, Pcm, Pcp**)

Descriptions

Under the assumption of the binomial distribution, the probability that itest report significant result can be numerically calculated with the parameters (**N, g_0, alpha, i**), the true distribution of D-Acc (**gamma, N_trial, Pcm, Pcp**). The correct estimation of the PMF without label information is also assumed (**pm_true = pm_est**). **pp_true** and **pm_true** is derived from **N_trial, Pcm, Pcp**. Multiple possibilities of **i, gamma, Pcp** can be specified.

Example:

```
i=1:3;
gamma=[0.6 0.7 0.8 0.9 1]
Pcp = 0.5:0.1:1
N_trial = 2
```

This example becomes identical to the above example of calc_Psig.

- Calculate the optimal i that maximize the expected statistical power (itest/calc_i_opt_wsum.m)

Usage

[i_opt,P_sig_w_sum] = calc_i_opt_wsum(P_sig,w)

Input

P_sig: Probability that i-test reports significant resut. (3-dimensional ([number of **i**] x [number of gamma] x [number of pp_true]) `matrix of probability`)
w: weight for **g** and **Pcp** ([Number of gamma] x [Number of Pcp] matrix of `probability`)

Output

i_opt: Index of the optimal i (`integer`)
P_sig_w_sum: weighted sum of the likelihood for each candidate **i** (vector of `probability`).

Descriptions

From output P_sig of calc_P_sig (or calc_P_sig_bino), the optimal i that maximize weighted sum of P_sig is calculated.

- Calculate likelihood of γ and $P_{correct|}$ under the binomial assumption (itest/calc_likelihood.m)

Usage

```
[Likelihood,g_mat,Pcp_mat] = calc_likelihood(SD,N_trial,Pcm,g_0,precision)
```

Input

SD: Experimental results (D-Acc) (`column vector of probability`)
N_trial: Number of trials for each participant (`integer`)
Pcm: Probability of the correct decoding in a trial for a participant without label information (= chance level) (`probability`)
g_0: Prevalence threshold (`probability;` default 0.5)
alpha: Statistical threshold (`probability;` default 0.05)

Output

Likelihood: Joint likelihood of the **Pcp** and **gamma**. (`2-dimensional matrix of probability`)
g_mat: matrix of **g**. (`2-dimensional matrix of probability`)
Pcp_mat: matrix of **Pcp**. (`2-dimensional matrix of probability`)

Descriptions

Calculate likelihood under the assumption of the binomial distribution, for combination of **gamma = (g_0+precision):precision:1** and **Pcp = (Pcm+precision):precision:1**. Each output is [number of possible **Pcp**] x [number of possible **gamma]** matrix, where **Pcp = Pcm+precision):precision:1**, and **gamma = (g_0+precision):precision:1.**

Example:

```
Pcm = 0.5
g_0 = 0.75
Precision = 0.25
```

Then,

```
g_mat = [0.75 1
         0.75 1
         0.75 1]
Pcp_mat = [0.5  0.5
           0.75 0.75
           1    1]
```

If,

```
Likelihood = [0.3  0.2
              0.6  0.4
              0.3  0.1]
```

The joint likelihood of **gamma = 0.75** and **Pcp = 0.5** is 0.3.

•Sub-functions for input parser

- Check if x is a probability (i.e. real number between 0 and 1)

```
y = isprob(x)
```

- Check if x is a positive integer

```
y = isposint(x)
```

- Check if p follows `pmf style`

```
y = isp(p,num_p)
```

To check if p follows `pmf1 style`, use **num_p=1.**