Speeding up Permutation Testing

Vamsi Ithapu

http://pages.cs.wisc.edu/~vamsi/pt_fast

November 17, 2013

The paper

- "Speeding up Permutation Testing in Neuroimaging"
- Joint work with Chris Hinrichs [1], Vikas Singh and Qinyuan Sun
- NIPS 2013 Spotlight

Basic Idea : *Traditional permutation testing procedure is computationally intensive. Our model leverages the structure of permutation testing matrix, and reduces the computation time by atleast 50 times without loosing any accuracy in estimating the null distribution.*

---

[1] Vamsi and Chris are joint first authors

Background

Consider a study with n subjects from two groups (ex. Diabetic vs. Non-diabetic). For each subject, a m dimensional data/measurement is obtained (voxels, ROIs, genes etc.).

Multiple hypothesis testing checks for group difference by

- Computing m univariate hypothesis tests (ex. t test)
- Calculating the corrected p–value by adjusting for multiple testing issues

Bonferroni method computes the corrected α threshold using union bound (i.e. averaging over m tests).

Problem : If m is large, Bonferroni's corrected $\alpha \ll$ true α

Permutation Testing - Background/Setup

Permutation testing is a random sampling method – a non–parametric method to estimate the FWER by sampling from Global/Max Null distribution.

If the two groups donot differ, then I can permute the group/class labels and end up with approximately same set of t statistics

Given m, n and T (numner of trials/permutations). Repeat T times

- Randomly "permute" group labels across n subjects – compute t statistics for m dimensions

– $m \times T$ permutation testing matrix (denoted by P).

Compute the max. t statistics for each permutation (column of P), and estimate the max. Null distribution

Compute p-value of "true" labeling using max. Null

Permutation Testing - continued

For a good estimate of max. Null, T should be very large.

Depending on m, n and T (number of random permutations), permutation testing is extremely computationally intensive.

- In neuroimaging, typically $m \sim 3 \times 10^5$, $n \sim 400$ and $T \sim 10^4$
- In Bioinformatics, typically $m \sim 1000$, $n \sim 10^3$ and $T \sim 10^3$

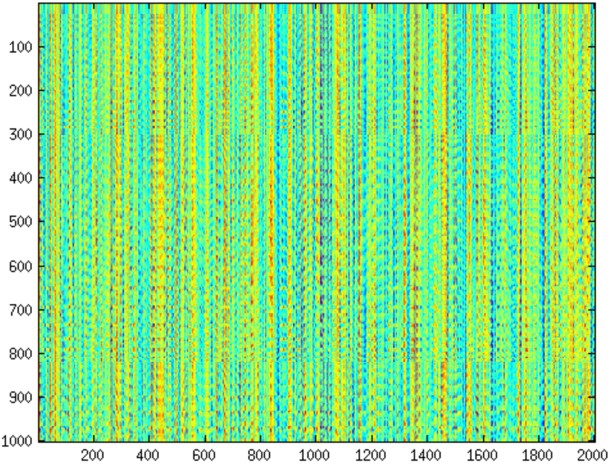
The computation time can be days, and weeks in some cases!!

Observation:

- P is "highly structured" – a combination of low–rank signal and high–rank residual.

example P

MRI data. 100 healthy vs. non-healthy. $m = 1000$, $T = 2000$



So what?

From a high–level viewpoint, this means

P is “highly structured”

$\Longrightarrow$ Each column looks “similar” to other columns, and each row looks “similar” to other rows

$\Longrightarrow$ If you give me “sufficiently many” random (i.e. at random positions) entries of P, I will give you a highly accurate estimate of the entire matrix P

Mathematically, $P = UW + S$, U is low rank and S is random residual – given some entries, I can estimate U, W and S (Matrix Completion)

Sufficiently many $\sim < 1\%$ sub–sampling !!

Evaluations Setup

Data

- MRI data from 4 studies of cognitively healthy vs. non-healthy subjects
- $n = 40, 50, 55$ and 70
- $m \sim 275000$ and $T = 10^4$

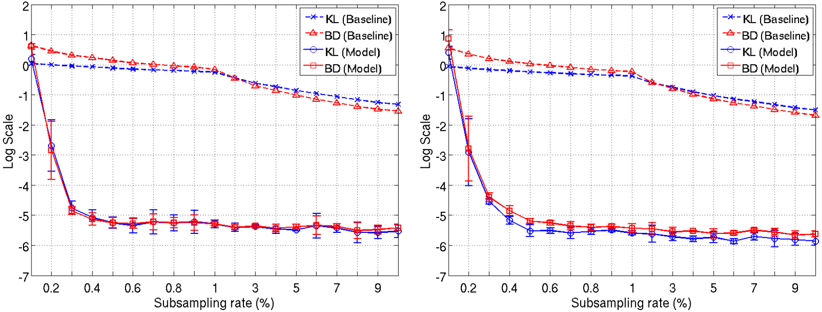
Questions

- Can we recover max. Null ?
- What is the computational speed-up ?
- How stable is the estimated α threshold ?

Baseline computes max Null from sub-sampled data directly (i.e. no completion of P)

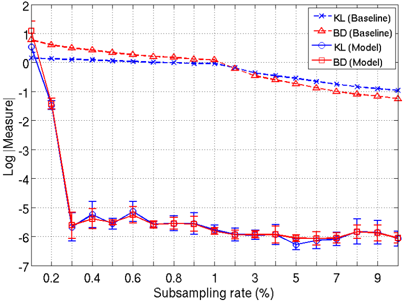
max Null recovery

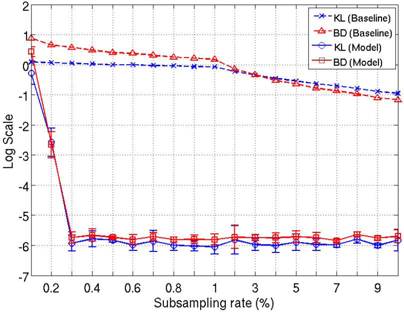
Recovery measued using D_{KL} (KL Divergence) and D_B (Bhattacharya Distance) in log–scale



max Null recovery

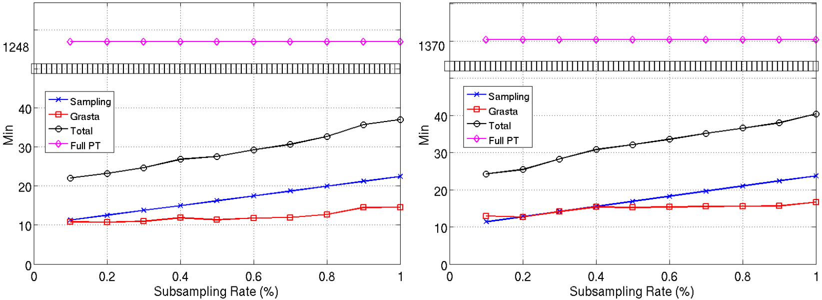
Recovery measued using D_{KL} (KL Divergence) and D_B (Bhattacharya Distance) in log–scale





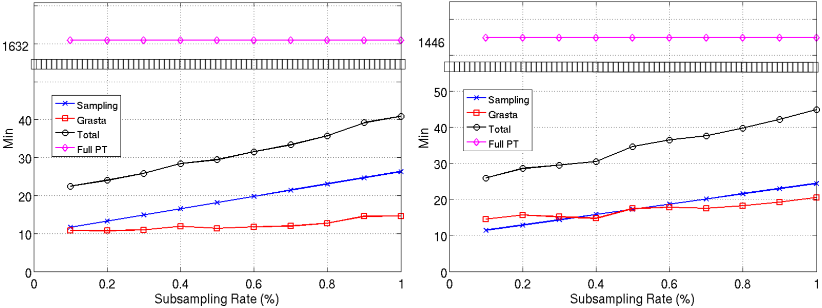
Computational Speed-up

Time measured in minutes.



Computational Speed-up

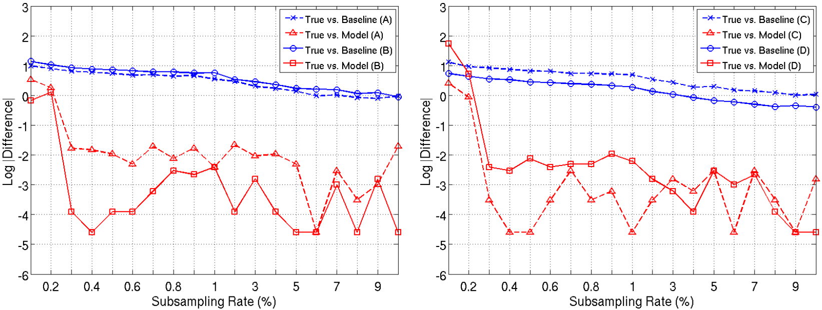
Time measured in minutes.



recovery vs. speed-up

Stability of α thresholds

t-statistic thresholds at $\alpha = 0.95$



Conclusion

- A novel method for estimating permutation testing matrix is proposed
- A computationl speed-up of > 50 is achieved while recovering max. Null upto a high degree of accuracy