Submitted to the Annals of Applied Statistics

SUPERVISED DISTANCE BASED MATCHING ALGORITHM TO EVALUATE THE EFFECT OF PUBLIC HEALTH INTERVENTIONS

BY HONGZHE ZHANG[1], JIASHENG SHI[1,*] AND JING HUANG[1,†]

[1] *Department of Biostatistics, Center for Clinical Epidemiology and Biostatistics, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, PA hongzhe.zhang@pennmedicine.upenn.edu;*
[*] *Jiasheng.Shi@Pennmedicine.upenn.edu;* [†] *jing14@pennmedicine.upenn.edu*

The abstract should summarize the contents of the paper. It should be clear, descriptive, self-explanatory and not longer than 200 words. It should also be suitable for publication in abstracting services. Formulas should be used as sparingly as possible within the abstract. The abstract should not make reference to results, bibliography or formulas in the body of the paper—it should be self-contained.

This is a sample input file. Comparing it with the output it generates can show you how to produce a simple document of your own.

1. Introduction.

Roaring cases of COVID-19 cases across the country due to omicron variant arouse much controversy on in-person education for children and adolescents. Reopening schools inevitably increases social interactions and contribute to the community transmissions of the disease. Policy makers must make the painful trade-off between teaching quality and COVID-19 virus prevention. An observational study that quantifies the increase of transmission rate due to school-reopening would help design related policies. However, such a study would subject to confounding factors caused by vastly different community factors. As a potential solution, "matching algorithms" select a subset of comparable observations in each treatment group and balance the distributions of (possibly confounding) baseline covariates therefore, remove the bias in estimated treatment effect. Most ideally, a re-opened county is matched to a closed county with the same community factors; but such exact matching is not feasible when the factors to be matched are more than just a few. The best we can do is to pair counties with "similar" baseline factors.

A distance measure describes how different two observations are in terms of their covariates. One can seamlessly pair an re-opened county with the "closest" controlled county. Popular distance measures include *Euclidean Distance* and *Mahalanobis Distance*. For M controlled, closed counties and N treated, re-opened counties each has p individual baseline characteristics stored separately in $\tilde{X}^c_{N\times p}$, $\tilde{X}^t_{M\times p}$. Denote i^{th} and j^{th} observations respectively from $\tilde{X}^c$, $\tilde{X}^t$ as $\tilde{x}^c_i$ and $\tilde{x}^t_j$, their *Euclidean Distance* and *Mahalanobis Distance* can be expressed as

$$(\tilde{x}^c_i - \tilde{x}^t_j)^T(\tilde{x}^c_i - \tilde{x}^t_j) \; and \; (\tilde{x}^c_i - \tilde{x}^t_j)^T S^{-1}(\tilde{x}^c_i - \tilde{x}^t_j) \tag{1}$$

where S is the sample covariance matrix. The *propensity score* which was proposed by Rosenbaum and Rubin (1983) as a substitution to distance function. One can control solely the propensity score to balance the overall distributions of all covariates in the control and treatment group. Assuming treatment assignments are independent, the propensity score for i^{th} can be written as

$$p(x_i) = P(i^{th} \; district \; is \; treated | X = x_i) \tag{2}$$

Keywords and phrases: First keyword, second keyword.

Rosenbaum and Rubin (1985) also proposed to use a logit transformation of $p(.)$ as the results oftentimes behave like a normal distribution. Since the propensity score is essentially a conditional probability of treatment assignment, it can be estimated by a logistic regression or similar classification algorithms.

However, existing distance measures used by matching algorithms weight all baseline covariates the same or simply weight them by their variances. Although this might be the best that we can do with no addition information, the best is not enough for large p. Contemporary observational studies suffers from the curse of dimensionality (large p). It dictates all observations to be at a similar distance from the others (Intro to high), and aforementioned distance measures fail as similarity measures.

One can improve the matching algorithms by using distance measures that "select" important confounders from all baseline covariates. In this paper, we consider a scenario where there are "domain experts" so the "bias contributions" of all baseline covariates are known to them Better to refer some studies like this. When the number of candidate counties gets large, the time required to manually match all candidates become prohibitive. Therefore, we try to learn the contributions a (small) set of observations that are already paired by the experts. More concretely, we would additionally have some paired observations from which we try to recover the experts' knowledge on the bias contribution for each baseline characteristics.

We do so by estimating a p-dimensional weight vector that stands for the "importance" of matching each covariates and accommodate this weight into traditional Euclidean distance-based method. We call this novel supervised, distance-based matching algorithm (A Name). (The name) obtains the weight from an optimization problem based on bias contribution learnt from matched pairs. A weighted Euclidean distance is then used to pair unmatched re-opened counties to the closest closed counties. By the setup, the number of matched pairs should be significantly less than the number of unmatched pairs. We adopt a self-taught learning framework to exploit the information in unmatched observations. The performance of the proposed algorithm is shown empirically based on simulation studies and (introduction to the real data).

2. Method. In this paper, we assume there are ℓ pairs of treatment and control subjects that are already matched by the experts. Each control and treatment observation again has p individual baseline characteristics. Without loss of generality, we place the ℓ control subjects in the first ℓ rows. In mathematical terms, we have $X_{2l \times p} = \begin{pmatrix} X^c \\ X^t \end{pmatrix}$ and both X^c and X^t are $\ell \times p$ matrices. Denote the i^{th} observation of X as x_i, we also let x_i to be paired with $x_{i+\ell}$ for $i = 1, ..., \ell$. We propose to learn a weighted Euclidean distance metric that ideally pairs the unmatched observations $\tilde{X}^c$, $\tilde{X}^t$ in the same way that the expert would have done.

2.1. *Distance Metric Learning.* Learning a distance metric with certain supervised knowledge has been explored and implemented in the computer science community by the name distance metric learning (DML). Its most prominent application is in the field of image retrieval, where algorithms were, for example, expected to retrieve the most similar images for an given image (Hoi et al., 2006; Bar-Hillel et al., 2005; Si et al., 2006).

Xing et al. consider a scenario when there are only two sets of observations which are known to be similar or dissimilar to each other. These two subsets can be written respectively as

$$S : (x_i, x_j) \text{ If } x_i, x_j \text{ are similar}$$
$$D : (x_i, x_j) \text{ If } x_i, x_j \text{ are dissimilar}$$

The target distance metric between two observations takes x_i, x_j the form of $d_A(x_i, x_j) = \sqrt{(x_i - x_j)^T A(x_i - x_j)}$. The key matrix A is estimated by solving

$$\begin{aligned} & min_A \sum_{x_i, x_j \in S} d_A(x_i, x_j) \\ & s.t. \sum_{x_i, x_j \in D} d_A(x_i, x_j) \geq 1, A \succeq 0 \end{aligned} \tag{3}$$

where $A \succeq 0$ means A is positive semi-definite. The objective function encourages A to place "similar" observations close each other while keeping dissimilar observations not "too close" to each other. The positive semi-definiteness constraint ensures that the metric $d_A(x_i, x_j)$ satisfies the non-negativity and the triangle inequality, while the lower bound on $\sum_{x_i, x_j \in D} d_A(x_i, x_j)$ makes sure A does not collapse all observations into a single point. This objective (3) generally needs to be solved by gradient descent and an application of iterative projections (Rockafellar, 2015). Si et al. pointed out that A learnt by (3) is not robust when training data $\{x_i\}_{i=1}^n$ are noisy or n is small. They chose to penalize the size of the matrix A and proposed an alternative objective function

$$\begin{aligned} & min_A \|A\|_F + C_S \sum_{x_i, x_j \in S} d_A(x_i, x_j) - C_D \sum_{x_i, x_j \in D} d_A(x_i, x_j) \\ & s.t. A \succeq 0 \end{aligned} \tag{4}$$

Hoi et al. (2010) replaces the penalization term $\|A\|_F$ by a "semi-supervised" term $L_A(x)$ that is computed on solely A and observations without any constraint $\{x : x_i \notin S \text{ and } x_i \notin D\}$. They also apply a different constraint on A instead of positive definiteness. The loss function is written as

$$\begin{aligned} & min_A \, L_A(x) + C_S \sum_{x_i, x_j \in S} d_A(x_i, x_j) - C_D \sum_{x_i, x_j \in D} d_A(x_i, x_j) \\ & s.t. \log(det(A)) > 0 \end{aligned} \tag{5}$$

In addition, Bar-Hillel et al. (2005) proposed a supervised distance measure based only on "similarity constraints". Their proposed algorithm is called relevant component analysis (RCA) which uses a distance metric of the same form as Xing et al. (2002), such that $d_C(x_i, x_j) = (x_i - x_j)^T \hat{C}^{-1}(x_i - x_j)$. Compared to Xing et al. (2002), RCA allows several (K) sets of similar observations each with n_k observations and estimates $\hat{C}^{-1}$ by $\frac{1}{K}\sum_{k=1}^K \sum_{i=1}^{n_k} (x_{ki} - m_k)^T (x_{ki} - m_k)$. The similar sets can be written as

$$S_k : (x_i, x_j) \text{ If } x_i, x_j \text{ are similar}, k = 1, \cdots, K$$

They claim that the matrix C would assign large weights to "relevant" variables that discriminates the K subsets of observations. Hoi et al. (2006) propose a discriminative component analysis (DCA) extend the RCA to take into account of dissimilarities between similarity groups. They define $\hat{C}_b$ as $\frac{1}{n_b}\sum_{k=1}^K \sum_{i \in D_j} (m_j - m_i)^T (m_j - m_i)$, where D_j includes the index of set that is dissimilar to set j. DCA outputs a distance metric $M = \hat{A}^T \hat{A}$ between $S_k, k = 1, \cdots, K$, and $\hat{A}$ is obtained by solving

$$max_A \frac{|A^T \hat{C}_b A|}{|A^T \hat{C} A|} \tag{6}$$

A range of distance metrics learnt from pairwise similarity conditions are designed for image data. These methods consider different and rather complex loss functions optimized by a convolutional neural networks. We refer the readers to a recent paper (Wang and Deng, 2021) for a complete survey of relevant methods.

On the other hand, unlike image retrieval, matching accuracy is not the sole property we ask from the distance metric when we are matching experiment units to evaluate the effect of public health intervention. We require both the matching procedure and the results itself should be interpretable. The DML algorithms are all based on Mahalanobis distance which is not interpretable as the similarities are measured in a transformed feature space. In addition, metrics such as DCA can only accommodate similarity / dissimilarity constraints between one partition of the data, which becomes problematic when a single observation appears in multiple constraints.

We propose an intuitive matching algorithm (name) that explicitly looks for a weight vector that simultaneously minimizes the weighted Euclidean distance between paired observations and maximizes the distance between the candidate observations that are not paired together. Weighted Euclidean distance is a more intuitive similarity measure as the scales of all variables are kept, and the estimated weight vector naturally provides a variable importance measure for each baseline characteristics. In addition, the solution to the algorithm is given by a closed form formula and avoids iterative optimization procedure like other algorithms. A detailed introduction to name is given in the next section.

2.2. *Proposed Method.*

2.2.1. *Optimization Problem Formulation.* In this paper, we assume the experts pair observations based on an unknown distance function $d(\cdot,\cdot)$. The i^{th} (controlled) observation x_i from X is paired with a $(i+l)^{th}$ (treated) observation x_j^t from X because

$$\arg\min_{j=1,\cdots,l} d(x_i, x_{l+j}) = i$$

It is also assumed that the distance function $d(\cdot,\cdot)$ is a squared weighted Euclidean distance, such that

$$d(x_i, x_j) = d_\beta(x_i, x_j) = \|\beta^T(x_i - x_j)\|_2^2$$

We can then learn the underlying distance function by learning the weight β from the training pairs. To recover pairings for the matched data points, β should firstly make the distance between any paired observations small. Define a indicator matrix $W_{ij}^w = \begin{cases} 1 & j = i+l \\ 0 & \text{otherwise} \end{cases}$ which equals 1 if two observations are paired, the sum of the distances between every pairs of matched observation can then be written as

$$\begin{aligned} Loss_w(\beta) &= \frac{1}{2}\Sigma_{i,j=1}^{2l}||\beta^T(x_i - x_j)||_2^2 W_{ij}^w \\ &= \beta^T X^T (D_w - W^w) X\beta \\ &= (X\beta)^T L_w X\beta \end{aligned} \tag{7}$$

where $D_w = diag(\sum_{j=1}^{2l} W_{ij}^w)$ and $L_w = D_w - W^w$ is know as the Laplacian matrix in the field of Graph Theory. We call (7) "within-pair loss". This name suggests an ideal β should make (7) as small as it is allowed. However, with only this objective, any β that assigns the same value of $f(\cdot)$ to matched data points achieve $Loss_w = 0$. It is then guaranteed for β to overfit the matched pairs and recover experts' decision poorly for $\tilde{X}^c$ to $\tilde{X}^t$. Therefore, we also consider the distances between observations in X^c and X^t that are not paired. Similarly, the sum of all such distances can be written as

$$\begin{aligned} Reward_b(\beta) &= \frac{1}{2}\Sigma_{i,j=1}^{2l}||\beta^T(x_i - x_j)||_2^2 W_{ij}^b \\ &= \beta^T X^T(D_b - W^b)X\beta \\ &= (X\beta)^T L_b X\beta \end{aligned} \tag{8}$$

where $W_{ij}^b = \begin{cases} 1 & j \neq i+l \;\&\; j \in \{l+1, ..., 2l\} \\ 0 & \text{otherwise} \end{cases}$ which equals 1 if two observations in different treatment groups are not paired, and $D_b = diag(\sum_{j=1}^{2l} W_{ij}^b)$ and $L_b = D_b - W^b$. We call (8) "between-pair reward". Since any β with a large size can achieve a large $Reward_b(\beta)$, we penalize the size of the β. We propose to maximize $g(\beta)$ expressed by (9).

$$\underset{\beta}{\text{maximize}} \; \frac{Reward_b(\beta)}{Loss_w(\beta)/\ell + Penal(\beta)} = g(\beta) \tag{9}$$

$Loss_w(\beta)$ is scaled by the number of pairs of training data only for numerical reason. Maximizing $g(\beta)$ obviously minimizes the loss term of the penalty terms and maximizes the reward term simultaneously.

2.2.2. *Optimization Problem Solution.* For reasons that will become clearer later, we use a two-norm penalization on the β. Adding a tuning parameter for the penalization on β, the objective function $g(\beta)$ can be re-written as

$$\frac{\beta^T X^T L_b X\beta}{\beta^T(X^T L_w X/\ell + \lambda\mathbf{I})\beta} \tag{10}$$

Define symmetric matrix $A = X^T L_w X/\ell + \lambda\mathbf{I}$, and decompose it as $\Gamma_A \Lambda_A \Gamma_A$, we have $A^{\frac{1}{2}} = \Gamma_A \Lambda_A^{\frac{1}{2}} \Gamma_A$ which is also symmetric. Write $\beta^T A\beta = \|A^{\frac{1}{2}}\beta\|^2$ and $\eta = \frac{A^{\frac{1}{2}}\beta}{\|A^{\frac{1}{2}}\beta\|}$, we have

$$\begin{aligned} g(\beta) &= \frac{\beta^T X^T L_b X\beta}{\beta^T(X^T L_w X/\ell + \lambda\mathbf{I})\beta} \\ &= \eta^T A^{-\frac{1}{2}} X^T L_b X A^{-\frac{1}{2}} \eta \\ &= \eta^T \Gamma\Lambda\Gamma\eta \;\; \text{With spectral decomposition } A^{-\frac{1}{2}} X^T L_b X A^{-\frac{1}{2}},\; \Gamma\Gamma^T = \mathbf{I} \\ &= \xi^T \Lambda \xi \;\; \text{and } \xi = \Gamma^T \eta \end{aligned} \tag{11}$$

whose maximum is obtained when $\xi = \begin{pmatrix} 1 \; 0 \cdots 0 \end{pmatrix}$. Hence, $\eta = \gamma_1$ and $\beta = A^{-\frac{1}{2}}\gamma_1$. This is the first eigenvector of $G(X) = A^{-1}X^T L_b X = (X^T L_w X/\ell + \lambda\mathbf{I})^{-1} X^T L_b X$. Note that only with a 2-norm penalization on the β can we get the neat closed-form solution.

2.3. *Asymptotic Property of β.* In this section, we look at how well an estimate based on training data can recover the underlying expert knowledge. Let's assume the experts are trained on a much larger data set $\tilde{X} = \begin{pmatrix}\tilde{X}_c^T \; \tilde{X}_t^T\end{pmatrix}^T, \tilde{X}_c = \begin{pmatrix}X_c^T \; \bar{X}_c^T\end{pmatrix}^T$ and $\tilde{X}_t = \begin{pmatrix}X_t^T \; \bar{X}_t^T\end{pmatrix}^T$.

Here X_c, X_t are the training data available to us, while $\bar{X}_c, \bar{X}_t$ are training data for experts that is unknown to us. Let's say there are ℓ_1 paris of observed training data and ℓ_2 paris of unobserved and $\ell = \ell_1 + \ell_2$. Since the purpose of the algorithm is to recover the expert knowledge, that is, the weight vector β. We define the weight β^* ontained from $\tilde{X}$ as the ground truth. Denote the estimated weight vector based on X as $\hat{\beta}$, the main results is given in **theorem 1**. Define a distance measure $dist(X, Z) = ||XX^T - ZZ^T||$, we have the following convergence results

THEOREM 2.1. *$dist(\hat{\beta}, \beta^*) = O_p(1/\sqrt{\ell_1})$ as $\ell_1, \ell \to \infty, \ell_1/\ell = C < 1$.*

The proof is given in the Appendix.

2.4. *A Self-taught Learning Framework.* The costly nature of matched data points guarantees the training data size 2ℓ to be much smaller than the size of unmatched data $m+n$. To make use of the (potential) additional information in unmatched data $\tilde{X}_c, \tilde{X}_t$, we adapt the self-taught learning framework (Raina et al., 2007). This semi-supervised procedure is given in **Algorithm 1**.

Algorithm 1: Self-taught Learning Procedure

```
Input: $\hat{\beta}$, $X$, $\tilde{X}_c$, $\tilde{X}_t$
for $i$ = 1, ..., MaxIte do
    Get the closest observations in $\tilde{X}^c$ for all $\tilde{X}^t$ with $\hat{\beta}$;
    Order the pairs with respect to within-pair distance, find the top $k$ pairs with the
      smallest within-pair distance;
    Add the $k$ pairs to $X$, delete the $k$ pairs from $\tilde{X}_c$, $\tilde{X}_t$;
    Obtain a new $\hat{\beta}$ from $X$;
end
Output: $\hat{\beta}$
```

Starting from the $\hat{\beta}$ estimated from original training data X, in each iteration, we add the most confidently predicted pairs into the training data X and repeat this process. The outputted β would be trained on a much larger set of matched pairs.

Since even the top K pairs are not guaranteed to be correctly paired like the original X, at each iterations of the algorithm, we are obtaining additional training data at the risk of introducing incorrectly paired data into X. This risk is precisely the probability of mismatch by the original algorithm. This suggests that we should use **Algorithm 1** with caution if the initial performance of the matching algorithm is inferior. On the other hand, if the initial performance is already close to perfect, it is also not wise to run the algorithm which risks eroding the original β by adding potentially incorrectly labelled paired to X in exchange for a chance of improving it slightly. Therefore, we argue that one should only run **Algorithm 1** if the initial performance is *mediocre* and the iterations should be stopped when its performance is adequate. We develop a simulation framework to help users determine the range of *mediocre* performance of name. The users would be able to adjust the simulation parameters to mimic their real data with the simulated data. More details will be given in the simulation section.

Besides setting a maximum number of iterations, we also recommend users to set a *Accuracy Cap*. As long as the held-out testing accuracy exceed this cap, the self-taught learning procedure should be stopped.

3. Results. We applied the name along with RCA, DCA, Euclidean Distance Matching and propensity score matching on synthetic datasets and real data small introduction. name is shown empirically to have higher matching accuracy than competing methods. We also investigate the robustness of its performance when the model assumptions are violated. At last, we demonstrated how to use simulation to access the potential performance of the self-taught learning framework.

Before looking at the results, we want to note that matching accuracy is a harsh criteria, and correctly matching observations among a large candidate pool is a hard problem. Compared with binary classification problems, a classifier that randomly classify all observations have an accuracy 0.5, and the probability for it to classify n observations wrong is $\frac{1}{2^n}$. The performance of random matchings were shown in the table below. We observe that the matching accuracy is extremely low and the probability to match no pair correctly increases with number of pairs.

Number of Pairs	Matching Accuracy	P(No Correct Matching)
5	0.2	0.3282
10	0.1	0.3562
15	0.066	0.3629
20	0.05	0.3584
30	0.033	0.3645
50	0.02	0.3704

3.1. *Synthetic Datasets.* In this simulation study, the "experts knowledge" are fixed functions and completely known. The name are trained on paired observations and tested on a set of held-out observations. The performances of the algorithms are measured by the proportions of matched pairs that are the same as the pairs matched by the "expert" in the testing data. We call this metric "matching accuracy". Note that we do not look at the metrics such as mean squared errors (Austin, 2014) that evaluate distributional balances. The expert matching results are assumed to be the best matching results possible, and algorithm were only evaluated by matching accuracy.

3.1.1. *Matching Accuracy Comparison.*

3.1.1.1. When the assumption is met. In this setting, we set the underlying true distance function to be a weighted Euclidean distance as assumed. Controlled observations $\{x_i^c\}_{i=1}^l$ are simulated from $MVN(\mu^c, 0.25I_p)$ such that $\mu^c = (i, ..., i)^T$. Treated observations $\{x_j^t\}_{j=1}^l$ are simulated from $MVN(\mu^t, 0.25\mathbf{I}_p)$ such that $\mu^t = \mu^c + (b, 0, ..., 0)$. Here, MVN stands for the multivariate normal distribution and b is a bias parameter that controls distributional difference between the controlled and treated population. We fix b at 0.5.

Although using the identity matrix as covariance matrix and including only one biased coordinate seems restrictive, we note that any pair of multivariate normal distribution $N(\mu_1, \Sigma_1)$, $N(\mu_2, \Sigma_2)$ can be converted to $N(\begin{pmatrix} b \\ \mathbf{O}_{p-1} \end{pmatrix}, \mathbf{I}_p)$ and $N(\mathbf{O}_p, \mathbf{I}_p)$ by an affine transformation (Schmee, 1986). Each control is matched to the closest treatment in terms of weighted Euclidean distance. We assume 2 random covariates are "principle confounders" with weights 0.9, and all other covariates with weights 0.05. The simulation results are robust to these choices. Since a unweighted Euclidean distance function that include only principle confounders yields near-optimal matching accuracy, we call all other covariates with weights 0.05 "noisy variables".

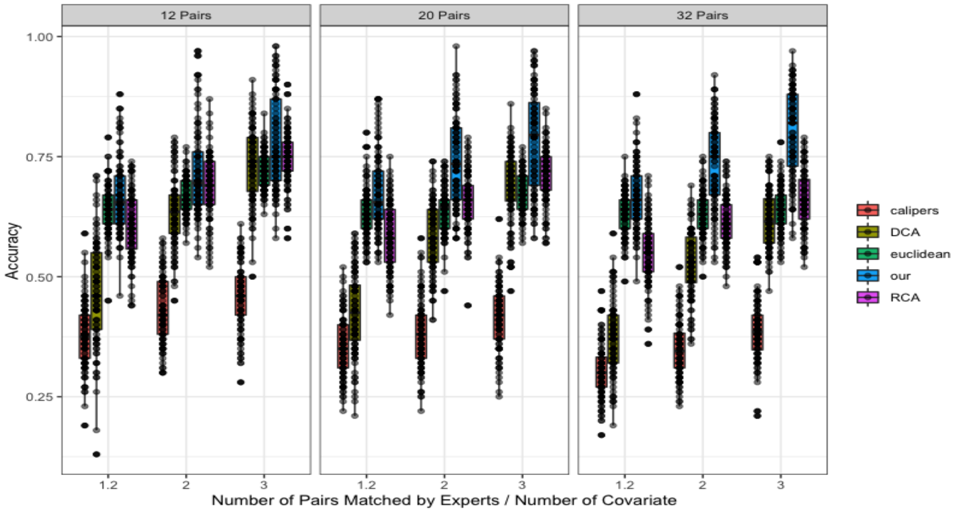

Figure 1: Matching Accuracy in Weighted Euclidean Distance Setting

3.1.1.2. When the assumptions is not met. In this scenario, the "expert" matches two observations only if the absolute differences between the principle confounders are within a prespecified threshold. This means the underlying distance function between x_i, x_j is now

$$d(x_i, x_j) = \begin{cases} \|w^T(x_i - x_j)\|_2^2 & \text{If } I(|x_{ik_1} - x_{jk_1}| < c)I(|x_{ik_2} - x_{jk_2} < c|) = 1 \\ \infty & \text{Otherwise} \end{cases} \tag{12}$$

where k_1, k_2 are the index for the principle confounders. In this scenario, the expert knowledge is no longer a weighted Euclidean distance, and it does not even satisfy triangle inequality property. We fix the threshold c to be 1.5 and simulate observations in the same way as the previous scenario.

3.1.1.3. Results. The matching accuracy of name is given by the boxplots in Figure 1 and Figure 2. Each point represent the performance of a certain algorithm on held-out testing data in one iteration. 100 iterations were performed for each set up. Since the number of meaningful confounders are fixed at 2, we can treat the ratio between the number of pairs of matched observations and p as a signal-to-noise rate. As the signal to noise rate increases, the performance of all supervised method improves in both set ups. The performance of Euclidean distance matching and propensity score matching do not change as they do not use paired observations.

In the linear set up, the name wins over its competitors by a decent margin. This margin becomes larger when p gets larger and the signal-to-noise ratio is fixed: all other methods seem to suffer from the increase in dimensionality even if the number of training pairs is increasing at the same rate while the performance of name stays the same.

When the underlying distance function is not exactly Euclidean, all algorithms performs worse. The matching accuracy of RCA and DCA appears to be extremely volatile and suggests the matching algorithms are sensitive to the training data. On the other hand, name is able to main the matching accuracy at roughly the linear setup level; the matching accuracy remains the same when p and number of training pairs increase at the same rate. This superior performance come from the fact that the contribution of each covariates are explicitly modeled by the weights β. Since the thresholds are taken only on the principle confounders, observations matched by name automatically falls in the thresholds.

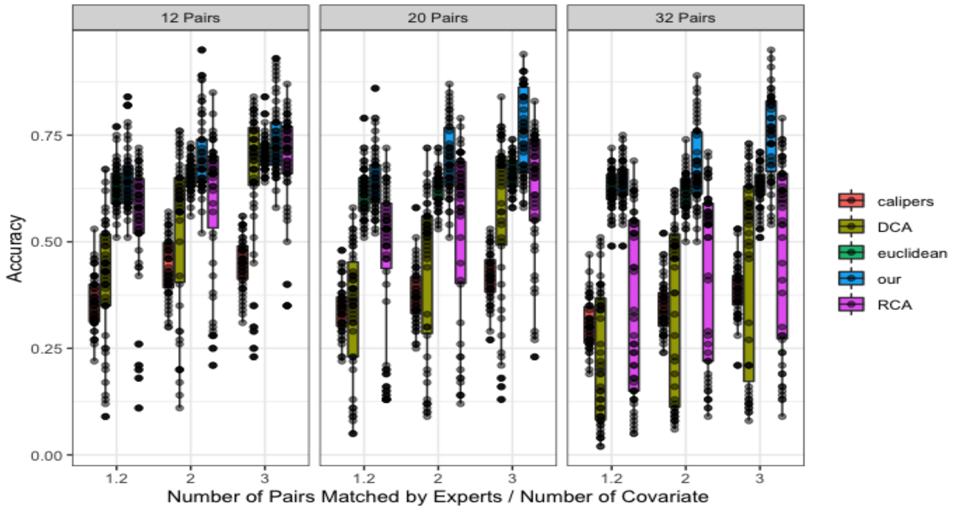

Figure 2: Matching Accuracy in Conjunctive Rules Setting

3.1.2. *Algorithm Robustness.* We observed from previous subsection that the name possesses 2 great performance properties when all assumptions were satisfied.

- The matching accuracy improves when the signal strength increases while the noise level is kept the same.
- When both the noise level and signal strength are kept the same, the matching accuracy does not deteriorate when the number of dimension increases.

Under the conjunctive rule setup, the distribution of matching accuracy is stretched upward instead of being shifted upward when the signal to noise ratio increases. At the mean time, there is no obvious relationship between matching accuracy and number of covariates.

We observed that the first property is only "partially" kept, while the second property broke down under the conjunctive rule setup when the model assumptions are grossly violated. In this subsection, we look into scenarios when data deviate from model assumptions in a controlled manner, namely when covariates are correlated and when there are interaction effects. The dimension of the simulated data in this section is fixed at 12.

Though independence between covariates are not explicitly assumed by name, high correlation between covariates might mislead name to distribute weights to noisy variables that are correlated with the principle confounders. The Figure 3 shows the matching performance when pairwise correlations are present in the data. We see the performance of name is not affected by correlation between covariates across a wide range of correlation strength.

We also consider the case when the product between principle confounders and other variables are confounders; in this case, the weight of the principle confounders depend on the magnitude of other variables. Order-two interactions between principle confounders and noise variables were randomly picked and included in the underlying weighted Euclidean distance. All interaction terms have weight 0.2, and the number of interactions is a proportion of p as shown in Figure 4.

We observe that the increase rate in matching accuracy decreases when interactions are present. In the extreme case, 40% of the noisy variables have interaction terms with principle confounders, and there is not much gain in increasing the number of training pairs.

In addition, the max matching accuracy that we observe drops if any interaction is present. This is expected as name cannot assign weights to unobserved variables.

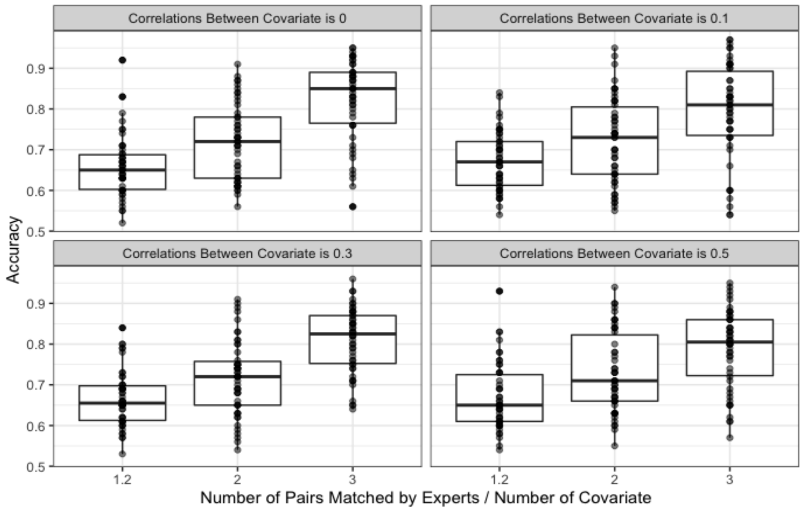

Figure 3

Lastly, the matching accuracy increases when the interaction terms are added and the signal to noise ratio is kept the same. Since the only new information source is the interaction terms, this suggests name learns from the interactions and assigns more weight to relevant variables even when the number of unobserved interactions is large. This hypothesis is also confirmed by the simulation results. We fetch the estimated weights of noisy variables and compare the magnitudes of those which are included in some interaction terms against those which are not. Their averaged differences in each simulation set up are shown by the table below. In this sense, name is robust against interactions.

# of Training Pairs	# of Interactions	Averaged Differences ×10
15	1	1.39
15	2	1.04
15	3	0.435
15	5	0.406
24	1	2.62
24	2	1.3
24	3	0.948
24	5	0.751
36	1	2.66
36	2	1.17
36	3	1.64
36	5	0.856

3.1.3. *Self-taught Learning.* Lastly, we look at the performance of the self-taught learning framework in the linear setting. We fix the number of matched pairs to be 15 and change its ratio between the number of unmatched pairs together with the number of covariates. The unmatched observations were simulated from the same distribution of the matched pairs. In each simulation setting, we run the self-taught learning for 10 iterations 50 times. The improvement of the matching accuracy is plotted as a function of initial accuracy and shown in blue by the 5. Each point in the plot is the results for one particular simulation; one can tell the procedure rarely suffers from negative learning. In addition, all blue lines have a quadratic

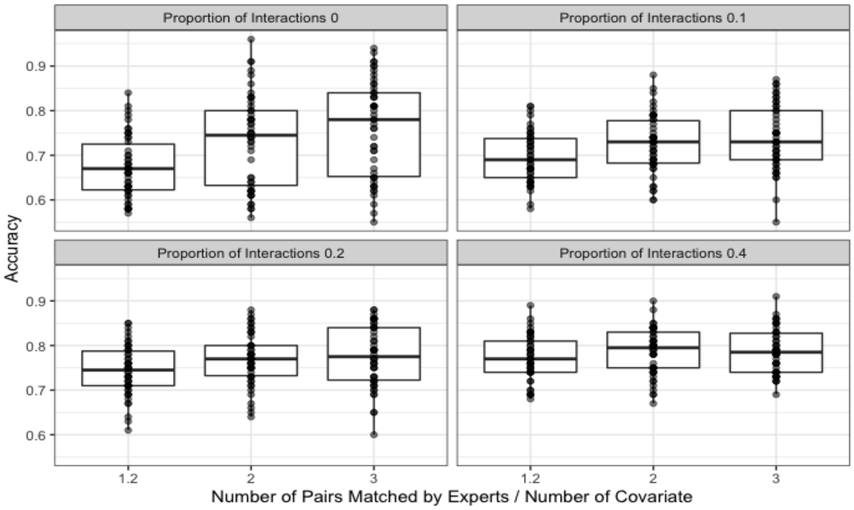

Figure 4

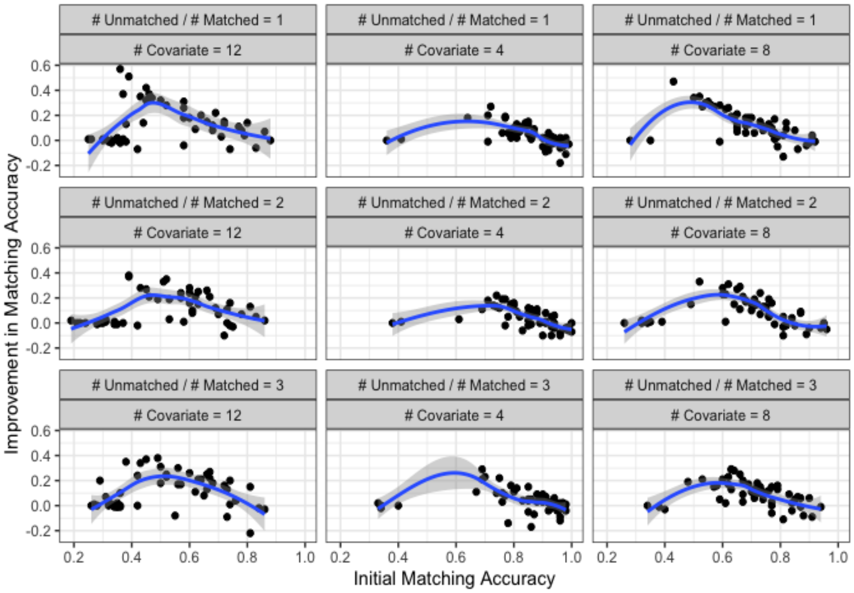

Figure 5

shape, which matches our expectation: name gains improvement from self-taught learning only when the initial performance is "mediocre".

The question left unanswered is how to find the range of mediocre performance for a specific problem. We recommend users to run the simulation with self-defined simulation parameters that are in line with their own data. One should use the self-taught procedure when the cross-validated performance is higher than the lowest initial accuracy that have an positive, averaged gain should be the and lower than the highest initial accuracy that have an positive, averaged gain. Moreover, one should set an "accuracy cap" to terminate the self-taught learning procedure when the current performance is higher than the upper bound we just defined.

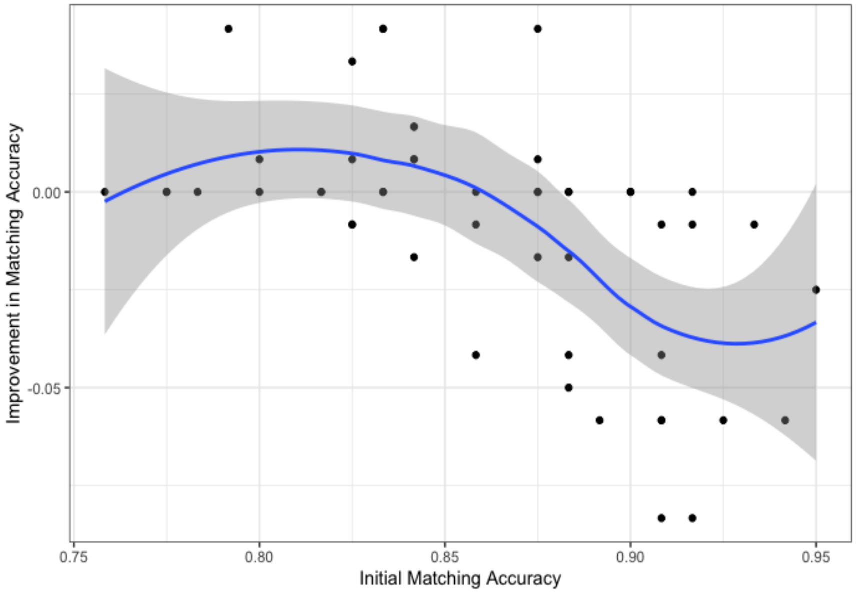

Figure 6

3.2. *Real Data.* We apply the name to a study designed to estimate the longitudinal effect of in-person schooling on SARS-CoV-2 county level transmission as measured by daily case incidence. reference. Based on some inclusion criteria, 229 counties were included in the initial study sample before matching, and 51 pairs of data points were matched by domain experts. We test how well name can learn replicate experts' decisions on these 51 pairs. We use all available baseline covariates when performing the name, which include geographical information (ex. state, Bureau of Economic Analysis (BEA) region), school activity level, mask enforcement strength, COVID-19 incidence; in total 13 variables. Since name take only numerical variables, for geographical indicators, we use the first 2 principal coordinates from multidimensional scaling (Cox and Cox, 2008). The cross-validated matching accuracies of name and its competitors are given in the table below.

	Euclidean Distance	RCA	DCA	name
Matching Accuracy	0.2	0.175	0.175	0.4

Since there are unpaired school districts besides the 51 paired ones, we naturally consider performing the self-taught learning procedure. Running the simulation with suitable parameters, however, suggest running the procedures may not help (see Figure 6). Running the procedure for 10 iterations indeed results in no improvement.

REFERENCES

Austin, P. C. (2014). A comparison of 12 algorithms for matching on the propensity score. *Statistics in medicine*, 33(6):1057–1069.

Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D., and Ridgeway, G. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of machine learning research*, 6(6).

Cox, M. A. and Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer.

Hoi, S. C., Liu, W., and Chang, S.-F. (2010). Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 6(3):1–26.

Hoi, S. C., Liu, W., Lyu, M. R., and Ma, W.-Y. (2006). Learning distance metrics with contextual constraints for image retrieval. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2072–2078. IEEE.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766.

Rockafellar, R. T. (2015). *Convex analysis*. Princeton university press.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Rosenbaum, P. R. and Rubin, D. B. (1985). Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38.

Schmee, J. (1986). An introduction to multivariate statistical analysis.

Si, L., Jin, R., Hoi, S. C., and Lyu, M. R. (2006). Collaborative image retrieval via regularized metric learning. *Multimedia Systems*, 12(1):34–44.

Wang, M. and Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429:215–244.

Xing, E., Jordan, M., Russell, S. J., and Ng, A. (2002). Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:521–528.

Appendix. Here we provide the proof for **theorem 1**. Recall that the solution of β given training data X is the first eigenvector of $G(X) = A^{-1}X^TL_bX = (X^TL_wX + \lambda I_p)^{-1}X^TL_bX$. Observe the L_w and L_b can be expressed as

$$L_w = \begin{pmatrix} I_\ell & -I_\ell \\ -I_\ell & I_\ell \end{pmatrix} \; L_b = \begin{pmatrix} \ell I_\ell & I_\ell - J_l \\ I_\ell - J_l & \ell I_\ell \end{pmatrix} \tag{13}$$

Here I_ℓ is an identity matrix of size ℓ and $J_l = \mathbb{1}^T\mathbb{1}$ is a size ℓ matrix of 1. We neglect the subscript when they are obvious. Without loss of generality, assume the $\bar{X}_c$, $\bar{X}_t$, X_c, X_t are centered; we have $X_c^TJX_c = 0$, $X_t^TJX_t = 0$. Thus,

$$L_b = -L_w + (k+1)\mathbf{I}_{2l} \tag{14}$$

substitute (14) into $G(X)/\ell$, we have

$$\begin{aligned} G(X) &= \frac{1}{\ell}(X^TL_wX + \lambda I)^{-1}X^T(-L_w + (k+1)\mathbf{I}_{2l})X \\ &= \underbrace{\frac{1}{\ell}(X^TL_wX + \lambda I)^{-1}X^T(-L_w)X}_{G_1(X)} + \underbrace{(X^TL_wX + \lambda I)^{-1}X^TX}_{G_2(X)} \end{aligned} \tag{15}$$

It will become obvious later that we need to give both an upper bound and a lower bound for $\sigma_p(G(X))$, the smallest eigenvalue of G(X). Define $\hat{\Sigma}_{\ell_1} = \frac{1}{2\ell_1}\sum_{k=1}^{2\ell_1}\Big(X_{c,k} - X_{t,k}\Big)^T \cdot \Big(X_{c,k} - X_{t,k}\Big)$ and denote the p largest eigenvalues of matrix A as $\lambda_1(A), \cdots, \lambda_p(A)$. We firstly post an assumption on the observation vectors and provide two useful lemmas.

ASSUMPTION 1. *For all random observation vectors $X_{c,k}, X_{t,k}$ in X_c, X_t; we have a bounded fourth moment $E(X_{t,k}) = E(X_{c,k}) = 0$ and $Var(X_{c,k}) = Var(X_{t,k}) = \Sigma$. In addition, the fourth moments of the their coordinates are bounded; i.e. $\exists M, \max_k E(X_{c,k,i}^4) < M$ and $\max_k E(X_{t,k,i}^4) < M$, $k = 1, \cdots, \ell_1$.*

The first lemma follows directly from the assumption.

LEMMA 3.1. *For all $k = 1, \cdots, \ell_1$ and $p = 1, \cdots, p$, we have $Var[(X_{c,k,i} - X_{t,k,i})(X_{c,k,j} - X_{t,k,j})] \leq 4M$.*

PROOF.

$$Var[(X_{c,k,i}-X_{t,k,i})(X_{c,k,j}-X_{t,k,j})] \tag{16}$$

$$\leq E(X^4_{c,k,i})+E(X^4_{t,k,i})+E(X^4_{c,k,j})+E(X^4_{t,k,j}) \tag{17}$$

$$\leq 4M \tag{18}$$

□

We then give the convergence results of $\hat{\Sigma}_{\ell_1}$ which is used multiple times later.

LEMMA 3.2. $||\hat{\Sigma}_{\ell_1}||_2 \xrightarrow{p} 2\lambda_1(\Sigma), \lambda_p(\hat{\Sigma}_{\ell_1}) \xrightarrow{p} 2\lambda_p(\Sigma), ||X^TX/2\ell_1||_2 \xrightarrow{p} \lambda_1(\Sigma), \lambda_p(X^TX/2\ell_1) \xrightarrow{p} \lambda_p(\Sigma)$

PROOF.

$$\hat{\Sigma}_{\ell_1} = \frac{1}{2\ell_1}\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big) \tag{19}$$

$$= \frac{1}{2\ell_1}\sum_{k=1}^{2\ell_1}\Big[\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big)-2\Sigma\Big]+2\Sigma \tag{20}$$

$$= \frac{1}{\sqrt{2\ell_1}}\sum_{k=1}^{2\ell_1}\Big[\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big)-2\Sigma\Big]/\sqrt{2\ell_1}+2\Sigma \tag{21}$$

$$:= \frac{1}{\sqrt{2\ell_1}}K+2\Sigma \tag{22}$$

Now with (21) and (22), we have K_{ij} with mean 0 and variance denoted by σ^2_{ij}. Define a unit vector γ such that $||\gamma||_2 = 1$. Consider

$$|\gamma^TK\gamma| = \sum_{i=1}^{p}\sum_{j=1}^{p}K_{i,j}\gamma_i\gamma_j \tag{23}$$

$$\leq \sum_{i=1}^{p}\sum_{j=1}^{p}|K_{i,j}|(\gamma_i^2+\gamma_j^2)/2 \tag{24}$$

$$\leq \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p}|K_{i,j}| \tag{25}$$

$$\leq \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p}a\sigma_{ij} \text{ with probability } > 1-p^2/a^2 \tag{26}$$

$$\leq ap^2\sqrt{M} \tag{27}$$

where in the last step we used Chebyshev's inequality. Take $a = p^{-2}\sqrt{M}\ell_1^{1/4}$, we have $\ell_1^{-1/4}|\gamma^TK\gamma| \xrightarrow{p} 1$ so $\lambda_1(K)/\sqrt{2\ell_1} = O_p(1)$ and $||\hat{\Sigma}_{\ell_1}||_2 \xrightarrow{p} 2\lambda_1(\Sigma), \lambda_p(\hat{\Sigma}_{\ell_1}) \xrightarrow{p} 2\lambda_p(\Sigma)$. Similarly, consider $X^TX/2\ell_1 = \frac{1}{\sqrt{2\ell_1}}\sum_{k=1}^{2\ell_1}(X_k^TX_k-\Sigma)/\sqrt{2\ell_1}+\Sigma := \frac{1}{\sqrt{2\ell_1}}D+\Sigma$. By central limit theorem, we have D_{ij} with mean 0 and variance denoted by $\tilde{\sigma}^2_{ij}$. Again, we have $\ell_1^{-1/4}|\gamma^TD\gamma| \xrightarrow{p} 1$ so $\lambda_1(D)/\sqrt{2\ell_1} = O_p(1)$ and $||X^TX/2\ell_1||_2 \xrightarrow{p} \lambda_1(\Sigma), \lambda_p(X^TX/2\ell_1) \xrightarrow{p} \lambda_p(\Sigma)$. □

Therefore, we have

$$||G_1(X)||_2 =||\frac{\lambda}{\ell_1}\Big[\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big)+\lambda I_p\Big]^{-1}-\frac{I_p}{\ell_1}||_2 \tag{28}$$

$$\leq||\frac{\lambda}{2\ell_1^2}\Big[\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big)/2\ell_1+\lambda I_p/2\ell_1\Big]^{-1}||_2+\frac{1}{\ell_1} \tag{29}$$

$$=\frac{\lambda}{2\ell_1^2}\frac{1}{\lambda_p(\hat{\Sigma}_{\ell_1})+\lambda/2\ell_1}+\frac{1}{\ell_1} \tag{30}$$

By lemma 2, we see $||G_1(X)||_2 = O_p(1/\ell_1)$. In addition,

$$||G_2(X)||_2 =||(\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big)/\ell_1+\lambda I/\ell_1)^{-1}X^TX/\ell_1||_2 \tag{31}$$

$$\leq\frac{1}{\sqrt{2}}||(\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\Big(X_{c,k}-X_{t,k}\Big)/2\ell_1+\lambda I/2\ell_1)^{-1}||_2\sqrt{2}||X^TX/2\ell_1||_2 \tag{32}$$

$$=\frac{1}{\lambda_p(\hat{\Sigma}_{\ell_1})+\lambda/2\ell_1}\lambda_1(\frac{X^TX}{2\ell_1}) \tag{33}$$

$$||G_2(X)^{-1}||_2 =||[\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\cdot\Big(X_{c,k}-X_{t,k}\Big)/\ell_1+\lambda I/\ell_1]||_2 \tag{34}$$

$$\leq||\sum_{k=1}^{2\ell_1}\Big(X_{c,k}-X_{t,k}\Big)^T\Big(X_{c,k}-X_{t,k}\Big)/2\ell_1+\lambda I/2\ell_1||_2||(X^TX/\ell_1)^{-1}||_2 \tag{35}$$

$$=\frac{\lambda_1(\hat{\Sigma}_{\ell_1})+\lambda/2\ell_1}{\lambda_p(\frac{X^TX}{2\ell_1})} \tag{36}$$

Again, by lemma 2, we see $||G_2(X)||_2 \xrightarrow{p} 1/2$ and $1/||G_2(X)^{-1}||_2 \xrightarrow{p} \lambda_p(\Sigma)/[2\lambda_1(\Sigma)]$. Observe that $\lambda_p(G(X)) \leq ||G(X)||_2 \leq ||G_1(X)||_2 + ||G_2(X)||_2$; in addition, $\lambda_p(G(X)) \geq \lambda_p(G_2(X)) - ||G_1(X)||_2 = 1/||G_2(X)^{-1}||_2 - ||G_1(X)||_2$. We conclude that asymptotically

$$\frac{1}{2} \geq \sigma_r(G(X)) \geq \frac{1}{2}\frac{\lambda_p(\Sigma)}{\lambda_1(\Sigma)}$$

Naturally, the convergence of $\hat{\beta}$ to β^* should depend on the size of the difference between $G(\tilde{X})$ and $G(X)$. Define $H = G_1(\tilde{X}) - G_1(X) + G_2(\tilde{X}) - G_2(X)$ so

$$||H||_2 \leq \underbrace{||G_1(\tilde{X}) - G_1(X)||_2}_{\textcircled{1}} + \underbrace{||G_2(\tilde{X}) - G_2(X)||_2}_{\textcircled{2}}$$

For the first part, we have

$$\textcircled{1} \leq ||\frac{\lambda}{2\ell_1^2}[\hat{\Sigma}_{\ell_1}+\lambda I_p]^{-1}-\frac{\lambda}{2\ell^2}[\hat{\Sigma}_{\ell}+\lambda I_p]^{-1}||_2+||I_p/\ell_1-I_p/\ell||_2 \tag{37}$$

$$\leq ||\frac{\lambda}{2\ell_1^2}[\hat{\Sigma}_{\ell_1}+\lambda I_p]^{-1}||+||\frac{\lambda}{2\ell^2}[\hat{\Sigma}_{\ell}+\lambda I_p]^{-1}||_2+||I_p/\ell_1-I_p/\ell||_2 \tag{38}$$

$$=\frac{\lambda}{2\ell_1^2}\frac{1}{\lambda_p(\hat{\Sigma}_{\ell_1})+\lambda/2\ell_1}+\frac{\lambda}{2\ell^2}\frac{1}{\lambda_p(\hat{\Sigma}_{\ell})+\lambda/2\ell}+||I_p/\ell_1-I_p/\ell||_2 \tag{39}$$

$$=O_p(\frac{1}{\ell_1^2})+O_p(\frac{1}{\ell^2})+O_p(1/\ell_1-1/\ell) \tag{40}$$

$$\overset{(i)}{=}O_p(1/\ell_1) \tag{41}$$

where in (i) we use the fact that $\ell_1/\ell=C$ and $C<1$. For the second part,

$$\textcircled{2}\leq ||[(\hat{\Sigma}_{\ell_1}+\lambda I/2\ell_1)^{-1}X^TX/2\ell_1 \tag{42}$$

$$-((\hat{\Sigma}_{\ell}+\lambda I/2\ell)^{-1}X^TX/2\ell]||_2 \tag{43}$$

$$\leq ||(\hat{\Sigma}_{\ell_1}+\lambda I/2\ell_1)^{-1}X^TX/2\ell_1-(2\Sigma+\lambda/\ell_1 I_p)^{-1}\Sigma||_2 \tag{44}$$

$$+||(2\Sigma+\lambda I_p/\ell_1)^{-1}\Sigma-(2\Sigma+\lambda I_p/\ell)^{-1}\Sigma||_2 \tag{45}$$

$$+||(2\Sigma+\lambda I_p/\ell)^{-1}\Sigma-(\hat{\Sigma}_{\ell}+\lambda I/2\ell)^{-1}X^TX/2\ell||_2 \tag{46}$$

Consider the second term (45) first, we have

$$||(2\Sigma+\lambda I_p/\ell_1)^{-1}\Sigma-(2\Sigma+\lambda I_p/\ell_2)^{-1}\Sigma||_2 \tag{47}$$

$$=||\gamma^T(2\Sigma+\lambda I_p/\ell_1)^{-1}\Sigma\gamma-\gamma^T(2\Sigma+\lambda I_p/\ell_2)^{-1}\Sigma\gamma||_2 \tag{48}$$

$$=||\gamma^T[diag(\frac{\lambda_1}{\lambda_1+\lambda/\ell_1},\cdots,\frac{\lambda_p}{\lambda_p+\lambda/\ell_1}) \tag{49}$$

$$-diag(\frac{\lambda_1}{\lambda_1+\lambda/\ell},\cdots,\frac{\lambda_p}{\lambda_p+\lambda/\ell})]\gamma||_2 \tag{50}$$

$$=||\gamma^T diag(\frac{\lambda/\ell_1-\lambda/\ell}{\lambda_1+\lambda/\ell_1},\cdots,\frac{\lambda/\ell_1-\lambda/\ell}{\lambda_p+\lambda/\ell_1})\gamma||_2 \tag{51}$$

$$=O(1/\ell_1-1/\ell)=O(1/\ell_1) \tag{52}$$

The first term (44) and the third term (46) can be treated in the same manner. Without loss of generality, let's consider the first term

$$||(\hat{\Sigma}_{\ell_1}+\lambda I/2\ell_1)^{-1}X^TX/2\ell_1-(2\Sigma+\lambda/\ell_1 I_p)^{-1}\Sigma||_2 \tag{53}$$

$$\leq||(\hat{\Sigma}_{\ell_1}+\lambda I/2\ell_1)^{-1}(X^TX/2\ell_1-\Sigma)||_2 \tag{54}$$

$$+||[(\hat{\Sigma}_{\ell_1}+\lambda I/2\ell_1)^{-1}-(2\Sigma+\lambda/\ell_1 I_p)^{-1}]\Sigma||_2 \tag{55}$$

(54) is easier to be dealt with, we have (54) $\leq ||(\hat{\Sigma}_{\ell_1}/2\ell_1+\lambda I/2\ell_1)^{-1}||_2||(X^TX/2\ell_1-\Sigma)||_2$. By lemma 2, we know its first term is constant while second term is equal to $||D/\sqrt{2\ell_1}||_2=O_p(1/\sqrt{\ell_1})$. Dealing with (55) is more complicated, firstly note that

$$(55)\leq||[(\hat{\Sigma}_{\ell_1}+\lambda I/2\ell_1)^{-1}-(2\Sigma+\lambda/\ell_1 I_p)^{-1}]||_2||\Sigma||_2$$

We only need to bound its first part. Define $A=-\hat{\Sigma}_{\ell_1}+2\Sigma, B=2\Sigma+\lambda I_p/\ell_1$, we can write the first part as

$$||(B-A)^{-1}-B^{-1}||_2 \tag{56}$$

$$=||(I_p - B^{-1}A)^{-1}B^{-1} - B^{-1}||_2 \tag{57}$$

$$\leq ||B^{-1}||_2||(I_p - B^{-1}A)^{-1} - I_p||_2 \tag{58}$$

$$\overset{(i)}{=} ||B^{-1}||_2||B^{-1}A + (B^{-1}A)^2 + \cdots ||_2 \tag{59}$$

$$\leq ||B^{-1}||_2 \sum_{i=1}^{\infty} ||(B^{-1}A)^k||_2 \tag{60}$$

$$= ||B^{-1}||_2||B^{-1}A||_2[1 + \sum_{i=2}^{\infty} ||B^{-1}A||_2^k] \tag{61}$$

$$\overset{(ii)}{\leq} 2||B^{-1}||_2||B^{-1}A||_2 \tag{62}$$

$$\overset{(iii)}{=} O_p(1/\sqrt{l_1}) \tag{63}$$

In step $(i), (ii)$, we use the fact that $||B^{-1}A||_2 < 1$ and (iii) follows from results in lemma 2. With similar techniques, we know $(46) = O_p(1/\sqrt{l})$.

We now borrow strength from the powerful Davis-Kahan $\sin(\theta)$ Theorem. Define $dist(X, Z) = ||XX^T - ZZ^T||$ which is invariant to global orthonormal transformation, the theorem says that

$$dist(\hat{\beta}, \beta^*) \leq \frac{||H||}{\lambda_p(G(X)) - ||H||} \tag{64}$$

We have proved that $||H||_2 \leq O_p(1/\sqrt{\ell_1}) + O_p(1/\sqrt{\ell}) + O_p(1/\ell_1) = O_p(1/\sqrt{\ell_1})$ while $\lambda_p(G(X))$ converges to constant, $dist(\hat{\beta}, \beta^*) = O_p(1/\sqrt{\ell_1})$ follows.