**Hello everyone, I am Yao and today my presented topic is focus on a methodology paper and its simulations: Penalized GEE for high dimensional longitudinal data analysis.**

**Introduction**

This is the flow of today’s presentation. We will present a thorough overview of the method and its theoretical theorems. If you find it interesting, or helpful with your research, you are welcome to discuss the details in QA session and after class.

Penalized generalized estimating equations, or Penalized-GEE, is an innovative statistical approach to analyze high-dimensional longitudinal data.

Traditional GEE models may not be able to handle the large number of covariates present in high-dimensional data, but the addition of penalty functions in Penalized-GEE can help address this issue.

These penalty functions are mathematical functions that shrink the coefficients of irrelevant covariates, leading to more accurate and efficient results.

The theoretical properties of P-GEE are also impressive, as they are consistent and asymptotically normal under certain conditions.

To evaluate the effectiveness of P-GEE, Monte Carlo simulations and real-world dataset applications have been conducted. These studies have shown that P-GEE performs well in identifying relevant covariates and improving prediction accuracy.

**Challenges**

* Longitudinal data analysis poses several challenges, especially when dealing with high-dimensional data. This type of data involves repeated measurements on a large number of covariates over time, with the number of variables much larger than the number of observations.
* And it is commonly seen in Large-scale long-term health studies, gene expression experiments…
* Traditional Generalized Estimating Equations, or GEE, may struggle in this setting with variable selection and parameter estimation due to the large number of covariates compared to the number of observations. This can lead to Overfitting: GEE models may overfit the high-dimensional data by including too many irrelevant covariates, which can result in poor generalization and low prediction accuracy.

**Review and Take home message about GEE**

* Before introducing the penalized GEE method, let’s review the generalized estimating equations and here are some take-home messages:
* GEE, is a statistical method used to analyze correlated data from longitudinal studies or clustered data.
* GEE estimates the population-average or marginal effect of the predictors on the outcome variable, rather than the subject-specific effect.
* To account for the correlation of within-subject data, GEE specifies a working correlation matrix structure, such as independence, AR(1), or exchangeable.
* However, misspecification can be problematic and affect the efficiency of the parameter estimates. To fix this, GEE can be used with the Huber-White “sandwich estimator” for robustness.
* Unlike likelihood-based methods, GEE is a quasi-likelihood method, which means only the first two moments, the mean and the covariance, matter. However, there is no readily available goodness-of-fit measure for GEE, making model selection unclear.
* Overall, GEE is a powerful tool for analyzing longitudinal data.

\\

**GEE with Diverging Number of Covariates**

Dr. Wang, the first and corresponding author of this methodology paper, also developed the asymptotic theory for GEE analysis when the number of covariates grows to infinity with the number of clusters.

This also provided us theoretical background for dimension reduction by variable selection when we work with high-dimensional longitudinal data

**Penalized GEE - overview**

The Penalized GEE approach adds penalty functions to the GEE model, shrinking the coefficients of irrelevant covariates and producing more accurate and efficient results.

Overall, P-GEE is a powerful tool that can help researchers better understand the dynamics of high-dimensional longitudinal data, and has the potential to advance research in fields such as genetics, epidemiology, and biomedical sciences.

**Penalized GEE – algorithm**

**Asymptotic theorem   
Monte carlo simulation – design**

* Three models : Penalized GEE (PGEE) with all zeros as initial beta values, unpenalized GEE with all zeros as initial beta values and the oracle GEE, which is the ideal model with the true beta values
* For each model, we also applied three working correlations structures: Independence, exchangeable and AR(1)
* First we need to select the tunning parameter lambda\_n in penalty function by a forth fold cross validation. Note that this step actually took a long time to process, and my simulation only calculated the accuracy of 0.1, not 0.01 due to the limited time.
* Set up & estimation

**Simulation 1 – correlated normal responses**

The correlated normal responses are generated from this model, with 200 subjects, 4 timepoints, and 200 covariates.

we generate xij from the multivariate normal distribution with mean 0 and an AR(1) covariance matrix with marginal variance 1 and auto-correlation coefficient 0.5.

The random errors are also generated from the multivariate normal distribution with marginal mean 0, marginal variance 1 and an exchangeable correlation matrix with parameter ρ.

We consider ρ = 0.5 and 0.8 to represent different strength of within cluster correlation

The results of Table on the right hand side, summarize the estimation accuracy and model selection properties of the penalized GEE, the unpenalized GEE and the oracle GEE for three different working correlation matrices and two different values of ρ.

We observe that in terms of estimation accuracy the penalized GEE procedure performs closely to the oracle GEE, and significantly reduces the MSE of the unpenalized GEE estimator.

Using the true correlation structure (exchangeable) in penalized GEE gives the smallest MSE, with greater gain when the within cluster association is stronger.

Furthermore, we observe that the unpenalized GEE generally does not lead to a sparse model. The penalized GEE successfully selects all covariates with nonzero coefficients and has a fairly small number of false positives.

**Monte carlo simulation – design**