NP-complete Problems and Large Datasets: Examples from the `Matching` Package

Jasjeet S. Sekhon

UC Berkeley

The Problem

- The size of datasets is growing fast in the social sciences—e.g., administrative data with millions of observations
- Our methods are becoming computationally intensive: MCMC, genetic optimization, simulating annealing, and **MATCHING**
- These methods are often used for problems whose asymptotic order is exponential: $\mathbf{O}(c^N),\ c > 1$
- We try to turn exponential problems into polynomial time problems:
  - $\mathbf{O}(N^2 log(N))$ for pair matching
  - $\mathbf{O}(N^3 log(N \max(\text{dist})))$ for full matching

The Problem

- The size of datasets is growing fast in the social sciences—e.g., administrative data with millions of observations
- Our methods are becoming computationally intensive: MCMC, genetic optimization, simulating annealing, and **MATCHING**
- These methods are often used for problems whose asymptotic order is exponential: $\mathbf{O}(c^N),\ c > 1$
- We try to turn exponential problems into polynomial time problems:
  - $\mathbf{O}(N^2 log(N))$ for pair matching
  - $\mathbf{O}(N^3 log(N \max(\text{dist})))$ for full matching

The Problem

- The size of datasets is growing fast in the social sciences—e.g., administrative data with millions of observations
- Our methods are becoming computationally intensive: MCMC, genetic optimization, simulating annealing, and **MATCHING**
- These methods are often used for problems whose asymptotic order is exponential: $\mathbf{O}(c^N),\ c > 1$
- We try to turn exponential problems into polynomial time problems:
  - $\mathbf{O}(N^2 log(N))$ for pair matching
  - $\mathbf{O}(N^3 log(N \max(\text{dist})))$ for full matching

The Limits of Computational Power

- Faster computers will take care of our problems?
- With faster and cheaper computers, it is easier to get more data
- Moore's Law is currently holding because of parallelization–e.g., Core Duo, Cell processor
- Is parallel code worth the effort?
- R was not designed for software development and computationally intensive algorithms—e.g., pass by value
- Software engineering is not well supported in R

The Limits of Computational Power

- Faster computers will take care of our problems?
- With faster and cheaper computers, it is easier to get more data
- Moore's Law is currently holding because of parallelization–e.g., Core Duo, Cell processor
- Is parallel code worth the effort?
- R was not designed for software development and computationally intensive algorithms—e.g., pass by value
- Software engineering is not well supported in R

The Limits of Computational Power

- Faster computers will take care of our problems?
- With faster and cheaper computers, it is easier to get more data
- Moore's Law is currently holding because of parallelization–e.g., Core Duo, Cell processor
- Is parallel code worth the effort?
- R was not designed for software development and computationally intensive algorithms—e.g., pass by value
- Software engineering is not well supported in R

The Limits of Computational Power

- Faster computers will take care of our problems?
- With faster and cheaper computers, it is easier to get more data
- Moore's Law is currently holding because of parallelization–e.g., Core Duo, Cell processor
- Is parallel code worth the effort?
- R was not designed for software development and computationally intensive algorithms—e.g., pass by value
- Software engineering is not well supported in R

The Limits of Computational Power

- Faster computers will take care of our problems?
- With faster and cheaper computers, it is easier to get more data
- Moore's Law is currently holding because of parallelization–e.g., Core Duo, Cell processor
- Is parallel code worth the effort?
- R was not designed for software development and computationally intensive algorithms—e.g., pass by value
- Software engineering is not well supported in R

Example: Matching

- The goal of matching is to make the distribution of observed covariates the same between treatment and control groups
- People measure covariate balance various ways:
  - minimize mean differences of observed confounders across matched treated and control units
  - minimize the discrepancy in QQ-plots across the variables
- The search space of possible matched datasets grows exponentially with N: it's a Traveling Salesman problem

Example: Matching

- The goal of matching is to make the distribution of observed covariates the same between treatment and control groups
- People measure covariate balance various ways:
  - minimize mean differences of observed confounders across matched treated and control units
  - minimize the discrepancy in QQ-plots across the variables
- The search space of possible matched datasets grows exponentially with N: it's a Traveling Salesman problem

Example: Matching

- The goal of matching is to make the distribution of observed covariates the same between treatment and control groups
- People measure covariate balance various ways:
  - minimize mean differences of observed confounders across matched treated and control units
  - minimize the discrepancy in QQ-plots across the variables
- The search space of possible matched datasets grows exponentially with N: it's a Traveling Salesman problem

Example: Matching

- The goal of matching is to make the distribution of observed covariates the same between treatment and control groups
- People measure covariate balance various ways:
  - minimize mean differences of observed confounders across matched treated and control units
  - minimize the discrepancy in QQ-plots across the variables
- The search space of possible matched datasets grows exponentially with N: it's a Traveling Salesman problem

Propensity Score (pscore)

- The propensity score is:

$$Pr(T_i = 1|X_i) = E(T_i|X_i)$$

- Almost always estimated by logistic regression
- It greatly helps to reduces the difficulty of the matching problem.
- If one matches on and balances the propensity score, one balances the confounders X of concern.
- Balance: to make the distributions the same between treatment and control groups
- Multidimensional covariate balance is difficult to measure

Propensity Score (pscore)

- The propensity score is:

$$Pr(T_i = 1|X_i) = E(T_i|X_i)$$

- Almost always estimated by logistic regression
- It greatly helps to reduces the difficulty of the matching problem.
- If one matches on and balances the propensity score, one balances the confounders X of concern.
- Balance: to make the distributions the same between treatment and control groups
- Multidimensional covariate balance is difficult to measure

Propensity Score (pscore)

- The propensity score is:

$$Pr(T_i = 1|X_i) = E(T_i|X_i)$$

- Almost always estimated by logistic regression
- It greatly helps to reduces the difficulty of the matching problem.
- If one matches on and balances the propensity score, one balances the confounders X of concern.
- Balance: to make the distributions the same between treatment and control groups
- Multidimensional covariate balance is difficult to measure

Searching for the Correct Propensity Score Model

- People use methods to simplify the problem of searching for the best set of matches: propensity score (pscore)
- This simplification turns the problem into a polynomial time problem: $\mathbf{O}(N^2 log(N))$ for pair matching
- How do we know what is the correct propensity score model?
- Matching on the "correct" pscore obtains covariate balance–it's a tautology!
- Trail-and-error specification search for the best pscore
- number of possible matched datasets grows exponentially with N.
- matching can make balance worse for some covariates

Searching for the Correct Propensity Score Model

- People use methods to simplify the problem of searching for the best set of matches: propensity score (pscore)
- This simplification turns the problem into a polynomial time problem: $\mathbf{O}(N^2 log(N))$ for pair matching
- How do we know what is the correct propensity score model?
- Matching on the "correct" pscore obtains covariate balance–it's a tautology!
- Trail-and-error specification search for the best pscore
- number of possible matched datasets grows exponentially with N.
- matching can make balance worse for some covariates

Searching for the Correct Propensity Score Model

- People use methods to simplify the problem of searching for the best set of matches: propensity score (pscore)
- This simplification turns the problem into a polynomial time problem: $\mathbf{O}(N^2 log(N))$ for pair matching
- How do we know what is the correct propensity score model?
- Matching on the "correct" pscore obtains covariate balance–it's a tautology!
- Trail-and-error specification search for the best pscore
- number of possible matched datasets grows exponentially with N.
- matching can make balance worse for some covariates

Searching for the Correct Propensity Score Model

- People use methods to simplify the problem of searching for the best set of matches: propensity score (pscore)
- This simplification turns the problem into a polynomial time problem: $\mathbf{O}(N^2 log(N))$ for pair matching
- How do we know what is the correct propensity score model?
- Matching on the "correct" pscore obtains covariate balance–it's a tautology!
- Trail-and-error specification search for the best pscore
- number of possible matched datasets grows exponentially with N.
- matching can make balance worse for some covariates

Searching for the Correct Propensity Score Model

- People use methods to simplify the problem of searching for the best set of matches: propensity score (pscore)
- This simplification turns the problem into a polynomial time problem: $\mathbf{O}(N^2 log(N))$ for pair matching
- How do we know what is the correct propensity score model?
- Matching on the "correct" pscore obtains covariate balance–it's a tautology!
- Trail-and-error specification search for the best pscore
- number of possible matched datasets grows exponentially with N.
- matching can make balance worse for some covariates

Genetic Matching (GenMatch)

Genetic matching is a new general method for performing multivariate matching. GenMatch:

- algorithmically maximizes the balance of observed potential confounders across matched treated and control units
- uses the pscore (if one has one) and weights covariates so that balance is maximized
- uses an evolutionary search algorithm to determine the weight each covariate is given
- genetic algorithm developed by Sekhon and Mebane (1998) is used.

Genetic Matching (GenMatch)

Genetic matching is a new general method for performing multivariate matching. GenMatch:

- algorithmically maximizes the balance of observed potential confounders across matched treated and control units
- uses the pscore (if one has one) and weights covariates so that balance is maximized
- uses an evolutionary search algorithm to determine the weight each covariate is given
- genetic algorithm developed by Sekhon and Mebane (1998) is used.

Genetic Matching (GenMatch)

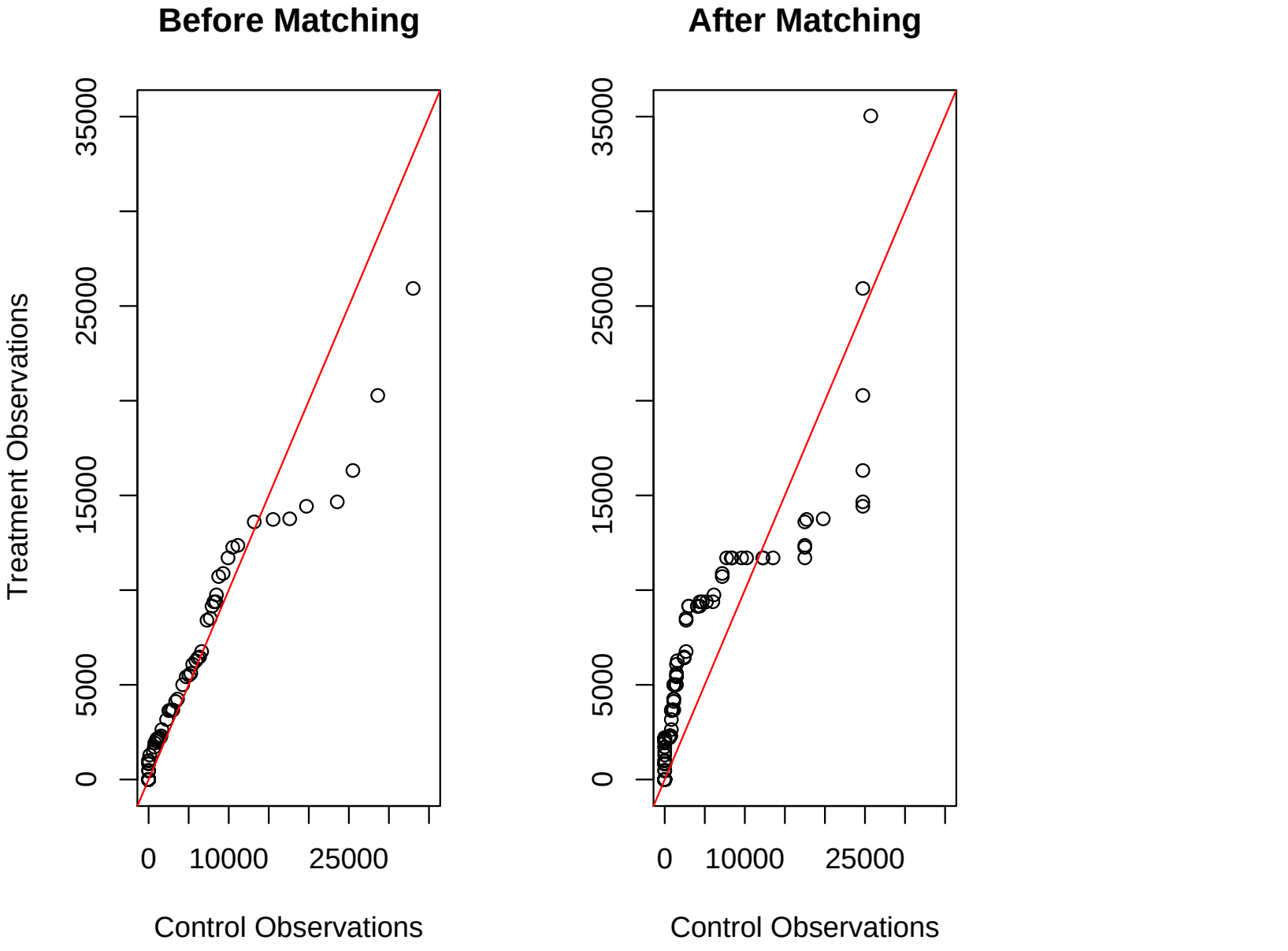
Genetic matching is a new general method for performing multivariate matching. GenMatch:

- algorithmically maximizes the balance of observed potential confounders across matched treated and control units
- uses the pscore (if one has one) and weights covariates so that balance is maximized
- uses an evolutionary search algorithm to determine the weight each covariate is given
- genetic algorithm developed by Sekhon and Mebane (1998) is used.

Pscore Matching Example

```
library(Matching)
data(lalonde)

Y  <- lalonde$re78   #the outcome of interest
Tr <- lalonde$treat  #the treatment of interest

# pscore model
glm1  <- glm(treat~age + educ + black +
             hisp + married + nodegr + re74  + re75,
             family=binomial, data=lalonde)

# Matching
rr1  <- Match(Y=Y, Tr=Tr, X=glm1$fitted)
```

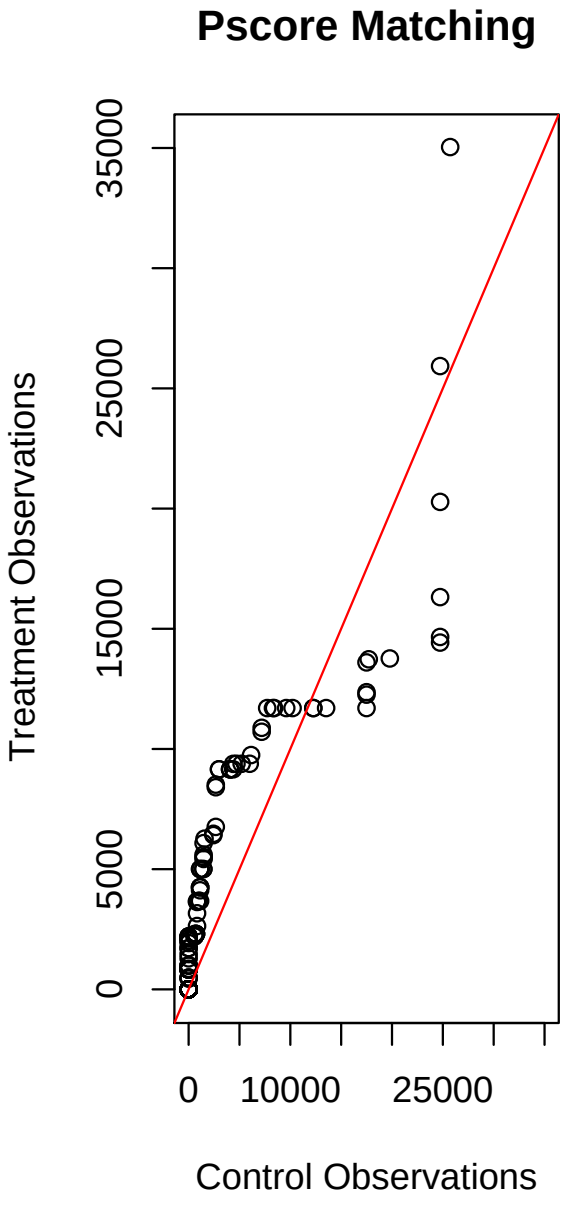
QQ Plot of Before and After Pscore Matching

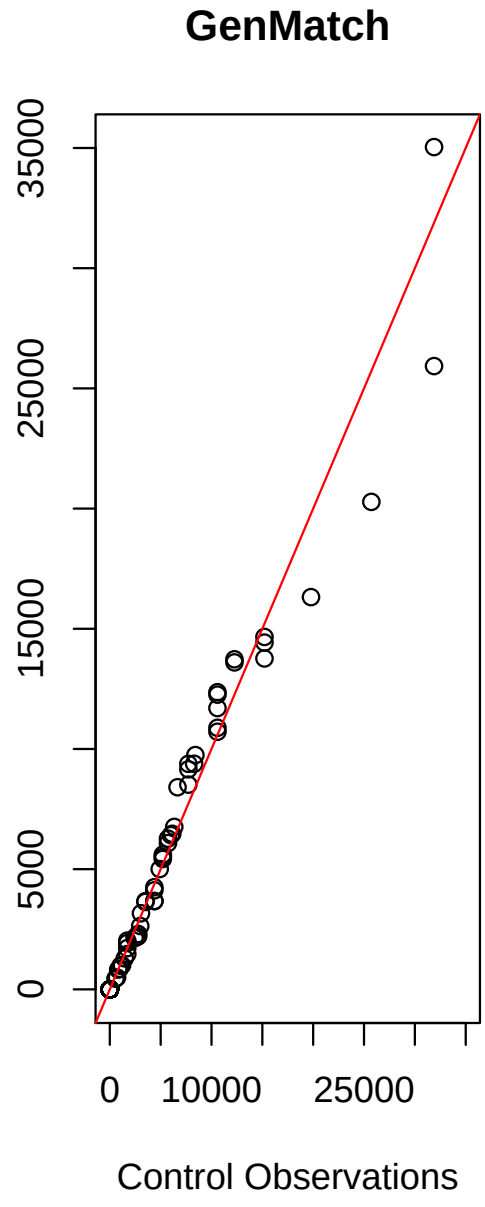


GenMatch Matching Example

```
X  <- cbind(age, educ, black, hisp, married, nodegr,
            re74, re75, u74, u75)

BalanceMatrix <- cbind(age, I(age^2), educ, I(educ^2),
                  black, hisp, married, nodegr, re74 ,
                  I(re74^2), re75, I(re75^2), u74, u75,
                  I(re74*re75), I(age*nodegr),
                  I(educ*re74), I(educ*re75))

gen1 <- GenMatch(Tr=Tr, X=X,
                  BalanceMatrix=BalanceMatrix)

mgen1 <- Match(Y=Y, Tr=Tr, X=X, Weight.matrix=gen1)
```

QQ Plot of Pscore and GenMatch





Properties of Matching Algorithms

- When can matching confounders make bias worse? e.g., what if the propensity score model is incorrect?
- All affinely invariant matching methods have the Equal Percent Bias Reduction (EPBR) property under some conditions.
- If X are distributed with ellipsoidal distributions, then the EPBR property holds for affinely invariant matching methods (Rubin and Thomas 1992).
- There is an extension to a restricted class of mixtures (Rubin and Stuart 2006): discriminant mixtures of proportional ellipsoidally symmetric distributions.

Properties of Matching Algorithms

- When can matching confounders make bias worse? e.g., what if the propensity score model is incorrect?
- All affinely invariant matching methods have the Equal Percent Bias Reduction (EPBR) property under some conditions.
- If X are distributed with ellipsoidal distributions, then the EPBR property holds for affinely invariant matching methods (Rubin and Thomas 1992).
- There is an extension to a restricted class of mixtures (Rubin and Stuart 2006): discriminant mixtures of proportional ellipsoidally symmetric distributions.

Properties of Matching Algorithms

- When can matching confounders make bias worse? e.g., what if the propensity score model is incorrect?
- All affinely invariant matching methods have the Equal Percent Bias Reduction (EPBR) property under some conditions.
- If X are distributed with ellipsoidal distributions, then the EPBR property holds for affinely invariant matching methods (Rubin and Thomas 1992).
- There is an extension to a restricted class of mixtures (Rubin and Stuart 2006): discriminant mixtures of proportional ellipsoidally symmetric distributions.

Properties of Matching Algorithms

- When can matching confounders make bias worse? e.g., what if the propensity score model is incorrect?
- All affinely invariant matching methods have the Equal Percent Bias Reduction (EPBR) property under some conditions.
- If X are distributed with ellipsoidal distributions, then the EPBR property holds for affinely invariant matching methods (Rubin and Thomas 1992).
- There is an extension to a restricted class of mixtures (Rubin and Stuart 2006): discriminant mixtures of proportional ellipsoidally symmetric distributions.

Equal Percent Bias Reduction (EPBR)

- Let Z be the expected value of X in the matched control group. Then we say that a matching procedure is EPBR if

$$E(X|T=1) - Z \quad = \gamma \{E(X|T=1) - E(X|T=0)\}$$

  for a scalar $0 \leq \gamma \leq 1$.
- We say that a matching method is EPBR for X because the percent reduction in the mean biases for each of the matching variables is the same.
- In general, if a matching method is not EPBR, then the bias for some linear function of X is increased.
- We may care about nonlinear functions of X.

Equal Percent Bias Reduction (EPBR)

- Let Z be the expected value of X in the matched control group. Then we say that a matching procedure is EPBR if

$$E(X|T=1) - Z \quad = \gamma \{E(X|T=1) - E(X|T=0)\}$$

  for a scalar $0 \leq \gamma \leq 1$.
- We say that a matching method is EPBR for X because the percent reduction in the mean biases for each of the matching variables is the same.
- In general, if a matching method is not EPBR, then the bias for some linear function of X is increased.
- We may care about nonlinear functions of X.

Equal Percent Bias Reduction (EPBR)

- Let Z be the expected value of X in the matched control group. Then we say that a matching procedure is EPBR if

$$E(X|T=1) - Z \quad = \gamma \{E(X|T=1) - E(X|T=0)\}$$

  for a scalar $0 \leq \gamma \leq 1$.
- We say that a matching method is EPBR for X because the percent reduction in the mean biases for each of the matching variables is the same.
- In general, if a matching method is not EPBR, then the bias for some linear function of X is increased.
- We may care about nonlinear functions of X.

Genetic Matching Computationally Intensive

- Raessler and Rubin (2005) use GenMatch for a dataset with over 2 million observations
- Political get-out-the-vote databases also have millions of observations
- Marketing data has millions of observations and thousands of covariates
- finding the best matches by brute force is a computational problem whose time increases exponentially with N. Traveling Salesman problem

Genetic Matching Computationally Intensive

- Raessler and Rubin (2005) use GenMatch for a dataset with over 2 million observations
- Political get-out-the-vote databases also have millions of observations
- Marketing data has millions of observations and thousands of covariates
- finding the best matches by brute force is a computational problem whose time increases exponentially with N. Traveling Salesman problem

Genetic Matching Computationally Intensive

- Raessler and Rubin (2005) use GenMatch for a dataset with over 2 million observations
- Political get-out-the-vote databases also have millions of observations
- Marketing data has millions of observations and thousands of covariates
- finding the best matches by brute force is a computational problem whose time increases exponentially with N. Traveling Salesman problem

Genetic Matching Computationally Intensive

- Raessler and Rubin (2005) use GenMatch for a dataset with over 2 million observations
- Political get-out-the-vote databases also have millions of observations
- Marketing data has millions of observations and thousands of covariates
- finding the best matches by brute force is a computational problem whose time increases exponentially with N. Traveling Salesman problem

To Optimize the Software

- Computationally intensive functions are written in C/C++
- Parallelize the outer loop and vectorize the inner loop
- Extensive use of BLAS and Lapack libraries
- The GenMatch function is parallelized, it can make use of multiple CPUs or nodes via the snow package: simple network of workstations
- Parallel execution is tricky: unexpected bottlenecks such as a cache-bottleneck when executing SSE3 instructions via BLAS
- Must pay attention to unexpected performance problems: memory allocation (malloc) on OS X

To Optimize the Software

- Computationally intensive functions are written in C/C++
- Parallelize the outer loop and vectorize the inner loop
- Extensive use of BLAS and Lapack libraries
- The GenMatch function is parallelized, it can make use of multiple CPUs or nodes via the snow package: simple network of workstations
- Parallel execution is tricky: unexpected bottlenecks such as a cache-bottleneck when executing SSE3 instructions via BLAS
- Must pay attention to unexpected performance problems: memory allocation (malloc) on OS X

To Optimize the Software

- Computationally intensive functions are written in C/C++
- Parallelize the outer loop and vectorize the inner loop
- Extensive use of BLAS and Lapack libraries
- The GenMatch function is parallelized, it can make use of multiple CPUs or nodes via the snow package: simple network of workstations
- Parallel execution is tricky: unexpected bottlenecks such as a cache-bottleneck when executing SSE3 instructions via BLAS
- Must pay attention to unexpected performance problems: memory allocation (malloc) on OS X

To Optimize the Software

- Computationally intensive functions are written in C/C++
- Parallelize the outer loop and vectorize the inner loop
- Extensive use of BLAS and Lapack libraries
- The GenMatch function is parallelized, it can make use of multiple CPUs or nodes via the snow package: simple network of workstations
- Parallel execution is tricky: unexpected bottlenecks such as a cache-bottleneck when executing SSE3 instructions via BLAS
- Must pay attention to unexpected performance problems: memory allocation (malloc) on OS X

To Optimize the Software

- Computationally intensive functions are written in C/C++
- Parallelize the outer loop and vectorize the inner loop
- Extensive use of BLAS and Lapack libraries
- The GenMatch function is parallelized, it can make use of multiple CPUs or nodes via the snow package: simple network of workstations
- Parallel execution is tricky: unexpected bottlenecks such as a cache-bottleneck when executing SSE3 instructions via BLAS
- Must pay attention to unexpected performance problems: memory allocation (malloc) on OS X

To Optimize the Software

- Computationally intensive functions are written in C/C++
- Parallelize the outer loop and vectorize the inner loop
- Extensive use of BLAS and Lapack libraries
- The GenMatch function is parallelized, it can make use of multiple CPUs or nodes via the snow package: simple network of workstations
- Parallel execution is tricky: unexpected bottlenecks such as a cache-bottleneck when executing SSE3 instructions via BLAS
- Must pay attention to unexpected performance problems: memory allocation (malloc) on OS X

Using Multiple Computer Chips to Run `GenMatch`

	1 CPU	2 CPUs	3 CPUs	4 CPUs
1780 Observations				
run time (seconds)	2557	1372	950	749
x CPU/1 CPU run time		0.54	0.37	0.29
1335 Observations				
run time (seconds)	826	475	317	255
x CPU/1 CPU run time		0.58	0.38	0.31
890 Observations				
run time (seconds)	532	338	233	193
x CPU/1 CPU run time		0.64	0.44	0.36

Using Multiple Computer Chips to Run `GenMatch`

	1 CPU	2 CPUs	3 CPUs	4 CPUs
1780 Observations				
run time (seconds)	2557	1372	950	749
x CPU/1 CPU run time		0.54	0.37	0.29
1335 Observations				
run time (seconds)	826	475	317	255
x CPU/1 CPU run time		0.58	0.38	0.31
890 Observations				
run time (seconds)	532	338	233	193
x CPU/1 CPU run time		0.64	0.44	0.36

Using Multiple Computer Chips to Run `GenMatch`

	1 CPU	2 CPUs	3 CPUs	4 CPUs
1780 Observations				
run time (seconds)	2557	1372	950	749
x CPU/1 CPU run time		0.54	0.37	0.29
1335 Observations				
run time (seconds)	826	475	317	255
x CPU/1 CPU run time		0.58	0.38	0.31
890 Observations				
run time (seconds)	532	338	233	193
x CPU/1 CPU run time		0.64	0.44	0.36

The Joy of Malloc

- Matching had far worse performance on OS X, why?
- Memory allocation issue: malloc
- OS X malloc makes system call at 15KB while Linux does at 256KB.
- On OS X, more page faults
- Rewrite functions to make fewer malloc calls (at the cost of more memory usage). Apple's OS X performance group helped with this.
- Use Doug Lea's malloc instead

The Joy of Malloc

- Matching had far worse performance on OS X, why?
- Memory allocation issue: malloc
- OS X malloc makes system call at 15KB while Linux does at 256KB.
- On OS X, more page faults
- Rewrite functions to make fewer malloc calls (at the cost of more memory usage). Apple's OS X performance group helped with this.
- Use Doug Lea's malloc instead

The Joy of Malloc

- Matching had far worse performance on OS X, why?
- Memory allocation issue: malloc
- OS X malloc makes system call at 15KB while Linux does at 256KB.
- On OS X, more page faults
- Rewrite functions to make fewer malloc calls (at the cost of more memory usage). Apple's OS X performance group helped with this.
- Use Doug Lea's malloc instead

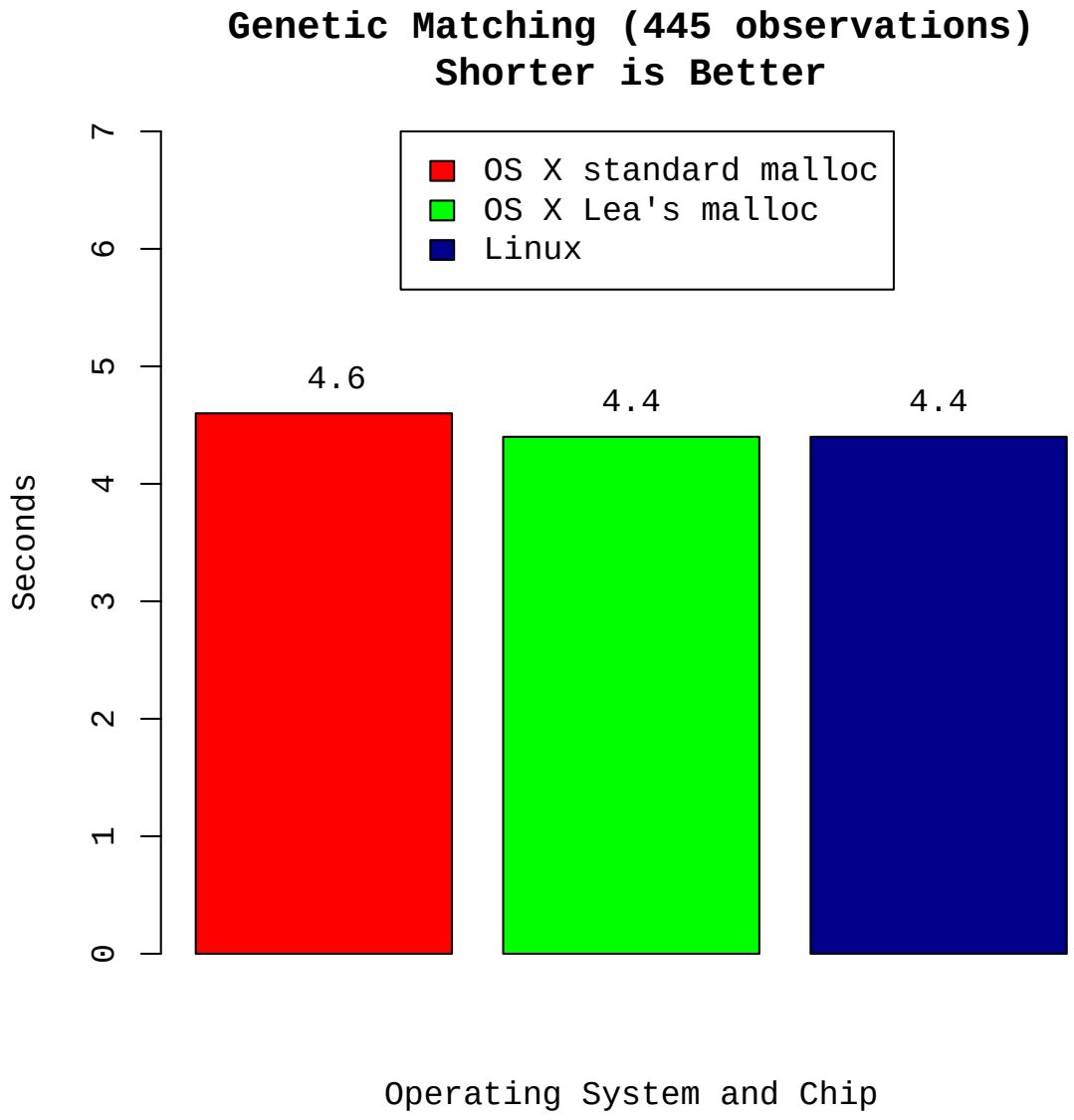
The Joy of Malloc

- Matching had far worse performance on OS X, why?
- Memory allocation issue: malloc
- OS X malloc makes system call at 15KB while Linux does at 256KB.
- On OS X, more page faults
- Rewrite functions to make fewer malloc calls (at the cost of more memory usage). Apple's OS X performance group helped with this.
- Use Doug Lea's malloc instead

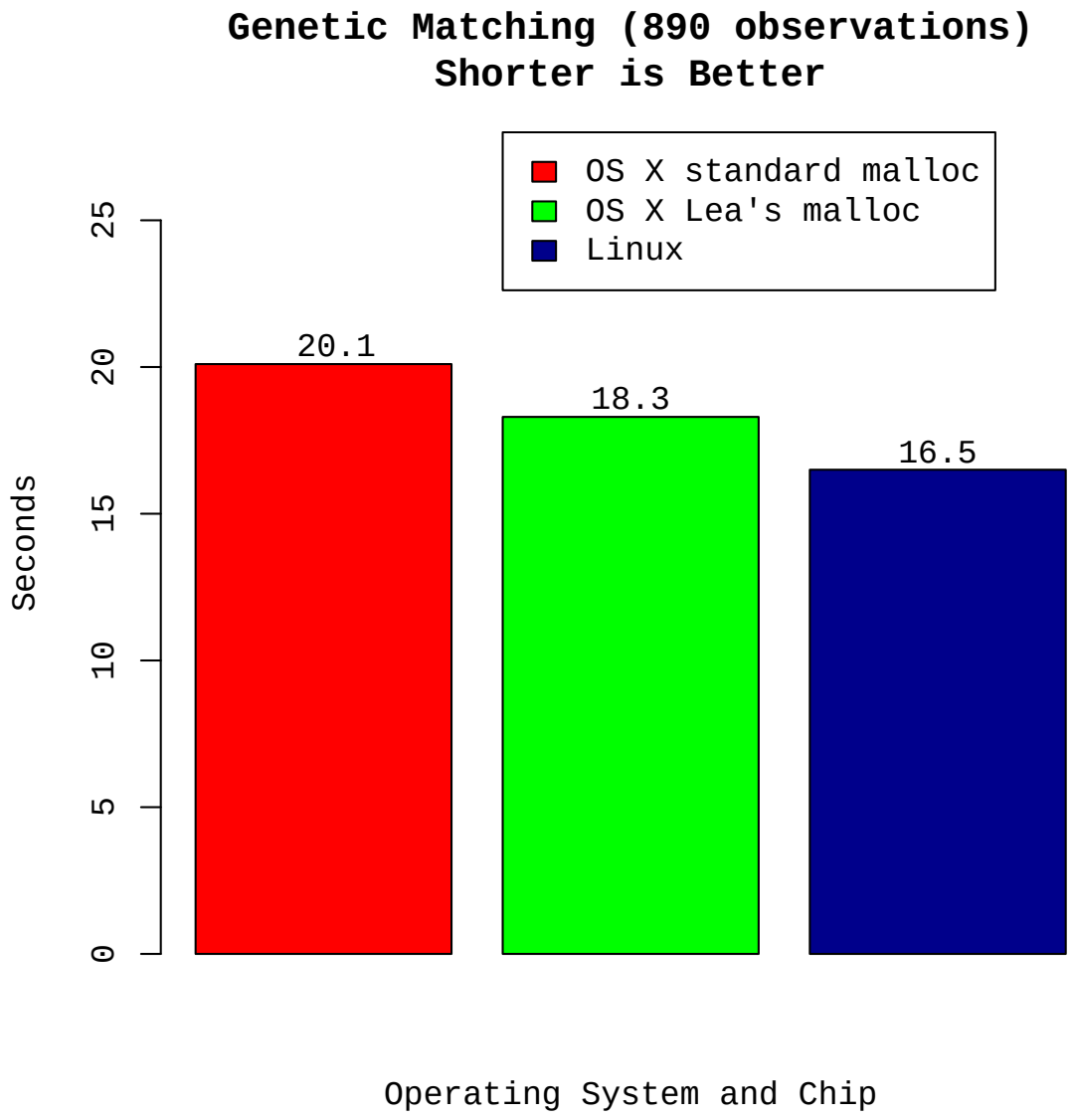
The Joy of Malloc

- Matching had far worse performance on OS X, why?
- Memory allocation issue: malloc
- OS X malloc makes system call at 15KB while Linux does at 256KB.
- On OS X, more page faults
- Rewrite functions to make fewer malloc calls (at the cost of more memory usage). Apple's OS X performance group helped with this.
- Use Doug Lea's malloc instead

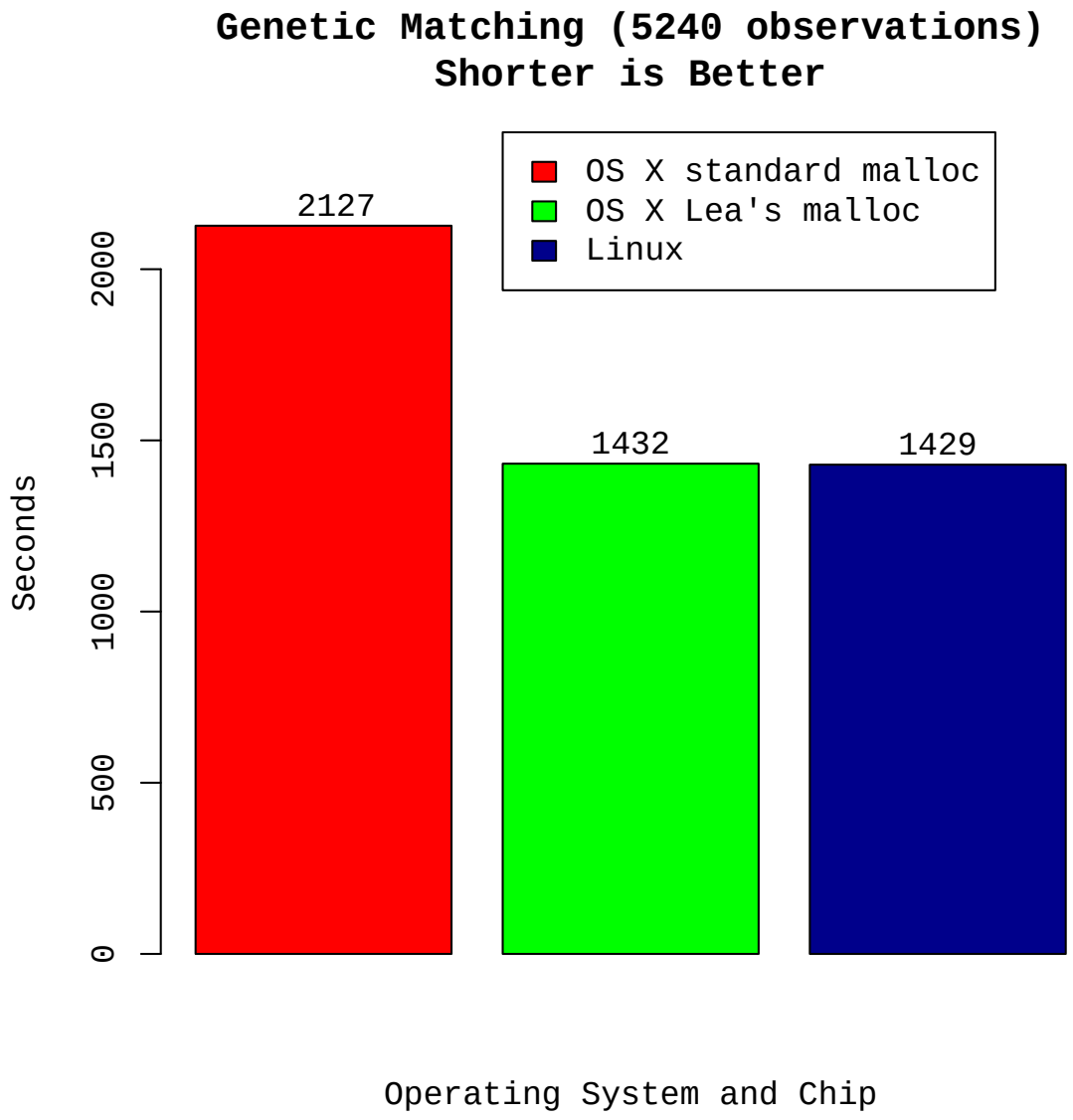
The Joy of Malloc



The Joy of Malloc



The Joy of Malloc



Quantum Computing

File Edit View History Bookmarks Tools Help

D-Wave
The Quantum Computing Company™

- Company >
- Technology >
- Applications >

WELCOME TO D-WAVE SYSTEMS

D-Wave Systems is the world's first and only source of quantum computing for commercial applications. We believe quantum technology in concert with classical, digital processors, can and will represent broad and significant advancements in the application of computer science.

In February 2007, D-Wave unveiled and demonstrated this technology publicly for the first time. The company plans to deliver field-deployable systems in 2008.

HIGHLIGHTS

D-Wave, Your Partner
Solving problems faster and more accurately gives rise to exciting new opportunities for channel partners, system integrators, and the open-source community.

D-Wave, the Technology
Get an insider's view of what makes it the next big thing — on a small (quantum) scale. Find out how you can capitalize on digital/quantum co-processing.

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Until Quantum Computing

- Parallel computing and vectorization
- Block-algorithms: the Matchby function
- A wide variety of matching procedures are supported, and in the future `optmatch` will be also
- Design issue for R:
  - The malloc problem impacts BLAS performance
  - Block-algorithms: R interface with SQL
  - Software engineering support...offload this to Parrot/JAVA?
  - Support to extend R internally is needed
- Software available at `http://SEKHON.BERKELEY.EDU` and on CRAN

Mahalanobis Distance

- The most common method of multivariate matching is based on the Mahalanobis distance. The Mahalanobis distance measure between any two column vectors is defined as:

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

  where X_i and X_j are two different observations and S is the sample covariance matrix of X.
- Mahalanobis distance is an appropriate distance measure if each covariate has an elliptic distribution whose shape is common between treatment and control groups (Mitchell and Krzanowski 1985, 1989).
- In finite samples, Mahalanobis distance will not be optimal.

Mahalanobis Distance

- The most common method of multivariate matching is based on the Mahalanobis distance. The Mahalanobis distance measure between any two column vectors is defined as:

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

  where X_i and X_j are two different observations and S is the sample covariance matrix of X.
- Mahalanobis distance is an appropriate distance measure if each covariate has an elliptic distribution whose shape is common between treatment and control groups (Mitchell and Krzanowski 1985, 1989).
- In finite samples, Mahalanobis distance will not be optimal.

More General Method of Measuring Distance

- A more general way to measure distance is defined by:

$$d(X_i, X_j) = \left\{ (X_i - X_j)' \left(S^{-1/2}\right)' W S^{-1/2} (X_i - X_j) \right\}^{\frac{1}{2}}$$

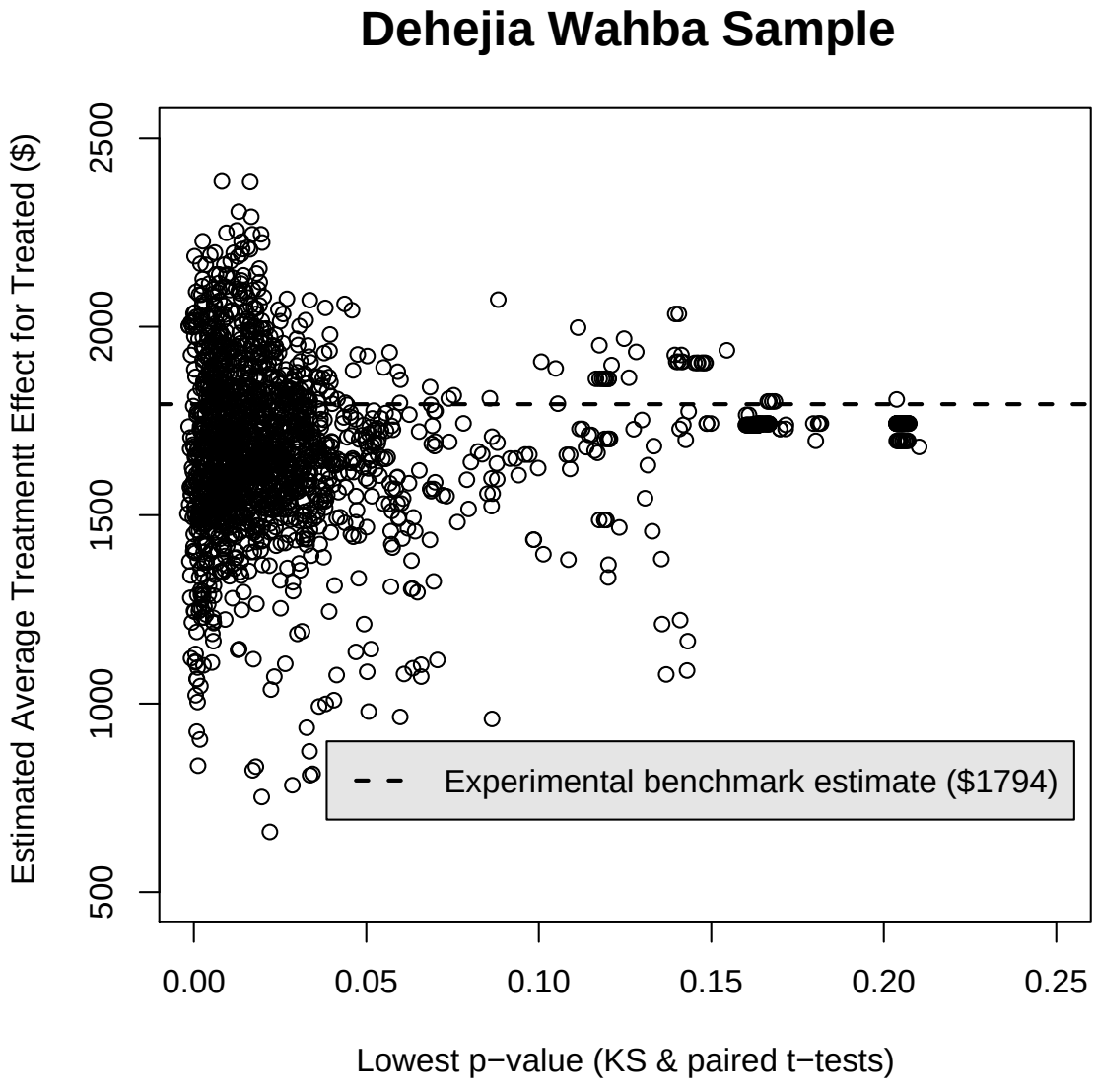
  where W is a $k \times k$ positive definite weight matrix and $S^{1/2}$ is the Cholesky decomposition of S which is the variance-covariance matrix of X.
- All elements of W are zero except down the main diagonal. The main diagonal consists of k parameters which must be chosen.
- This leaves the problem of choosing the free elements of W. For identification, there are only $k - 1$ free parameters.

More General Method of Measuring Distance

- A more general way to measure distance is defined by:

$$d(X_i, X_j) = \left\{ (X_i - X_j)' \left(S^{-1/2}\right)' W S^{-1/2} (X_i - X_j) \right\}^{\frac{1}{2}}$$

  where W is a $k \times k$ positive definite weight matrix and $S^{1/2}$ is the Cholesky decomposition of S which is the variance-covariance matrix of X.
- All elements of W are zero except down the main diagonal. The main diagonal consists of k parameters which must be chosen.
- This leaves the problem of choosing the free elements of W. For identification, there are only $k - 1$ free parameters.


Dehejia Wahba Sample
Estimated Average Treatmentt Effect for Treated ($)
500
1000
1500
2000
2500
0.00
0.05
0.10
0.15
0.20
0.25
Lowest p−value (KS & paired t−tests)
Experimental benchmark estimate ($1794)

Parameterization

- GenMatch uses the propensity score if it is known or if it can be estimated.
- The propensity score is estimated and its linear predictor, $\hat{\mu}$, is matched upon along with the covariates X once they have been adjusted so as to be uncorrelated with the linear predictor.
- Combining is good because:
  - Propensity score matching is good at minimizing the discrepancy along the propensity score
  - Mahalanobis distance is good at minimizing the distance between individual coordinates of X (orthogonal to the propensity score) (Rosenbaum and Rubin 1985).

Parameterization

- GenMatch uses the propensity score if it is known or if it can be estimated.
- The propensity score is estimated and its linear predictor, $\hat{\mu}$, is matched upon along with the covariates X once they have been adjusted so as to be uncorrelated with the linear predictor.
- Combining is good because:
  - Propensity score matching is good at minimizing the discrepancy along the propensity score
  - Mahalanobis distance is good at minimizing the distance between individual coordinates of X (orthogonal to the propensity score) (Rosenbaum and Rubin 1985).

Parameterization

- GenMatch uses the propensity score if it is known or if it can be estimated.
- The propensity score is estimated and its linear predictor, $\hat{\mu}$, is matched upon along with the covariates X once they have been adjusted so as to be uncorrelated with the linear predictor.
- Combining is good because:
  - Propensity score matching is good at minimizing the discrepancy along the propensity score
  - Mahalanobis distance is good at minimizing the distance between individual coordinates of X (orthogonal to the propensity score) (Rosenbaum and Rubin 1985).

Optimization

- Many loss functions are possible. Such as:
  - minimize the largest discrepancy
  - minimize the mean or median discrepancy
  - minimize some other quantile
  - restrict the above to only uniformly improving moves
- By default, the algorithm attempts to minimize the largest discrepancy at every step (minimizing the infinity norm).
- For a given set of matches resulting from a given W, the loss is defined as the maximum of the cumulative probability distribution functions of a variety of standardized statistics—i.e., the minimum "p-value" observed across a series of balance "tests."

Optimization

- Many loss functions are possible. Such as:
  - minimize the largest discrepancy
  - minimize the mean or median discrepancy
  - minimize some other quantile
  - restrict the above to only uniformly improving moves
- By default, the algorithm attempts to minimize the largest discrepancy at every step (minimizing the infinity norm).
- For a given set of matches resulting from a given W, the loss is defined as the maximum of the cumulative probability distribution functions of a variety of standardized statistics—i.e., the minimum "p-value" observed across a series of balance "tests."

Optimization

- Many loss functions are possible. Such as:
  - minimize the largest discrepancy
  - minimize the mean or median discrepancy
  - minimize some other quantile
  - restrict the above to only uniformly improving moves
- By default, the algorithm attempts to minimize the largest discrepancy at every step (minimizing the infinity norm).
- For a given set of matches resulting from a given W, the loss is defined as the maximum of the cumulative probability distribution functions of a variety of standardized statistics—i.e., the minimum "p-value" observed across a series of balance "tests."

Optimization

- Many loss functions are possible. Such as:
  - minimize the largest discrepancy
  - minimize the mean or median discrepancy
  - minimize some other quantile
  - restrict the above to only uniformly improving moves
- By default, the algorithm attempts to minimize the largest discrepancy at every step (minimizing the infinity norm).
- For a given set of matches resulting from a given W, the loss is defined as the maximum of the cumulative probability distribution functions of a variety of standardized statistics—i.e., the minimum "p-value" observed across a series of balance "tests."

Optimization

- Many loss functions are possible. Such as:
  - minimize the largest discrepancy
  - minimize the mean or median discrepancy
  - minimize some other quantile
  - restrict the above to only uniformly improving moves
- By default, the algorithm attempts to minimize the largest discrepancy at every step (minimizing the infinity norm).
- For a given set of matches resulting from a given W, the loss is defined as the maximum of the cumulative probability distribution functions of a variety of standardized statistics—i.e., the minimum "p-value" observed across a series of balance "tests."

Optimization

- Many loss functions are possible. Such as:
  - minimize the largest discrepancy
  - minimize the mean or median discrepancy
  - minimize some other quantile
  - restrict the above to only uniformly improving moves
- By default, the algorithm attempts to minimize the largest discrepancy at every step (minimizing the infinity norm).
- For a given set of matches resulting from a given W, the loss is defined as the maximum of the cumulative probability distribution functions of a variety of standardized statistics—i.e., the minimum "p-value" observed across a series of balance "tests."

Measuring Balance

- There are many different ways of measuring balance and I cannot summarize the vast literature here.
- The choice of balance statistics will be domain specific. E.g., use randomization inference if you can as Bowers and Hansen (2005) do.
- But the measures should be sensitive to different departures from balance.
- It is important the maximum discrepancy be small. "*p*-values" conventionally understood to signal balance (e.g., 0.10) are often too low to produce reliable estimates.
- the *p*-values from these balance "tests" cannot be interpreted as true probabilities

Measuring Balance

- There are many different ways of measuring balance and I cannot summarize the vast literature here.
- The choice of balance statistics will be domain specific. E.g., use randomization inference if you can as Bowers and Hansen (2005) do.
- But the measures should be sensitive to different departures from balance.
- It is important the maximum discrepancy be small. "*p*-values" conventionally understood to signal balance (e.g., 0.10) are often too low to produce reliable estimates.
- the *p*-values from these balance "tests" cannot be interpreted as true probabilities

Measures of Balance

- Algorithm uses cumulative probability distribution functions of a variety of standardized statistics which are often thought to be test statistics
- Formal hypothesis tests of balance should not be conducted because no measure of balance is a monotonic function of bias and because balance should be optimized without limit.
- However, descriptive measures of discrepancy ignore information related to bias
- By default, *t*-tests for the difference of means and nonparametric bootstrap Kolmogorov-Smirnov distributional test statistics are used.
- The analyst may use any measures she desires—e.g., additional nonlinear functions and higher order interactions.

Measures of Balance

- Algorithm uses cumulative probability distribution functions of a variety of standardized statistics which are often thought to be test statistics
- Formal hypothesis tests of balance should not be conducted because no measure of balance is a monotonic function of bias and because balance should be optimized without limit.
- However, descriptive measures of discrepancy ignore information related to bias
- By default, *t*-tests for the difference of means and nonparametric bootstrap Kolmogorov-Smirnov distributional test statistics are used.
- The analyst may use any measures she desires—e.g., additional nonlinear functions and higher order interactions.

Measures of Balance

- Algorithm uses cumulative probability distribution functions of a variety of standardized statistics which are often thought to be test statistics
- Formal hypothesis tests of balance should not be conducted because no measure of balance is a monotonic function of bias and because balance should be optimized without limit.
- However, descriptive measures of discrepancy ignore information related to bias
- By default, *t*-tests for the difference of means and nonparametric bootstrap Kolmogorov-Smirnov distributional test statistics are used.
- The analyst may use any measures she desires—e.g., additional nonlinear functions and higher order interactions.

Measures of Balance

- Algorithm uses cumulative probability distribution functions of a variety of standardized statistics which are often thought to be test statistics
- Formal hypothesis tests of balance should not be conducted because no measure of balance is a monotonic function of bias and because balance should be optimized without limit.
- However, descriptive measures of discrepancy ignore information related to bias
- By default, *t*-tests for the difference of means and nonparametric bootstrap Kolmogorov-Smirnov distributional test statistics are used.
- The analyst may use any measures she desires—e.g., additional nonlinear functions and higher order interactions.

Genetic Optimization

- The optimization problem described above is difficult and irregular, and we utilize an evolutionary algorithm developed by Sekhon and Mebane (1998) called GENOUD.
- Theorems in support of GAs are based on interpreting them as finite and irreducible Markov chains.
- Random search also works better than the usual matching methods, but is less efficient than GENOUD.

Genetic Optimization

- The optimization problem described above is difficult and irregular, and we utilize an evolutionary algorithm developed by Sekhon and Mebane (1998) called GENOUD.
- Theorems in support of GAs are based on interpreting them as finite and irreducible Markov chains.
- Random search also works better than the usual matching methods, but is less efficient than GENOUD.

Issues with R

- The malloc problem effects system BLAS performance
- I could not link R against Dough Lea's malloc (dmalloc) stably
- But dmalloc is used in the Windows port (by B. Ripley)
- But Windows is a different tree!
- I compile Matching against dmalloc
- A better way to extend R is needed

Issues with R

- The malloc problem effects system BLAS performance
- I could not link R against Dough Lea's malloc (dmalloc) stably
- But dmalloc is used in the Windows port (by B. Ripley)
- But Windows is a different tree!
- I compile Matching against dmalloc
- A better way to extend R is needed

Issues with R

- The malloc problem effects system BLAS performance
- I could not link R against Dough Lea's malloc (dmalloc) stably
- But dmalloc is used in the Windows port (by B. Ripley)
- But Windows is a different tree!
- I compile Matching against dmalloc
- A better way to extend R is needed

Issues with R

- The malloc problem effects system BLAS performance
- I could not link R against Dough Lea's malloc (dmalloc) stably
- But dmalloc is used in the Windows port (by B. Ripley)
- But Windows is a different tree!
- I compile Matching against dmalloc
- A better way to extend R is needed

Bowers, Jake and Ben Hansen. 2005. "Attributing Effects to A Cluster Randomized Get-Out-The-Vote Campaign." Technical Report #448, Statistics Department, University of Michigan. `http://www-personal.umich.edu/~jwbowers/PAPERS/bowershansen2006-10TechReport.pdf`.

Mitchell, Ann F. S. and Wojtek J. Krzanowski. 1985. "The Mahalanobis Distance and Elliptic Distributions." *Biometrika* 72 (2): 464–467.

Mitchell, Ann F. S. and Wojtek J. Krzanowski. 1989. "Amendments and Corrections: The Mahalanobis Distance and Elliptic Distributions." *Biometrika* 76 (2): 407.

Raessler, S. and D. B. Rubin. 2005. "Complications when using nonrandomized job training data to draw causal inferences." *Proceedings of the International Statistical Institute*.

Rosenbaum, Paul R. and Donald B. Rubin. 1985. "Constructing a Control Group Using Multivariate Matched Sampling Methods That Incorporate the Propensity Score." *The American Statistician* 39 (1): 33–38.

Rubin, Donald B. and Elizabeth A. Stuart. 2006. "Affinely