

Genetic Matching

October 28, 2012

Overview

- Two tasks for matching inference (again):
 1. Create the matched strata to eliminate bias (imbalance)
 2. Do these matches look **comparable?**

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k
- Use Mahalanobis Distances
 - Equal Percent Bias Reduction (EPBR) property

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k
- Use Mahalanobis Distances
 - Equal Percent Bias Reduction (EPBR) property
- Limitations of Mahalanobis Matching

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k
- Use Mahalanobis Distances
 - Equal Percent Bias Reduction (EPBR) property
- Limitations of Mahalanobis Matching
 - Non-Ellipsoidal X (dichotomous, log, etc)

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k
- Use Mahalanobis Distances
 - Equal Percent Bias Reduction (EPBR) property
- Limitations of Mahalanobis Matching
 - Non-Ellipsoidal X (dichotomous, log, etc)
 - Ellipsoidal finite data

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k
- Use Mahalanobis Distances
 - Equal Percent Bias Reduction (EPBR) property
- Limitations of Mahalanobis Matching
 - Non-Ellipsoidal X (dichotomous, log, etc)
 - Ellipsoidal finite data
 - Nonlinear Reductions in Bias

Create Matched Strata

- Match on Multivariate X in k dimensions, X_k
- Use Mahalanobis Distances
 - Equal Percent Bias Reduction (EPBR) property
- Limitations of Mahalanobis Matching
 - Non-Ellipsoidal X (dichotomous, log, etc)
 - Ellipsoidal finite data
 - Nonlinear Reductions in Bias
 - “Prioritized” Reduction in Bias

Do Matches Look Comparable

- Balance tautology
 - Propensity score is a balancing score on X
 - Obtaining **balance** on X is the goal of matching

Do Matches Look Comparable

- Balance tautology
 - Propensity score is a balancing score on X
 - Obtaining **balance** on X is the goal of matching
- Genetic Matching incorporates balancing directly in the algorithm
 - Generalizing Mahalanobis matching
 - Prioritizes differences on certain covariates
 - Optimizes matching for some target: balance on X_p

Mahalanobis Matching

- Define a multivariate distance on X between i and j :

$$md(X_i, X_j) = \left\{ (X_i - X_j)^T S^{-1} (X_i - X_j) \right\}^{\frac{1}{2}}$$

Mahalanobis Matching

- Define a multivariate distance on X between i and j :

$$md(X_i, X_j) = \left\{ (X_i - X_j)^T S^{-1} (X_i - X_j) \right\}^{\frac{1}{2}}$$

- Note that S is the sample covariance matrix of X
 - Normalize the distances
 - Gives equal (normalized) weight to each covariate in X

Weighted Mahalanobis Matching

- Some covariates may be more important than others

Weighted Mahalanobis Matching

- Some covariates may be more important than others
- Intuitively, we already weight covariates either at 0 or 1 in measuring $md(X_i, X_j)$

Weighted Mahalanobis Matching

- Some covariates may be more important than others
- Intuitively, we already weight covariates either at 0 or 1 in measuring $md(X_i, X_j)$
- Define X_p as a full matrix of covariates and X_k as the k -dimensional matrix we match on, where $k < p$

Weighted Mahalanobis Matching

- Some covariates may be more important than others
- Intuitively, we already weight covariates either at 0 or 1 in measuring $md(X_i, X_j)$
- Define X_p as a full matrix of covariates and X_k as the k -dimensional matrix we match on, where $k < p$
- Let \tilde{w} be a vector of weights so that:

$$\tilde{w} = \{w_1 = 1, w_2 = 1, \dots, w_k = 1, w_{k+1} = 0, \dots, w_p = 0\}$$

Weighted Mahalanobis Matching

- Generalize this weighting in a flexible way

$$\tilde{w} \in \{\underline{w}, \bar{w}\}$$

Weighted Mahalanobis Matching

- Generalize this weighting in a flexible way

$$\tilde{W} \in \{\underline{W}, \bar{W}\}$$

- Incorporate these weights in W to rescale unit distances

$$wmd(X_i, X_j) = \left\{ (X_i - X_j)^T (S^{-\frac{1}{2}})^T W (S^{-\frac{1}{2}}) (X_i - X_j) \right\}^{\frac{1}{2}}$$

Weighted Mahalanobis Matching

- Generalize this weighting in a flexible way

$$\tilde{w} \in \{\underline{w}, \bar{w}\}$$

- Incorporate these weights in W to rescale unit distances

$$wmd(X_i, X_j) = \left\{ (X_i - X_j)^T (S^{-\frac{1}{2}})^T W (S^{-\frac{1}{2}}) (X_i - X_j) \right\}^{\frac{1}{2}}$$

- W is a $k \times k$ positive definite weight matrix, with \tilde{w} the diagonal elements (usually 0 in the off-diagonals)

Weighted Mahalanobis Matching

- Generalize this weighting in a flexible way

$$\tilde{w} \in \{\underline{w}, \bar{w}\}$$

- Incorporate these weights in W to rescale unit distances

$$wmd(X_i, X_j) = \left\{ (X_i - X_j)^T (S^{-\frac{1}{2}})^T W (S^{-\frac{1}{2}}) (X_i - X_j) \right\}^{\frac{1}{2}}$$

- W is a $k \times k$ positive definite weight matrix, with \tilde{w} the diagonal elements (usually 0 in the off-diagonals)
- $S^{-\frac{1}{2}}$ is the cholesky decomposition of S sample covariance matrix

$$(S^{-\frac{1}{2}})^T (S^{-\frac{1}{2}}) = S$$

Example

- Suppose we have $x_1 \sim N(\mu_1, 1)$ and $x_2 \sim N(\mu_2, 1)$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Example

- Suppose we have $x_1 \sim N(\mu_1, 1)$ and $x_2 \sim N(\mu_2, 1)$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Then:

$$\begin{aligned} wmd(X_i, X_j) &= \left\{ (X_i - X_j)^T W (X_i - X_j) \right\}^{\frac{1}{2}} \\ &= \left\{ w_1(x_{1i} - x_{1j})^2 + w_2(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \end{aligned}$$

Example

- Suppose we have $x_1 \sim N(\mu_1, 1)$ and $x_2 \sim N(\mu_2, 1)$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Then:

$$\begin{aligned} wmd(X_i, X_j) &= \left\{ (X_i - X_j)^T W (X_i - X_j) \right\}^{\frac{1}{2}} \\ &= \left\{ w_1 (x_{1i} - x_{1j})^2 + w_2 (x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \end{aligned}$$

Example

- Suppose we have $x_1 \sim N(\mu_1, 1)$ and $x_2 \sim N(\mu_2, 1)$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Then:

$$\begin{aligned} wmd(X_i, X_j) &= \left\{ (X_i - X_j)^T W (X_i - X_j) \right\}^{\frac{1}{2}} \\ &= \left\{ w_1(x_{1i} - x_{1j})^2 + w_2(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \end{aligned}$$

- If $2w_1 = w_2$

Example

- Suppose we have $x_1 \sim N(\mu_1, 1)$ and $x_2 \sim N(\mu_2, 1)$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Then:

$$\begin{aligned} wmd(X_i, X_j) &= \left\{ (X_i - X_j)^T W (X_i - X_j) \right\}^{\frac{1}{2}} \\ &= \left\{ w_1(x_{1i} - x_{1j})^2 + w_2(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \end{aligned}$$

- If $2w_1 = w_2$

$$= \left\{ w_1(x_{1i} - x_{1j})^2 + 2w_1(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}}$$

Example

- Suppose we have $x_1 \sim N(\mu_1, 1)$ and $x_2 \sim N(\mu_2, 1)$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Then:

$$\begin{aligned} wmd(X_i, X_j) &= \left\{ (X_i - X_j)^T W (X_i - X_j) \right\}^{\frac{1}{2}} \\ &= \left\{ w_1(x_{1i} - x_{1j})^2 + w_2(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \end{aligned}$$

- If $2w_1 = w_2$

$$\begin{aligned} &= \left\{ w_1(x_{1i} - x_{1j})^2 + 2w_1(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \\ &\approx \left\{ (x_{1i} - x_{1j})^2 + 2(x_{2i} - x_{2j})^2 \right\}^{\frac{1}{2}} \end{aligned}$$

Example: Big Points

- Weighting rescales (upweights) relative differences on some covariates over other differences

Example: Big Points

- Weighting rescales (upweights) relative differences on some covariates over other differences
- Affine Invariant Matching
 - Divide all weights by w_1 , affine transformation of wmd
 - Matching on (1) and (2) results in the same matches

Example: Big Points

- Weighting rescales (upweights) relative differences on some covariates over other differences
- Affine Invariant Matching
 - Divide all weights by w_1 , affine transformation of wmd
 - Matching on (1) and (2) results in the same matches
- How do we choose the weights \tilde{w} ?

Evolutionary Algorithm to Select Weights

- The relevant task is to identify the k weights

Evolutionary Algorithm to Select Weights

- The relevant task is to identify the k weights
- Pick weights to minimize differences on X_p across treated and control units after matching

Evolutionary Algorithm to Select Weights

- The relevant task is to identify the k weights
- Pick weights to minimize differences on X_p across treated and control units after matching
- Genetic Matching
 - Evolutionary algorithm to optimize weights over multiple 'generations'
 - Optimize means reduces imbalance on included X covariates

Genetic Matching Workflow

- T is treatment vector
- X_k is covariates for matching (“Match Matrix”)
- X_p is covariates for balancing (“Balance Matrix”)
- Note: X_p can include **any**, **all** or **none** of X_k

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances
 - c. Compute a balance metric $\mathbb{B}_0(s)$, a function of all s matched pairs

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances
 - c. Compute a balance metric $\mathbb{B}_0(s)$, a function of all s matched pairs
 4. Select the m^{th} weights (\tilde{w}_0^m) that maximizes $\mathbb{B}_0(s)$

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances
 - c. Compute a balance metric $\mathbb{B}_0(s)$, a function of all s matched pairs
 4. Select the m^{th} weights (\tilde{w}_0^m) that maximizes $\mathbb{B}_0(s)$
- At next generation g_1 to g_n :

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances
 - c. Compute a balance metric $\mathbb{B}_0(s)$, a function of all s matched pairs
 4. Select the m^{th} weights (\tilde{w}_0^m) that maximizes $\mathbb{B}_0(s)$
- At next generation g_1 to g_n :
 5. Retain \tilde{w}_{t-1}^m as the best starting value from g_{t-1}

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances
 - c. Compute a balance metric $\mathbb{B}_0(s)$, a function of all s matched pairs
 4. Select the m^{th} weights (\tilde{w}_0^m) that maximizes $\mathbb{B}_0(s)$
- At next generation g_1 to g_n :
 5. Retain \tilde{w}_{t-1}^m as the best starting value from g_{t-1}
 6. Repeat steps 2 to 4 from above, maximizing $\mathbb{B}_t(s)$ at each step

Genetic Matching Workflow

- At starting generation g_0 :
 1. Select starting \tilde{w}_0
 2. Devise m weight vectors using evolutionary 'string' operators (called 'populations')
 3. For each weight vector:
 - a. Compute $wmd(X_k, \tilde{w}_0)$
 - b. Match i treated to j control units using $wmd(X_{ki}, X_{kj} \tilde{w}_0)$ distances
 - c. Compute a balance metric $\mathbb{B}_0(s)$, a function of all s matched pairs
 4. Select the m^{th} weights (\tilde{w}_0^m) that maximizes $\mathbb{B}_0(s)$
- At next generation g_1 to g_n :
 5. Retain \tilde{w}_{t-1}^m as the best starting value from g_{t-1}
 6. Repeat steps 2 to 4 from above, maximizing $\mathbb{B}_t(s)$ at each step
 7. Stop by user control or once no more 'fitness' gains over last few generations

Why Genetic Matching

- Can incorporate a propensity score, orthogonalizing X
 - GenMatch approaches the propensity score if correct, but with lower mean square error
- Loss function minimizes the maximum discrepancy
 - User can pass any desired loss function of X_p after matching
- When EPBR fails GenMatch reliably reduces imbalances on included covariates
- But...design trumps analysis
 - Why trust the ignorability assumption?
 - Balance checks?
 - Placebo tests?