Nonparametric Distributed Learning Architecture for Big Data: Algorithm and Applications

Scott Bruce, *Student Member, IEEE*, Zeda Li, *Student Member, IEEE*, Hsiang-Chieh Yang, and Subhadeep Mukhopadhyay*, *Member, IEEE*

Abstract—Dramatic increases in the size and complexity of modern datasets have made traditional "centralized" statistical inference prohibitive. In addition to computational challenges associated with big data learning, the presence of numerous data types (e.g. discrete, continuous, categorical, etc.) makes automation and scalability difficult. A question of immediate concern is how to design a data-intensive statistical inference architecture without changing the basic statistical modeling principles developed for "small" data over the last century. To address this problem, we present `MetaLP`, a flexible, distributed statistical modeling framework suitable for large-scale data analysis, where statistical inference meets big data computing. This framework consists of three key components that work together to provide a holistic solution for big data learning: (i) partitioning massive data into smaller datasets for *parallel processing* and efficient computation, (ii) modern nonparametric learning based on a specially designed, orthonormal data transformation leading to *mixed data algorithms*, and finally (iii) combining *heterogeneous* "local" inferences from partitioned data using *meta-analysis* techniques to arrive at the "global" inference for the original big data. We present an application of this general theory in the context of a nonparametric two-sample inference algorithm for Expedia personalized hotel recommendations based on 10 million search result records.

Index Terms—Nonparametric mixed data modeling; LP transformation; Distributed statistical learning; Heterogeneity; Meta-analysis; Data-parallelism.

✦

1 INTRODUCTION

Motivation. Expedia is a large online travel agency and has a strong interest in understanding how user, search, and hotel characteristics influence booking behavior. As a result, Expedia released a dataset [1] containing 52 variables of user and hotel characteristics (e.g. search criteria, hotel features and pricing, user purchase history, competitor pricing, etc.) from 10 million hotel search results collected over a window of the year 2013. These factors will ultimately be used to optimize hotel search results and increase booking rates. For this purpose, we develop a scalable, distributed algorithm that we refer to as `MetaLP`. This learning algorithm can mine search data from millions of travelers, in a completely nonparametric manner, to find important features that best predict customers' likelihood to book a hotel. This is an important large-scale machine learning problem.

The Volume Problem. This kind of "tall" data structure, whose number of observations can run into the millions and billions, frequently arises in astronomy, marketing, neuroscience, e-commerce, and social networks. These massive datasets cannot be stored or analyzed by a single computer all-at-once using standard data analysis software. This creates a major bottleneck for statistical modeling and inference. We seek to develop a framework that allows data scientists to systematically apply the tools and algorithms developed prior to the "age of big data" for massive data problems.

The Variety Problem. Another challenge is in developing a standard algorithm that can work across different data types, known as the mixed data problem [2]. The Expedia dataset contains variables of different types (e.g. continuous, categorical, discrete, etc.), and each requires a different statistical method for inference. A few examples of traditional statistical measures for $(Y; X)$ data include: (1) Pearson's ϕ-coefficient: Y and X both binary, (2) Wilcoxon statistic: Y binary and X continuous, (3) Kruskal-Wallis statistic: Y discrete multinomial and X continuous, and many more. Computational implementation of traditional statistical algorithms for large, mixed data thus become dauntingly complex as they require data type information to calculate the proper statistic. To streamline this process, we need to develop unified computing algorithms that yield appropriate statistical measures without demanding data type information from the user. To achieve this goal, we design a customized, discrete, orthonormal, polynomial-based transformation, the LP-Transformation [3], [4], suitable for arbitrary random variable X. This transformation can be viewed as a nonparametric, data-adaptive generalization of Norbert Wiener's Hermite polynomial chaos-type representation [5]. This easy-to-implement LP-transformation based approach allows us to extend and integrate classical and modern statistical methods for nonparametric feature selection, thus providing the foundation to build automatic algorithms for large, complex datasets.

The Scalability Problem. Finally, the most crucial issue is to develop a scalable algorithm for large datasets, like the Expedia example. With the evolution of big data structures, new processing capabilities relying on distributed, parallel processing have been developed for efficient data

• *Temple University, Department of Statistical Science Philadelphia, PA, 19122.*
*Corresponding Author E-mail: deep@temple.edu

Algorithm	Nonparametric	Inference	Modeling	Speed	Heterogeneity
MetaLP	✓	✓	✓	✓	✓
BLB [9]	✓	✓	✗	✗	✗
SAVGM [10]	✓	✓	✗	✗	✗
KL-Weighting [11]	✗	✓	✓	✗	✗
AEE [12]	✗	✓	✓	✓	✗
Split-and-conquer [13]	✗	✓	✓	✓	✗

TABLE 1: Scope of `MetaLP` and other existing methods.

manipulation and analysis. This paper presents a statistical inference framework for massive data that can fully exploit the power of parallel computing architecture and can be easily embedded into the MapReduce framework [6]. We design the statistical "map" function and "reduce" function for massive data variable selection by integrating many modern statistical concepts and ideas introduced in Sections 2 and 3. Doing so allows for faster processing of big datasets, while maintaining the ability to obtain accurate statistical inference without losing information. Another appealing aspect of our distributed statistical modeling strategy is that it is equally applicable for small and big data, thus providing a unified approach to modeling.

Related Literature. Several statistical distributed learning schemes for big data have been proposed in the literature. The divide and recombine (D&R) approach [7], [8] to the analysis of large, complex data provides a general statistical approach to analyzing big data in a way that is mostly considered embarrassingly parallel. In this setting, communication-efficient algorithms have been developed for various tasks such as assessing estimator quality [9], statistical optimization [10], and model aggregation [11], based on bootstrap resampling and subsampling techniques. Parallel algorithms have also been designed for large-scale parametric linear regression [12], [13]. These proposals address important challenges in analyzing large, complex data, but there are still significant hurdles to clear in developing a holistic framework for big data learning. Table 1 provides a comparison of the `MetaLP` learning framework with these proposals to better illustrate where this work fits into the existing distributed learning landscape.

The `MetaLP` framework broadens the existing scope of big data learning challenges that can be addressed in four important ways. First, the methods of [12], [13] are based on parametric modeling assumptions. However, these assumptions often do not hold when analyzing large, complex data and present automation difficulties, as these assumptions are inherently data type dependent. On the other hand, the `MetaLP` framework is model–free in the sense that it is nonparametric and does not assume any specific model form. Thus, it is more applicable for big data analytics. Second, while the communication-efficient algorithms [9], [10], [11] are flexible to accommodate various statistics and model forms, they require specific instances from the user in order to conduct inference and modeling. In contrast, our `MetaLP` framework relies on statistics based on the LP-transformation for nonparametric inference and modeling due to its favorable theoretical properties and its ability to solve the mixed data problem (see Section 3). Third, `MetaLP` also enjoys a considerable reduction in computation time

compared to other methods which rely on computationally intensive bootstrap resampling [9], [11] and subsampling [10] techniques in order to generate local inferences. Lastly, characteristics across subpopulations may vary significantly, even under purely random data partitioning (see Section 4.2 and Supplementary Section B), which is known as heterogeneity [14]. Using meta-analysis principles, `MetaLP` assigns optimal weights to each local inference, which properly accounts for any potential heterogeneity. These weights then determine the influence of each local inference on the final global inference. This approach provides a crucial advantage over methods relying on equal weighting schemes where heterogeneity can spoil inference (see Sections 2.3 and 4.2). In summary, this work provides the basis to develop a general and systematic massive data analysis framework that can simultaneously perform nonparametric statistical modeling and inference and can be adapted for a variety of learning problems.

Organization. Section 2 provides the basic statistical formulation and overview of the `MetaLP` algorithmic framework. Section 3 covers details of the individual elements of the distributed statistical learning framework, addressing the important issue of heterogeneity in big data along with a concrete nonparametric parallelizable variable selection algorithm. Section 4 evaluates the effectiveness of our proposed variable selection algorithm and compares it with other popular methods through simulation studies. Section 5 provides an in-depth analysis of the motivating Expedia dataset using the framework to conduct variable selection under different settings to determine which hotel and user characteristics influence booking behavior. Section 6 provides some concluding remarks and discusses the future direction of this work. Supplementary materials are also available discussing two examples on how the `MetaLP` framework provides a new understanding and resolution for problems related to Simpson's Paradox and Stein's Paradox, the relevance of `MetaLP` for small-data, and the `R` scripts for MapReduce implementation.

2 STATISTICAL FORMULATION OF BIG DATA ANALYSIS

Our research is motivated by a real business problem of optimizing personalized web marketing for Expedia with the goal of improving customer experience and look-to-book ratios[1] by identifying key factors that affect consumer

---

1. The number of people who visit a travel agency web site compared to the number who make a purchase. This ratio measures the effectiveness of an agency in securing purchases.

choices. This prototypical digital marketing case study allows us to address the following more general data modeling challenge, which finds its applicability in many areas of modern data-driven science, engineering, and business:

How can we design nonparametric distributed algorithms that work on large amounts of data (that cannot be stored or processed by just one machine) to find the most important features that affect certain outcomes?

At first glance, this may look like a simple two-sample inference problem that can be solved by some trivial generalization of existing 'small-data' statistical methods, but in reality, this is not the case. In this article we perform a thorough investigation of the theoretical and practical challenges present in big data analysis. We emphasize the role of statistics in big data analysis and provide an overview of the three main components of our statistical theory along with the modeling challenges they are designed to overcome. In what follows, we present the conceptual building blocks of `MetaLP`, a large-scale distributed learning tool that allows big data users to run statistical procedures on large amounts of data. Figure 1 outlines the architecture.
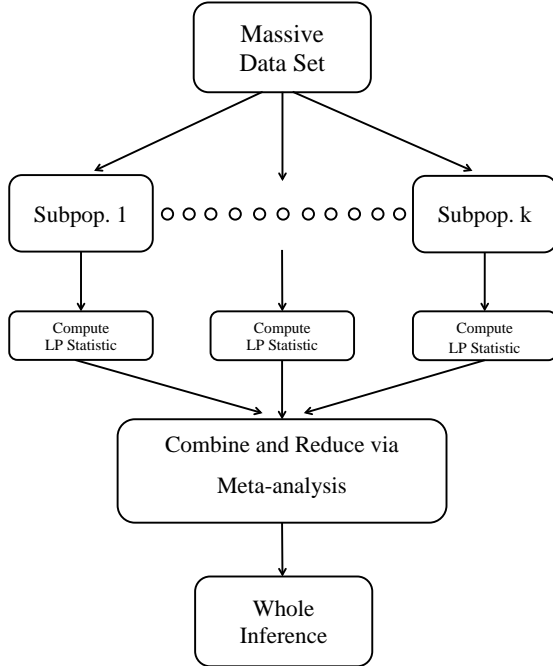

Fig. 1: `MetaLP` large-scale distributed statistical inference architecture.

2.1 Partitioning Massive Datasets

Dramatic increases in the size of datasets have created a major bottleneck for conducting statistical inference in a traditional, 'centralized' manner, where we have access to the full data. The first, and quite natural, idea to tackle the volume problem is to divide the big data into several smaller datasets, similar to modern parallel computing database systems like Hadoop and Spark as illustrated in Figure 2. However, simply dividing the dataset does not allow data scientists to conquer the problem of big data

analysis. There are many unsettled questions that we have to carefully address using proper statistical tools to arrive at an appropriate solution.

Users must select a data partitioning scheme to divide the original large data into several smaller parts and assign them to different nodes for processing. The most common technique is random partitioning. However, users may choose other strategies, like spatial or temporal partitioning, in order to use the inherent structure of the data. Also, the original massive dataset may already be partitioned by some natural grouping variable, in which case an algorithm that can accommodate pre-existing partitions is desirable. The number of partitions could also be defined by the user who may consider a wide range of cost metrics including the number of processors required, CPU time, job latency, memory utilization, and more.

Another important, and often overlooked, consideration when choosing a partitioning scheme is that the characteristics of the subpopulations created may vary largely. This is known as heterogeneity [14] and is an unavoidable obstacle for divide-and-conquer style inference models. Heterogeneity can certainly impact data-parallel inference, so we incorporate diagnostics to measure the severity of the problem (see Section 3.5) and data-adaptive regularization to adjust effect size estimates accordingly (see Section 3.6). This allows users to detect the presence of heterogeneity in a given data partition and offers robustness to various data partitioning options in the estimation.

2.2 LP Statistics for Mixed Data

Massive datasets typically contain a multitude of data types, and the Expedia dataset is no exception. Figure 2 shows three predictor variables with different data types in the Expedia dataset: `promotion_flag` (binary), `srch_length_of_stay` (discrete count), and `price_usd` (continuous). In order to construct appropriate statistical measures for identifying important variables, traditional algorithmic approaches demand two pieces of information: (1) values and (2) data type information for every variable. This requirement produces considerable complications in computational implementation and creates serious roadblocks for building systematic and automatic algorithms for large, complex data. Thus, the question of immediate concern is:

How can we develop a unified computing formula, with automatic built-in adjustments, that yields appropriate statistical measures, without requiring data type information from the user?

To tackle this 'data variety' or 'mixed data' problem, we design a custom-constructed, discrete, orthonormal, polynomial-based transformation, called LP-Transformation. This data transformation provides a generic and universal representation of any random variable, defined in Section 3.1. We use this transformation technique to represent the data in a new LP Hilbert space. This data-adaptive transformation will allow us to construct unified learning algorithms by compactly expressing them as inner products in the LP Hilbert space.

2.3 Combining Information via Meta-Analysis

Eventually, the goal of having a distributed inference procedure critically depends on the question:

Subpopulation 1

booking_bool	promotion_flag	srch_length_of_stay	price_usd
1	0	2	164.59
0	1	7	284.48
1	0	1	194.34
⋮	⋮	⋮	⋮
0	1	3	371.27

•
•
•

Subpopulation k

booking_bool	promotion_flag	srch_length_of_stay	price_usd
1	1	1	125.65
1	0	3	149.32
0	1	1	224.46
⋮	⋮	⋮	⋮
1	1	3	174.89

Fig. 2: Illustration of a partitioned data set with k subpopulations and various data types. Three variables in the *Expedia* dataset are shown. The target variable Y, booking_bool, indicates whether or not the hotel was booked. The three predictor variables shown are X_1 promotion_flag (indicates if a sale price promotion was displayed), X_2 srch_length_of_stay (search criterion for number of nights stayed), and X_3 price_usd (displayed hotel price).

How to judiciously combine the 'local' inferences executed in parallel by different servers to get the 'global' inference for the original big data?

To resolve this challenge, we make a novel connection with meta-analysis. Section 3.2 describes how we can use meta-analysis to parallelize the statistical inference process for massive datasets. Furthermore, we seek to provide a distribution estimator for the LP-statistics via a confidence distribution (CD) that contains information for virtually all types of statistical inference (e.g. estimation, hypothesis testing, confidence intervals, etc.). Section 3.4 discusses the use of CD-based meta-analysis, which plays a key role in integrating local inferences to construct a comprehensive answer for the original data. These new connections allow data scientists to fully utilize the parallel processing power of large-scale clusters for designing unified and efficient big data statistical inference algorithms.

To conclude, we have discussed the architectural overview of MetaLP, which addresses the challenge of developing an inference framework for data-intensive applications without requiring modifications to the core statistical principles developed for 'small' data. Next, we describe the theoretical underpinnings, algorithmic foundation, and implementation details of our data-parallel, large-scale MetaLP inference model.

3 ELEMENTS OF DISTRIBUTED STATISTICAL LEARNING

In this section, we introduce the key concepts of our proposed method by connecting several classical and modern statistical ideas to develop a comprehensive inference framework. We highlight along the way how these new ideas and connections address the real challenges of big data analysis as noted in Section 2.

3.1 LP United Statistical Algorithm and Universal Representation

To address the mixed data problem, we introduce a nonparametric statistical modeling framework based on an LP approach to data analysis [3].

Data Transformation and LP Hilbert Functional Space Representation. Our approach relies on an alternative representation of the data in the LP Hilbert space, which will be defined shortly. The new representation shows how each explanatory variable, regardless of data type, can be represented as a linear combination of *data-adaptive* orthogonal LP basis functions. This data-driven transformation will allow us to construct unified learning algorithms in the LP Hilbert space. Many traditional and modern statistical measures developed for different data types can be compactly expressed as inner products in the LP Hilbert space. The following is the fundamental result for LP basis function representation.

Theorem 3.1 (LP representation). *Random variable X (discrete or continuous) with finite variance admits the following decomposition: $X - \mathbb{E}(X) = \sum_{j>0} T_j(X; X) \, \mathbb{E}[X T_j(X; X)]$ with probability 1.*

$T_j(X; X)$, for $j = 1, 2, \ldots$, are score functions constructed by Gram Schmidt orthornormalization of the powers of $T_1(X; X) = \mathcal{Z}(F^{\mathrm{mid}}(X; X))$. Where $\mathcal{Z}(X) = (X - \mathbb{E}[X])/\sigma(X)$, $\sigma^2(X) = \mathrm{Var}(X)$, and the mid-distribution transformation of a random variable X is defined as

$$F^{\mathrm{mid}}(x; X) = F(x; X) - .5p(x; X) \tag{1}$$

where $p(x; X) = \Pr[X = x]$, $F(x; X) = \Pr[X \leq x]$. We construct the LP score functions on $0 < u < 1$ by letting $x = Q(u; X)$, where $Q(u; X) = \inf\{x : F(x) \geq u\}$ is the quantile function of the random variable X and

$$S_j(u; X) = T_j(Q(u; X); X). \tag{2}$$

Why is it called the LP-basis? Note that our specially designed basis functions vary naturally according to data type unlike the fixed Fourier and wavelet bases as shown in Figure 3. Note an interesting similarity of the shapes of LP score functions and shifted Legendre Polynomials for the *continuous* feature `price_usd`. In fact, as the number of distinct values of a random variable $A(X) \to \infty$ (moving from discrete to continuous data type), the shape converges to smooth Legendre Polynomials. To emphasize this universal limiting shape, we call it an **Legendre-Polynomial-like** (**LP**) orthogonal basis. For any general X, LP-polynomials are piecewise-constant orthonormal functions over $[0, 1]$, as shown in Figure 3. This data-driven property makes the LP transformation uniquely advantageous in constructing a generic algorithmic framework to tackle the mixed data problem.

Constructing Measures by LP Inner Product. Define the two-sample LP statistic for variable selection of a *mixed random* variable X (either continuous or discrete) based on our specially designed score functions

$$\begin{aligned} \text{LP}[j; X, Y] &= \mathbb{E}[T_j(X; X)T_1(Y; Y)], \\ &= \text{Cor}[T_j(X; X), Y]. \end{aligned} \tag{3}$$

To prove Equation (3), which expresses our variable selection statistic as an LP-inner product measure, verify the following for Y binary,

$$\mathcal{Z}(y; Y) = T_1(y; Y) = \begin{cases} -\sqrt{\dfrac{p}{1 - p}} & \text{for } y = 0 \\ \sqrt{\dfrac{1 - p}{p}} & \text{for } y = 1. \end{cases}$$

LP statistic properties. Using empirical process theory, we can show that the sample LP measures $\sqrt{n}\widehat{\text{LP}}[j; X, Y]$, asymptotically converge to i.i.d. standard normal distributions [3].

As an example of the power of LP-unification, we describe $\widehat{\text{LP}}[1; X, Y]$ that systematically reproduces all the traditional linear statistical variable selection measures for different data types of X. Note that the nonparametric Wilcoxon method to test the equality of two distributions can equivalently be represented as $\text{Cor}(\mathbb{I}\{Y = 1\}, F^{\text{mid}}(X; X))$, which leads to the following important alternative LP representation result.

Theorem 3.2. *Two sample Wilcoxon Statistic W can be computed as*

$$W(X, Y) = \widehat{\text{LP}}[1; X, Y]. \tag{4}$$

Our computing formula for the Wilcoxon statistic using $\widehat{\text{LP}}[1; X, Y]$ offers automatic *adjustments for data with ties*; hence, no further tuning is required. Furthermore, if we have X and Y both binary (i.e. data from the two variables can be represented in a 2×2 table), then we have
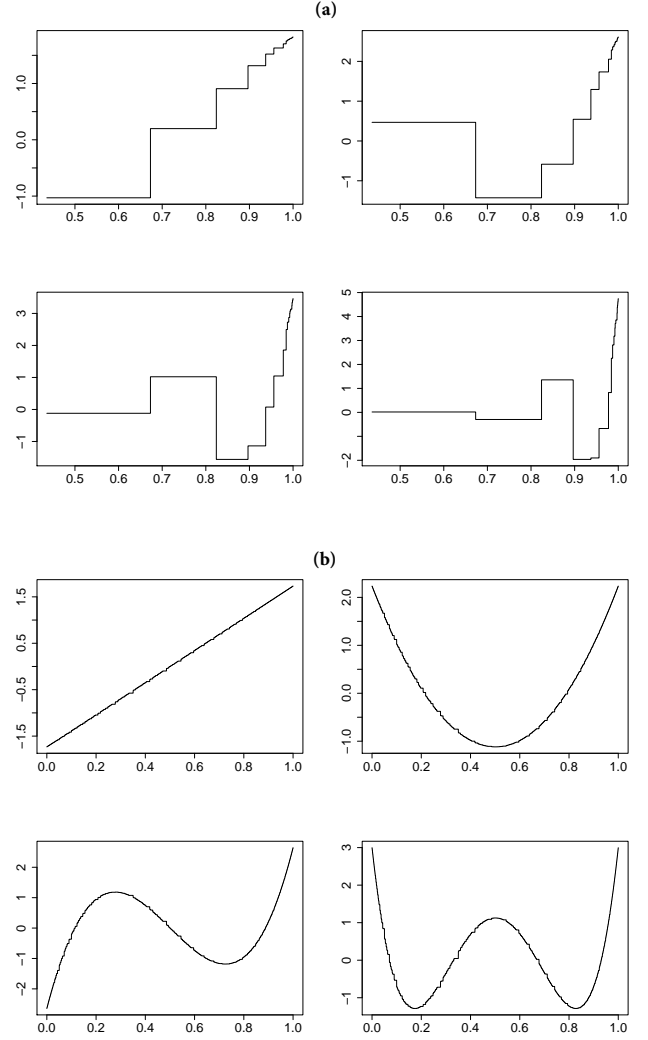

Fig. 3: (a) Top 2×2 panel shows the shape of the first four LP orthonormal score functions for the variable `length_of_stay`, which is a discrete random variable taking values $0, \ldots, 8$; (b) Bottom 2×2 panel shows the shape of the LP score functions for the continuous variable `price_usd`.

$$T_1(0; X) = -\sqrt{P_{2+}/P_{1+}}, \quad T_1(1; X) = \sqrt{P_{1+}/P_{2+}}$$
$$T_1(0; Y) = -\sqrt{P_{+2}/P_{+1}}, \quad T_1(1; Y) = \sqrt{P_{+1}/P_{+2}}, \tag{5}$$

where $P_{i+} = \sum_j P_{ij}$ and $P_{+j} = \sum_i P_{ij}$, and P_{ij} denotes the entry for the ith row and jth column of the 2×2 probability table, and

$$\begin{aligned} \widehat{\text{LP}}[1; X, Y] &= \mathbb{E}[T_1(X; X)T_1(Y; Y)], \\ &= \sum_{i=1}^{2}\sum_{j=1}^{2} P_{ij}T_1(i - 1; X)\,T_1(j - 1; Y), \\ &= (P_{11}P_{22} - P_{12}P_{21})/(P_{1+}P_{+1}P_{2+}P_{+2})^{1/2}. \end{aligned} \tag{6}$$

Following result summarizes the observation in (6).

Theorem 3.3. *For a 2×2 contingency table with Pearson correlation ϕ, we have,*

$$\phi(X, Y) = LP[1; X, Y]. \tag{7}$$

Beyond Linearity. Higher order Wilcoxon statistics are LP statistics of higher order score functions, $T_j(X; X)$, which detect *distributional* differences as in variability, skewness, or tail behavior for two different classes. The LP statistics $LP[j; X, Y]$ for $j > 1$ can capture how the distribution of a variable changes over classes and is applicable for mixed data types.

To summarize, LP statistics allow data scientists to write a *single* computing formula for any variable X, irrespective of its data type, with a *common* metric and asymptotic characteristics. This leads to a huge practical benefit in designing a unified method for combining distributed 'local' inferences without requiring data type information for the variables.

3.2 Meta-Analysis and Data-Parallelism

The objective of this section is to provide a new way of thinking about the problem: how to appropriately combine 'local' inferences to derive reliable and robust conclusions for the original large dataset? This turns out to be one of the most crucial, and heavily neglected, aspects of data-intensive modeling that decides the fate of big data inference. Here we introduce the required statistical framework that can answer the key question: how to compute individual weights for each partitioned dataset? Our framework adopts the concept of meta-analysis to provide a general recipe for constructing such algorithms for large-scale parallel computing. This will allow us to develop statistical algorithms that can balance computational speed and statistical accuracy.

Brief Background on Meta-Analysis. Meta-analysis [15] is a statistical technique by which information from independent studies is assimilated, which has its origins in clinical settings. It was developed primarily to combat the problem of under-powered "small data" studies. A key benefit of this approach is the aggregation of information leading to improved statistical power as opposed to less precise inference derived from a single study. A huge amount of literature exists on meta-analysis, including a careful review of recent developments [16], which includes 281 references.

Relevance of Meta-analysis for big data inference? Unlike the classical situation, we don't have statistical power issues for big data problems. However, we are unable to analyze the whole dataset all-at-once using a single machine in a classical inferential setup. We apply meta-analysis from a completely different perspective and motivation, as a tool to facilitate distributed inference for massive datasets. This novel connection provides a statistically sound mechanism to combine "local" inferences by determining the optimal weighting strategy [15].

We partition big data systematically into several subpopulations over a distributed database, estimate parameters of interest in each subpopulation separately, and then combine results using meta-analysis as demonstrated in Figure 1. Thus, meta-analysis provides a way to pool information from subpopulations and produce a singular, powerful combined inference for the original large dataset. In some circumstances, the dataset may already be partitioned (e.g. each group could be an image or a large text document) and stored in different servers based on some reasonable partitioning scheme. Our distributed statistical framework can work with these predefined groupings as well by combining them using the meta-analysis framework to arrive at the final combined inference.

We call this statistical framework, which utilizes both LP statistics and meta-analysis methodology, as `MetaLP`, and it consists of two parts: (i) the LP statistical map function or algorithm (that tackles the "variety" problem), and (ii) the meta-analysis methodology for merging the information from all subpopulations to get the final inference.

3.3 Confidence Distribution and LP Statistic Representation

The Confidence Distribution (CD) is a distribution estimator, rather than a point or interval estimator, for a particular parameter of interest. From the CD, all traditional forms of statistical estimation and inference (e.g. point estimation, confidence intervals, hypothesis testing) can be produced. Moreover, CDs can be utilized within the meta-analysis framework, as we will show in the next section. More specifically, the CD is a sample-dependent distribution function on the parameter space that can represent confidence intervals of all levels for a parameter of interest.

While the CD was first defined in [17], [18] extended the concept to asymptotic confidence distributions (aCDs). A comprehensive review of the concept can be found in [19].

Definition 3.1. Suppose Θ is the parameter space for an unknown parameter of interest, θ, and ω is the sample space corresponding to data $\mathbf{X}_n = \{X_1, X_2, \ldots, X_n\}^T$. Then a function $H_n(\cdot) = H_n(\mathbf{X}, \cdot)$ on $\omega \times \Theta \rightarrow [0, 1]$ is a confidence distribution (CD) if: (i) for each given $\mathbf{X}_n \in \omega$, $H_n(\cdot)$ is a continuous cumulative distribution function on Θ; (ii) at the true parameter value $\theta = \theta_0$, $H_n(\theta_0) = H_n(\mathbf{X}, \theta_0)$, as a function of the sample $\mathbf{X}_n$, following the uniform distribution $U[0, 1]$. The function $H_n(\cdot)$ is an asymptotic CD (aCD) if the $U[0, 1]$ requirement holds only asymptotically for $n \rightarrow \infty$ and the continuity requirement on $H_n(\cdot)$ can be relaxed.

The CD is a function of both a random sample and the parameter of interest. The additional requirement in (i) is that for each sample, the CD should be a distribution function on the parameter space. The $U[0, 1]$ requirement in (ii) allows us to construct confidence intervals from a CD easily, meaning that $(H_n^{-1}(\alpha_1), H_n^{-1}(1-\alpha_2))$ is a $100(1-\alpha_1-\alpha_2)\%$ confidence interval for the parameter θ_0 for any $\alpha_1 > 0$, $\alpha_2 > 0$, and $\alpha_1 + \alpha_2 < 1$.

Generally, the CD can easily be derived from the stochastic internal representation [20] of a random variable and a pivot, $\Psi(S, \theta)$. The distribution of the pivot should not depend on the parameter, θ, where θ is the parameter of interest and S is a statistic derived from the data. Here, we derive the CD for the LP statistic. Suppose $\widehat{LP}[j; X, Y]$ is the estimated jth LP statistic for the predictor variable X and

binary response Y. The limiting asymptotic normality of the empirical LP statistic can be compactly represent as:

$$\mathrm{LP}[j; X, Y] \Big| \widehat{\mathrm{LP}}[j; X, Y] = \widehat{\mathrm{LP}}[j; X, Y] + \frac{Z}{\sqrt{n}}, \quad (8)$$

which is the stochastic internal representation of the LP statistic, similar to the stochastic differential equations representation. Thus, we have the following form of the confidence distribution, which is the cumulative distribution function of $\mathcal{N}(\widehat{\mathrm{LP}}[j; X, Y], 1/n)$:

$$H_\Phi(\mathrm{LP}[j; X, Y]) = \Phi\left(\sqrt{n}\left(\mathrm{LP}[j; X, Y] - \widehat{\mathrm{LP}}[j; X, Y]\right)\right). \quad (9)$$

The above representation satisfies the conditions in the CD definition as $n \to \infty$ and therefore is the asymptotic CD of $\mathrm{LP}[j; X, Y]$.

3.4 Confidence Distribution-based Meta-Analysis

Using the theory presented in Section 3.3, we can estimate the CD for the LP statistics for each of the subpopulations, $H(\mathrm{LP}_\ell[j; X, Y])$, and the corresponding point estimators, $\widehat{\mathrm{LP}}_\ell[j; X, Y]$, for $\ell = 1, \ldots, k$. The next step of our `MetaLP` algorithm is to judiciously combine information contained in the CDs for all subpopulations to arrive at the combined CD, $H^{(c)}(\mathrm{LP}[j; X, Y])$, based on the whole dataset for each specific variable X. The framework relies on a confidence distribution-based approach to meta-analysis [21]. The combining function for CDs across k different studies can be expressed as:

$$\begin{aligned} &H^{(c)}(\mathrm{LP}[j; X, Y]) \\ &= G_c\{g_c(H(\mathrm{LP}_1[j; X, Y]), \ldots, H(\mathrm{LP}_k[j; X, Y]))\}. \end{aligned} \quad (10)$$

The function G_c is determined by the monotonic g_c function: $G_c(t) = P(g_c(U_1, \ldots, U_k) \le t)$, where $U_1, \ldots, U_k$ are independent $U[0, 1]$ random variables. A popular and useful choice for g_c is

$$g_c(u_1, \ldots, u_k) = \alpha_1 F_0^{-1}(u_1) + \ldots + \alpha_k F_0^{-1}(u_k), \quad (11)$$

where $F_0(\cdot)$ is a given cumulative distribution function and $\alpha_\ell \ge 0$, with at least one $\alpha_\ell \ne 0$, are generic weights. $F_0(\cdot)$ could be any distribution function, which highlights the flexibility of the proposed framework. Hence, the following theorem introduces a reasonable proposed form of the combined aCD for $\mathrm{LP}[j; X, Y]$.

Theorem 3.4. *Setting $F_0^{-1}(t) = \Phi^{-1}(t)$ and $\alpha_l = \sqrt{n_\ell}$, where n_ℓ is the size of subpopulation $\ell = 1, \ldots, k$, the following combined aCD for $\mathrm{LP}[j; X, Y]$) follows:*

$$\begin{aligned} &H^{(c)}(\mathrm{LP}[j; X, Y]) \\ &= \Phi\left[\left(\sum_{\ell=1}^k n_\ell\right)^{1/2}\left(\mathrm{LP}[j; X, Y] - \widehat{\mathrm{LP}}^{(c)}[j; X, Y]\right)\right] \end{aligned} \quad (12)$$

with

$$\widehat{\mathrm{LP}}^{(c)}[j; X, Y] = \frac{\sum_{\ell=1}^k n_\ell \widehat{\mathrm{LP}}_\ell[j; X, Y]}{\sum_{\ell=1}^k n_\ell} \quad (13)$$

where $\widehat{\mathrm{LP}}^{(c)}[j; X, Y]$ and $\left(\sum_{\ell=1}^k n_\ell\right)^{-1}$ are the mean and variance respectively of the combined aCD for $\mathrm{LP}[j; X, Y]$.

To prove this theorem, verify that replacing $H(\mathrm{LP}_\ell(j; X, Y))$ by (9) in Equation (10) along with the choice of combining function given in (11), where $F_0^{-1}(t) = \Phi^{-1}(t)$ and $\alpha_\ell = \sqrt{n_\ell}$, we have

$$\begin{aligned} &H^{(c)}(\mathrm{LP}[j; X, Y]) \\ &= \Phi\left[\frac{1}{\sqrt{\sum_{\ell=1}^k n_\ell}} \sum_{\ell=1}^k \sqrt{n_\ell} \frac{\mathrm{LP}[j; X, Y] - \widehat{\mathrm{LP}}_\ell[j; X, Y]}{1/\sqrt{n_\ell}}\right]. \end{aligned}$$

3.5 Diagnostic of Heterogeneity

Heterogeneity is a common issue with divide, combine, and conquer approaches to big data analysis and is caused by different characteristics across subpopulations. This issue is often ignored and can easily spoil the big data discovery process by producing very different statistical estimates, which may not faithfully reflect the original parent dataset. Therefore, we diagnose and quantify the degree to which each variable suffers from heterogeneous subpopulation groupings using the I^2 statistic [14]. Define Cochran's Q statistic:

$$Q = \sum_{\ell=1}^k \alpha_\ell \left(\widehat{\mathrm{LP}}_\ell[j; X, Y] - \widehat{\mathrm{LP}}^{(c)}[j; X, Y]\right)^2, \quad (14)$$

where $\widehat{\mathrm{LP}}_\ell[j; X, Y]$ is the estimated LP-statistic from subpopulation ℓ, α_ℓ is the weight for subpopulation ℓ as defined in Theorem 3.4, and $\widehat{\mathrm{LP}}^{(c)}[j; X, Y]$ is the combined meta-analysis estimator. Compute the I^2 statistic by

$$I^2 = \begin{cases} \frac{Q - (k-1)}{Q} \times 100\% & \text{if } Q > (k-1); \\ 0 & \text{if } Q \le (k-1); \end{cases} \quad (15)$$

where k is the number of subpopulations. As a general rule of thumb, $0\% \le I^2 \le 40\%$ indicates heterogeneity among subpopulations is not severe.

3.6 τ^2 Regularization to Tackle Heterogeneity in Big Data

Variations among the subpopulations impact LP statistic estimates, which are not properly accounted for in the Theorem 3.4 model specification. This is especially severe for big data analysis, as it is very likely that a substantial number of variables may be affected by heterogeneity across subpopulations. To better account for the heterogeneity in our distributed statistical inference framework, following [15], we introduce an additional parameter, τ^2, to account for uncertainty due to heterogeneity across subpopulations. This results in a hierarchical model structure:

$$\widehat{\mathrm{LP}}_\ell[j; X, Y] \Big| \mathrm{LP}_\ell[j; X, Y], s_i \overset{\mathrm{iid}}{\sim} N(\mathrm{LP}_\ell[j; X, Y], s_i^2), \quad (16)$$

$$\mathrm{LP}_\ell[j; X, Y] \Big| \mathrm{LP}[j; X, Y], \tau \overset{\mathrm{iid}}{\sim} N(\mathrm{LP}[j; X, Y], \tau^2), \quad (17)$$

where $\ell = 1, \ldots, k$. This model describes two sources of variability of the LP statistic: variation between different subpopulations, and sampling variability within each subpopulation. Note that when $\tau = 0$, all the subpopulation LP effect size estimates, $\widehat{\mathrm{LP}}_\ell[j; X, Y]$, come from a *single*, homogeneous distribution. Thus, when I^2 indicates the

| Observations | Partition Parameter | Mean Absolute | Accuracy | | Run Time (seconds) | | Speed |
n	γ	Error ($\times 10^5$)	Full Data	MetaLP	Full Data	MetaLP	Increase
5,000	0.3	79.87		1		0.09	10.9
	0.4	119.69	1	1	0.98	0.08	12.3
	0.5	189.43		1		0.05	19.6
50,000	0.3	10.66		1		0.38	20.9
	0.4	19.69	1	1	7.95	0.20	39.8
	0.5	34.42		1		0.13	61.2
500,000	0.3	1.56		1		2.02	41.4
	0.4	3.20	1	1	83.66	1.01	82.8
	0.5	6.20		1		0.67	124.9
1,000,000	0.3	0.89		1		3.36	53.2
	0.4	1.86	1	1	178.65	1.59	112.4
	0.5	3.19		1		1.17	152.7

TABLE 2: Comparison of estimation and run times for full data and `MetaLP` LP statistic estimation. Mean absolute error is reported for the Meta LP estimates of the full data LP statistics across all variables. Accuracy is defined as the proportion of replications correctly selecting the true model variables, $\{X_1, X_2, X_3\}$. Run times are reported along with the speed increase using the distributed `MetaLP` approach.

presence of "excess" variability among $\{\widehat{LP}_1, \ldots, \widehat{LP}_k\}$, beyond random fluctuation alone, it is important to introduce the second layer in (17) to account for that heterogeneity. See Sections 4.2 and 5.4 for more discussion on this topic.

Under the new model specification, the CD of the LP statistic for the ℓ-th group is $H(LP_\ell[j; X, Y]) = \Phi((LP[j; X, Y] - \widehat{LP}_\ell[j; X, Y])/(\tau^2 + s_\ell^2)^{1/2})$ where $s_\ell = 1/\sqrt{n_\ell}$. The following theorem provides the form of the combined aCD under this specification.

Theorem 3.5. *Setting $F_0^{-1}(t) = \Phi^{-1}(t)$ and $\alpha_\ell = 1/\sqrt{(\tau^2 + (1/n_\ell))}$, where n_ℓ is the size of subpopulation $\ell = 1, \ldots, k$, the following combined aCD for $LP[j; X, Y]$ follows:*

$$H^{(c)}(LP[j; X, Y]) =$$
$$\Phi\left[\left(\sum_{\ell=1}^{k} \frac{1}{\tau^2 + (1/n_\ell)}\right)^{1/2} (LP[j; X, Y] - \widehat{LP}^{(c)}[j; X, Y])\right]$$

with

$$\widehat{LP}^{(c)}[j; X, Y]) = \frac{\sum_{\ell=1}^{k}(\tau^2 + (1/n_\ell))^{-1}\widehat{LP}_\ell[j; X, Y]}{\sum_{\ell=1}^{k}(\tau^2 + (1/n_\ell))^{-1}}$$

(18)

where $\widehat{LP}^{(c)}[j; X, Y])$ and $(\sum_{\ell=1}^{k} 1/(\tau^2 + (1/n_\ell)))^{-1}$ are the mean and variance respectively of the combined aCD for $LP[j; X, Y]$.

The proof is similar to that for Theorem 3.4. The DerSimonian and Laird [22] and restricted maximum likelihood estimators of the data-adaptive heterogeneity regularization parameter τ^2 are provided in Supplementary Section E.

4 SIMULATION STUDIES

In our simulation studies, we investigate the performance of our `MetaLP` approach compared with the oracle full data LP estimates, as well as with existing methods. We evaluate the methods from four perspectives: 1) accuracy in estimating the oracle full data LP statistics, 2) ability to correctly classify important variables and noise variables, 3) computational efficiency in terms of run time, and 4) performance under the influence of heterogeneity. The dataset we considered has the form $(\mathbf{X}_i, Y_i) \sim P$, i.i.d, for $i = 1, 2, \ldots, n$, where $\mathbf{X}_i \in \mathbb{R}^p$ and $Y_i \in (0, 1)$. We generate dataset

from the model $Y_i \sim \text{Bernoulli}(P(\beta_1 X_{1i}^2 + \mathbf{X}_{(-1)i}^T \boldsymbol{\beta}_{-1}))$, where $P(u) = \exp(u)/(1 + \exp(u))$. $\mathbf{X}_{-1}$ and $\boldsymbol{\beta}_{-1}$ mean all X's except X_1 and all β's except β_1. We set $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p) = (2, -1.5, 3, 0, \ldots, 0)^T$ to be a p-dimensional coefficient vector, where $p = 50$, and then generate three important variables, X_{1i}, X_{2i}, and X_{3i}, from StudentT(30), Binomial($n = 15, p = 0.1$), and Bernoulli($p = 0.2$) respectively. The remaining noise features are generated from the standard normal distribution.

4.1 Comparison with Full Data Estimates

In this section, we evaluate the ability of the distributed `MetaLP` approach to consistently estimate the oracle full data LP statistics under various partitioning schemes. Comparisons are made in terms of variable selection and run time (see Table 2). Let n be the total number of observations in one dataset, where $n = 5,000, 50,000, 500,000,$ and 1 million. For each setting of n, we generate 100 datasets and randomly partition the dataset with n total observations into $k = \lfloor n^\gamma + 0.5 \rfloor$ subpopulations with roughly equal numbers of observations, where $\gamma = 0.3, 0.4, 0.5$.

Table 2 provides the mean absolute error ($\times 10^5$) for the `MetaLP` LP statistic estimates of the oracle full data LP statistics across all 50 variables. All mean absolute errors are small, indicating estimation using the distributed `MetaLP` approach is consistent with estimation using the whole dataset. Note that errors increase as the number of partitions, k, increase for fixed n. This is expected as the number of observations in each partition decreases as the number of partitions increases for fixed n. However, for fixed k, errors are inversely proportional to the number of observations n. Table 2 also compares the `MetaLP` and oracle full data LP variable selection methods in terms of accuracy in selecting the true model variables, $\{X_1, X_2, X_3\}$, and computation time. Second order LP statistics are used to test for significance for X_1, since it has a second order impact on the dependent variable, and first order LP statistics are used to detect significance of other variables. Note that both methods correctly select all the three important variables every time, which suggests that the distributed approach is comparable to the full data approach in selecting important variables. However, our distributed `MetaLP`

approach saves a considerable amount of time compared to the non-distributed approach (i.e. computing LP statistics from the whole dataset all-at-once). We list speed improvements (how many times faster the `MetaLP` algorithm is over the full data approach) in the last column of Table 2. For example, when $n = 1,000,000$ and $\gamma = 0.5$, `MetaLP` is about *150 fold faster*.

4.2 Comparison with Other Methods

In this section, we compare the performance of our proposed `MetaLP` framework with two nonparametric, communication-efficient, distributed inference algorithms: BLB [9] and SAVGM [10]. As noted in Table 1, BLB and SAVGM provide a way to conduct distributed inference for a given estimator provided by users. In order to make a fair comparison, we use empirical LP statistic estimators for BLB and SAVGM methods. We call these methods LP-BLB and LP-SAVGM, respectively, to reflect that they are based on LP statistics. Similar to Section 4.1, we compare the methods based on their abilities to accurately estimate the oracle full data LP statistics, as well as their abilities to differentiate between important and noise variables. We calculate the mean square deviance (MSD) of the distributed LP statistic estimates from the oracle full data LP statistics,

$$\text{MSD} = \frac{1}{R} \sum_{r=1}^{R} \left\{ \widehat{\text{LP}}_r^* - \text{LP}_r^{(\text{full})} \right\}^2, \qquad (19)$$

where R is the number of simulated repetitions, $\widehat{\text{LP}}_r^*$ are the distributed LP statistic estimates for a specific method, and $\text{LP}_r^{(\text{full})}$ are the oracle full data LP statistics.

We use the same model as in Section 4.1 to generate $R = 100$ realizations of the simulated data for each $n = 10,000, 50,000, 100,000$ with $\gamma = 0.3$ in determining the number of subpopulations for all methods. For LP-BLB, we set the number of bootstrap samples taken within each subpopulation to be 100, following [9]. For LP-SAVGM, we fix the sub-sampling ratio to be 0.08. The upper portion of Table 3 summarizes the results.

The relative MSD, $\text{MSD}_{\text{LP-BLB}}/\text{MSD}_{\text{MetaLP}}$ and $\text{MSD}_{\text{LP-SAVGM}}/\text{MSD}_{\text{MetaLP}}$, are all greater than 1, which means `MetaLP` is more accurate on average for all three important variables and sample sizes. The LP-BLB method relies on bootstrap resampling to estimate the distribution of the statistic locally. It has been noted that "*the bootstrap distribution is an approximate confidence distribution*" [19], [23], so there is not much difference in terms of local estimation. Hence, the improvement in MSD of the `MetaLP` method over the LP-BLB method largely comes from the different approach to combining inferences. Rather than weighting each local inference equally, as the LP-BLB method does, `MetaLP` assigns optimal weights to each local inference adjusting for possible heterogeneity. As mentioned previously, even under purely random partitioning with equal sample sizes, heterogeneity may exist (see Supplementary Section B). SAVGM is essentially a bias correction method for divide and recombine estimators. If SAVGM is applied to unbiased estimators, as noted by [10], it could increase the variance of the estimator substantially. This is consistent with our simulation results

Equal Subpopulation Size						
Methods	n	Relative MSD			Mean	Speed
		X_1	X_2	X_3	Extra FD	Increase
LP-BLB	10,000	1.06	1.92	2.22	1.80	125
	50,000	1.06	2.68	2.98	1.75	106
	100,000	1.14	3.46	3.89	1.48	97
LP-SAVGM	10,000	1.46	35.87	39.87	-0.66	1.20
	50,000	1.61	64.01	52.79	-0.29	1.09
	100,000	1.67	99.97	88.61	-0.14	1.05
Unequal Subpopulation Size						
Methods	n	Relative MSD			Mean	Speed
		X_1	X_2	X_3	Extra FD	Increase
LP-BLB	10,000	2.42	16.61	3.45	2.22	121
	50,000	8.50	241.45	4.08	2.86	102
	100,000	12.06	585.41	6.46	4.08	96
LP-SAVGM	10,000	3.88	74.63	42.58	0.29	1.12
	50,000	11.14	323.22	317.83	1.31	1.08
	100,000	23.02	727.88	909.04	1.41	1.05

TABLE 3: Comparison of methods in estimating full data LP statistics and variable selection. Relative MSD (e.g. $\text{MSD}_{\text{LP-BLB}}/\text{MSD}_{\text{MetaLP}}$) compares the accuracy in estimating the oracle full data LP statistics. Mean extra false discovery (FD) is the average number of additional noise variables selected by other methods compared to `MetaLP`. Speed increase captures how many times faster the `MetaLP` algorithm runs compare to other methods. Upper portion: under equal subpopulation size; lower portion: under unequal subpopulation size.

indicating LP–SAVGM performs significantly worse than `MetaLP` in terms of MSD.

In terms of the accuracy in variable selection, all methods correctly select the three important variables on every run. Extra mean false discovery (FD) is the average number of additional noise variables incorrectly determined to be important by the LP-BLB/LP-SAVGM methods compared to the `MetaLP` method. For example, when $n = 50,000$, the LP-BLB method, on average, falsely selects 1.75 additional noise variables compared to the `MetaLP` method. It should be noted that LP-SAVGM performs well in terms of selecting fewer noise variables due to the inflated variance of the LP statistic estimates.

Computational savings is a crucial consideration for big data analysis. Note that `MetaLP` is around 100 times faster than LP-BLB. The additional LP-BLB run time comes from the need to resample each subpopulation numerous times in order to obtain the bootstrap estimate, while our approach calculates the LP estimates for each subpopulation in one shot. The LP-SAVGM is relatively comparable to `MetaLP`, where the additional run time is due to the need for sub-sampling to perform the bias correction.

Next, we investigate the impacts of heterogeneity on the different methods. For this exercise, we will define heterogeneity in terms of varying subpopulation sizes, letting the size increase linearly. In particular, we set the size of the first subpopulation to be 500 and increase the size by 150 for the second subpopulation, and so on. The lower portion of Table 3 shows that the relative MSD increases dramatically, indicating that heterogeneity has substantial negative impacts on both LP-BLB and LP-SAVGM, while `MetaLP` remains robust under this setting. It also should

be noted that, unlike in the equal size case, `MetaLP` tends to outperform LP-SAVGM in terms of mean extra false discoveries, especially when the total sample size is large.

5 EXPEDIA PERSONALIZED HOTEL SEARCH DATASET

Based on `MetaLP`, in this section we develop a model-free, parallelizable, two-sample feature selection algorithm for big data and apply it to the Expedia digital marketing problem. Detailed discussions on each of the following components of our big data two-sample inference model are given in the next sections:

- (Section 5.1) Data Description.
- (Section 5.2) Data Partitioning.
- (Section 5.3) LP Map Function.
- (Section 5.4) Heterogeneity Diagnostic and Regularization.
- (Section 5.5) Meta Reducer via LP Confidence Distribution.
- (Section 5.6) Robustness to Size and Number of Subpopulations.

5.1 Data Description

Fig. 4: On top (a) is a snapshot of a search window with search criteria variables; on the bottom (b) is a list of ranked hotels returned by Expedia with hotel characteristic variables.

Expedia provided a large dataset ($n = 9,917,530$) of hotel search results collected over a window of the year 2013 [1] in order to better understand the factors that influence booking behavior. Data are generated by online

customers who first provide search criteria for their desired travel plans (e.g. length of stay, destination, number of children, etc.) to the Expedia website (see Figure 4a). Expedia then returns an ordered list of available hotels along with important hotel information (e.g. hotel name, price, star rating, promotion, etc.) for customers to review and consider booking for their travel plans (see Figure 4b). In the background, Expedia also records important user information (e.g. visitor location, search history, etc.) and competitor pricing and availability for hotels listed, which may impact booking behavior. Expedia then records how customers interact with each hotel listed (e.g. ignored, clicked, booked). We are primarily interested in understanding which factors influence the binary response variable, `booking_bool`, indicating whether the hotel was booked or not. Descriptions of representative variables and data types are provided in Supplementary Section A.

5.2 Data Partitioning

We consider two different partitioning schemes that are appropriate for the Expedia dataset: 1) random partitioning, which results in homogeneous, similarly sized subpopulations, and 2) predefined partitioning, which results in heterogeneous, disproportionately sized subpopulations.

Step 1. We randomly assign search lists, which are collections of observations with the same search id in the dataset, to 200 different subpopulations. Random assignment of search lists rather than individual observations ensures that sets of hotels viewed in the same search session are all contained in the same subpopulation. Note that the number of subpopulations chosen can be adapted to meet the processing and time requirements of different users. We show in Section 5.6 that our method is robust to different numbers of subpopulations as the inference remains unchanged.

There may be situations where natural groupings exist in the dataset, which can be directly used to form subpopulations. For example, the available Expedia data could be grouped naturally by the country where each visitor to the Expedia website resides, `visitor_location_country_id`.

Our framework can directly utilize these predetermined subpopulations for processing rather than requiring the massive data to be gathered and randomly assigned to subpopulations. However, this partitioning scheme may result in heterogeneous subpopulations, so extra steps must be taken to address this issue as described in Section 5.4. For the Expedia dataset, Figure 5 shows the number of observations for the 20 largest subpopulations from partitioning by `visitor_location_country_id`. The top three largest countries by number of observations contain 74% of the total observations, and the leading country contains almost 50% of the total observations. On the other hand, random partitioning results in roughly equal sample sizes across subpopulations (50,000 each).

5.3 LP Map Function

We tackle the data variety problem by developing automated mixed data algorithms using LP statistical data modeling tools.
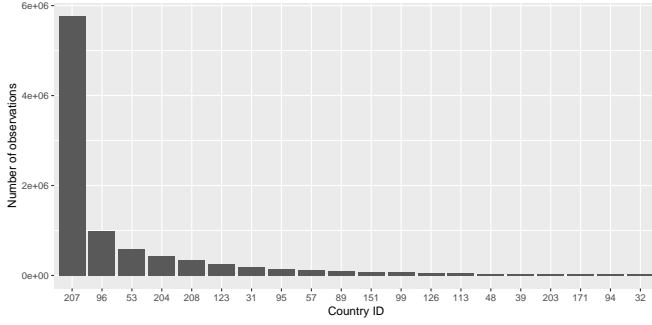
Fig. 5: Number of observations for the 20 largest subpopulations from partitioning by `visitor_location_country_id`.

Step 2. Following the theory in Section 3.1, we construct LP score polynomials, $T_j(x; X_i)$, for each variable based on each partitioned input dataset. Figure 3 shows the LP basis polynomials for variables `variable_length_of_stay` (discrete) and `price_usd` (continuous).

Step 3. Estimate $\mathrm{LP}_\ell[j; X_i, Y]$, which denotes the jth LP statistic for the ith variable in the ℓth subpopulation,

$$\widehat{\mathrm{LP}}_\ell[j; X_i, Y] = n_\ell^{-1} \sum_{k=1}^{n_\ell} T_j(x_k; X_i) T_1(y_k; Y). \quad (20)$$

Step 4. Compute the corresponding LP confidence distribution given by

$$\Phi\left(\sqrt{n}\left(\mathrm{LP}_\ell[j; X_i, Y] - \widehat{\mathrm{LP}}_\ell[j; X_i, Y]\right)\right), \quad (21)$$

for $i = 1, \ldots, 45$ variables across $\ell = 1, \ldots, 200$ random subpopulations (or 233 predefined subpopulations defined by `visitor_location_country_id`).

5.4 Heterogeneity Diagnostic and Regularization

Figure 6 shows the distribution of the first order LP statistic estimates for variable `price_usd` across different subpopulations based on random and `visitor_location_country_id` partitioning. It is clear that random partitioning produces relatively homogeneous LP statistic estimates as the distribution is much more concentrated. On the other hand, `visitor_location_country_id` partitioning results in heterogeneous LP statistic estimates, which is reflected in the dispersion of the corresponding histogram. In fact, the standard deviation of the first order LP statistic under `visitor_location_country_id` partitioning is about 15 times more than that of the random partition, which further highlights the underlying heterogeneity issue. Thus, care must be taken to account for this heterogeneity in a judicious manner that ensures consistent inference. We advocate the method mentioned in Section 3.5.

Step 5. Compute the Cochran's Q-statistic using (14) and I^2 heterogeneity index (15) based on $\mathrm{LP}_1[j; X_i, Y], \ldots, \mathrm{LP}_k[j; X_i, Y]$ for each i and j, where k is the number of subpopulations. Under random partitioning, the subpopulations are fairly homogeneous, with respect to all variables, as all I^2 statistics are below 40% (see Figure 7(a)). However, `visitor_location_country_id` partitioning divides data into heterogeneous subpopulations for
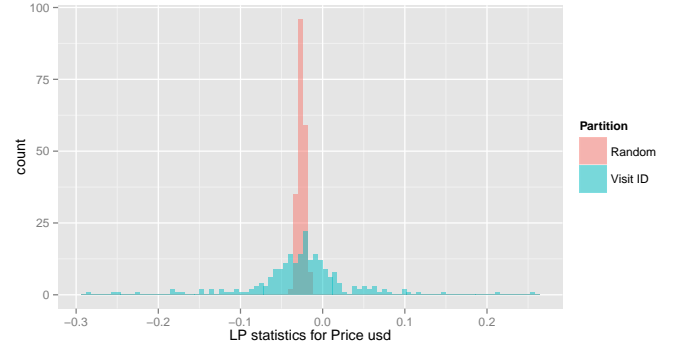

Fig. 6: Distribution of LP statistic estimates for the variable `price_usd` based on random partitioning and `visitor_location_country_id` partitioning.

some variables as shown in Figure 7(b) (i.e. some variables have I^2 values outside the permissible range of 0 to 40% before correction).

Step 6. Compute the DerSimonian and Laird data-driven estimate

$$\hat{\tau}_i^2 = \max\left\{0, \frac{Q_i - (k-1)}{n - \sum_\ell n_\ell^2/n}\right\}, \quad i = 1, \ldots, p.$$

One can also use other enhanced estimators, like the restricted maximum-likelihood estimator, as discussed in Supplementary Section E. I^2 diagnostics *after* correction using τ^2 regularization are shown in Figure 7(b). Note that all I^2 values after correction fall within the acceptable range of 0 to 40%. This result demonstrates that our framework can resolve heterogeneity issues among subpopulations through τ^2 regularization, which protects the validity of the meta-analysis approach.

5.5 Meta Reducer via LP Confidence Distribution

This step combines confidence distribution estimates of LP statistics from different subpopulations to estimate the combined confidence distribution of the LP statistic for each variable as outlined in Section 3.6.

Step 7. Use τ^2-corrected weights to properly take into account the heterogeneity effect. Compute $\widehat{\mathrm{LP}}^{(c)}[j; X, Y])$ by (18) and the corresponding LP confidence distribution using Theorem 3.5.

The resulting 95% confidence intervals for each variable under both random and `visitor_location_country_id` partitioning can be found in Figure 8. Variables with indexes 43, 44, and 45 have highly significant positive relationships with `booking_bool`, the binary response variable. Those variables are `prop_location_score2`, the second score quantifying the desirability of a hotel's location, `promotion_flag`, if the hotel had a sale price promotion specifically displayed, and `srch_query_affinity_score`, the log probability a hotel will be clicked on in Internet searches. There are three variables that have highly negative impacts on hotel booking: `price_usd`, displayed price of the hotel for the given search, `srch_length_of_stay`, number of nights stay that was searched, and `srch_booking_window`,
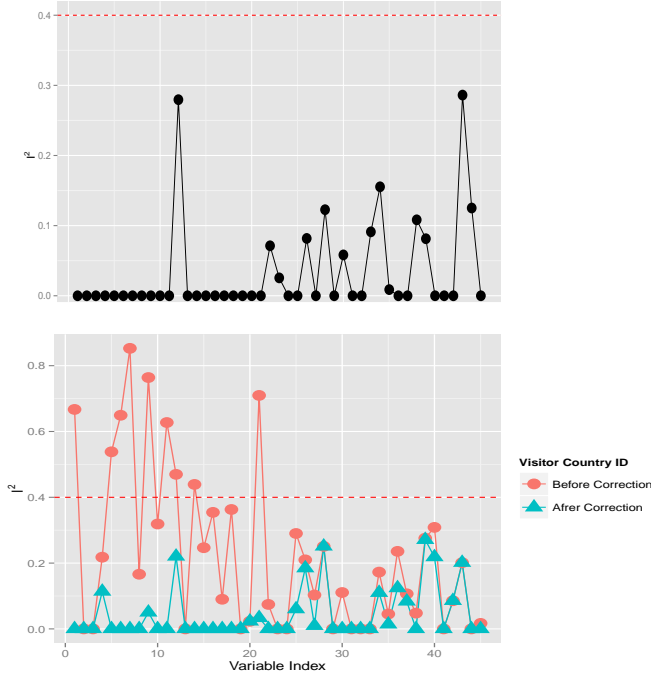
Fig. 7: On top (a) is a plot of the I^2 Diagnostic under random partitioning; on bottom (b) is a comparison of the I^2 diagnostic for the `visitor_location_country_id` partitioning before τ^2 correction and after τ^2 correction.
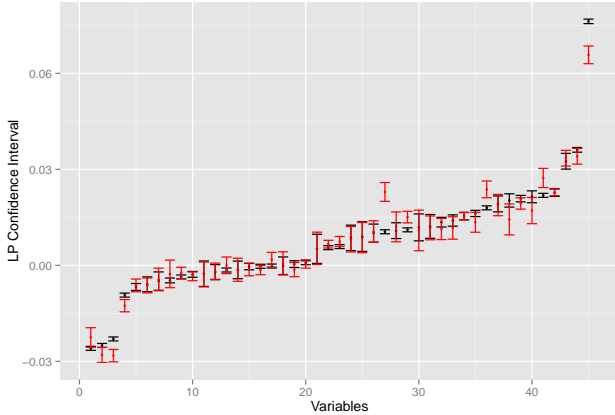

Fig. 8: 95% Confidence intervals for LP statistics for each variable in the Expedia dataset under random partitioning (black) and `visitor_location_country_id` partitioning (red).

number of days in the future the hotel stay started from the search date. Moreover, there are several variables whose LP statistic confidence intervals include zero, which means those variables have an insignificant influence on hotel booking. The top five most influential variables in terms of absolute value of LP statistic point estimates are `prop_location_score2`, `promotion flag`, `srch_query_affinity_score`, `price_usd`, and `srch_length_of_stay` (see Table 4). Intuitively, users are more likely to book hotels with desirable locations (high `prop_location_score2` values), special promotions (`promotion_flag=1`), and high probabilities of being clicked (high `srch_query_affinity_score` values).

The variables we selected are also among the list of top important variables identified by the winners of the ICDM 2013 competition [24], which required participants to develop hotel ranking algorithms for all user search queries based on the features in the Expedia dataset. This speaks to the usefulness of these selected features for downstream analytical tasks (e.g. classification, ranking, etc.).

Note that the confidence intervals for each of the variables under both partitioning schemes are very similar, resulting in similar variable selection outcomes. Four of the top five influential variables identified under random partitioning are also in the top five influential variables identified under `visitor_location_country_id` partitioning (see Table 4). The impact of heterogeneity on the results under `visitor_location_country_id` partitioning can be seen in Figure 8 as the confidence intervals are generally wider than those derived under random partitioning. This can be attributed to extra variability among subpopulations captured by τ^2 due to different characteristics among subpopulations defined by country.

Rank	Random partition	Predetermined partition
1	prop_location_score2	prop_location_score2
2	promotion_flag	promotion_flag
3	srch_query_affinity_score	srch_query_affinity_score
4	price_usd	srch_length_of_stay
5	srch_length_of_stay	srch_booking_window

TABLE 4: Top five influential variables by random partitioning and predetermined partition

5.6 Robustness to Size and Number of Subpopulations

Due to different capabilities of computing systems available to users, users may choose different sizes and numbers of subpopulations for distributed computing. This requires our algorithm to be robust to different numbers and sizes of subpopulations for practical applications. To assess robustness, we compare LP statistic estimates generated from multiple random partitions with different numbers of subpopulations ($k = 50, 100, 150, 200, 250, 300, 350, 400, 450, 500$) for the Expedia dataset. Figure 9 presents LP statistic 95% confidence intervals for three influential variables and three insignificant variables calculated from partitions with varying numbers of subpopulations. Note that the intervals are consistent, even as the number of subpopulations increase (i.e. the number of observations in each subpopulation decrease), which is evidence of stable estimation.

6 FINAL REMARKS

To address the major challenges associated with big data analysis, we have outlined a general theoretical foundation in this article, which we believe may provide the missing link between small data and big data science. Our research shows how the traditional and modern 'small' data modeling tools can be successfully adapted and connected for developing powerful, big data analytic tools by leveraging distributed computing environments.

In particular, we have proposed a nonparametric two sample inference algorithm that has the following two-fold practical significance for solving real-world data mining
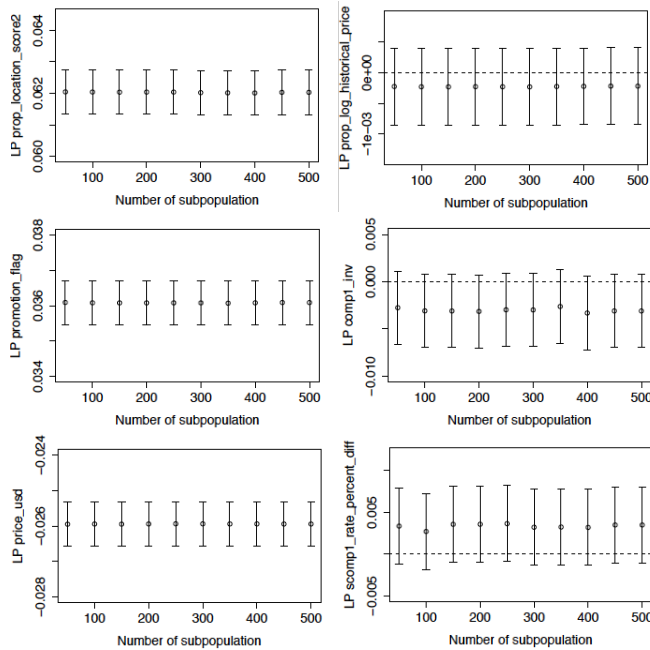
Fig. 9: LP statistics and 95% confidence intervals for six variables across different numbers of subpopulations (dotted line is at zero).

problems: (1) scalability for large data by exploiting distributed computing architectures using a confidence distribution based meta-analysis framework, and (2) automation for mixed data using a united LP computing formula. Undoubtedly, our theory can be adapted for other common data mining problems, and we are currently investigating how the proposed framework can be utilized to develop parallelizable regression and classification algorithms for big data.

Instead of developing distributed versions of statistical algorithms on a case-by-case basis, here we develop a generic platform to extend traditional and modern statistical modeling tools to large datasets using scalable, distributed algorithms. We believe this research is a great stepping stone towards developing a United Statistical Algorithm [25] to bridge the increasing gap between the theory and practice of small and big data analysis.

ACKNOWLEDGMENTS

REFERENCES

[1] Kaggle Inc., "Personalize Expedia hotel searches - ICDM 2013," 2013. [Online]. Available: https://www.kaggle.com/c/expedia-personalized-sort/data

[2] E. Parzen and S. Mukhopadhyay, "LP mixed data science: Outline of theory," *arXiv*, 2013. [Online]. Available: https://arxiv.org/pdf/1311.0562v2.pdf

[3] S. Mukhopadhyay and E. Parzen, "LP approach to statistical modeling," *arXiv*, 2014. [Online]. Available: https://arxiv.org/abs/1405.2601

[4] S. Mukhopadhyay, "Large-scale mode identification and data-driven sciences," *Electronic Journal of Statistics*, vol. 11, no. 1, pp. 215–240, 2017.

[5] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proceedings of the 6th OSDI*, December 2004, pp. 137–150. [Online]. Available: https://www.usenix.org/legacy/events/osdi04/tech/full_papers/dean/dean.pdf

[7] S. Guha, R. Hafen, J. Rounds, J. Xia, J. Li, B. Xi, and W. S. Cleveland, "Large complex data: divide and recombine (D&R) with RHIPE," *Stat*, vol. 1, no. 1, pp. 53–67, 2012.

[8] W. S. Cleveland and R. Hafen, "Divide and recombine (D&R): Data science for large complex data," *Statistical Analysis and Data Mining*, vol. 7, no. 6, pp. 425–433, 2014.

[9] A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan, "A scalable bootstrap for massive data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 4, pp. 795–816, 2014.

[10] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, pp. 3321–3363, 2013. [Online]. Available: http://jmlr.org/papers/v14/zhang13b.html

[11] J. Han and Q. Liu, "Bootstrap model aggregation for distributed statistical learning," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 1795–1803. [Online]. Available: http://papers.nips.cc/paper/6049-bootstrap-model-aggregation-for-distributed-statistical-learning.pdf

[12] N. Lin and R. Xi, "Aggregated estimating equation estimation," *Statistics and Its Interface*, vol. 4, pp. 73–83, 2011.

[13] X. Chen and M. Xie, "A split-and-conquer approach for analysis of extraordinarily large data," *Statistica Sinica*, vol. 24, no. 4, pp. 1655–1684, 2014.

[14] J. Higgins and S. G. Thompson, "Quantifying heterogeneity in a meta-analysis," *Statistics in Medicine*, vol. 21, no. 11, pp. 1539–1558, 2002.

[15] L. V. Hedges and I. Olkin, *Statistical methods for meta-analysis*. Orlando: Academic Press, 1985.

[16] A. J. Sutton and J. P. T. Higgins, "Recent developments in meta-analysis," *Statistics in Medicine*, vol. 27, no. 5, pp. 625–650, 2008.

[17] T. Schweder and N. L. Hjort, "Confidence and likelihood," *Scandinavian Journal of Statistics*, vol. 29, no. 2, pp. 309–332, 2002.

[18] K. Singh, M. Xie, and W. E. Strawderman, "Combining information from independent sources through confidence distributions," *The Annals of Statistics*, vol. 33, no. 1, pp. 159–183, 02 2005.

[19] M. Xie and K. Singh, "Confidence distribution, the frequentist distribution estimator of a parameter: A review," *International Statistical Review*, vol. 81, no. 1, pp. 3–39, 2013.

[20] E. Parzen, "Discussion of Confidence distribution, the frequentist distribution estimator of a parameter: A review," *International Statistical Review*, vol. 81, no. 1, pp. 48–52, 2013.

[21] M. Xie, K. Singh, and W. E. Strawderman, "Confidence distributions and a unifying framework for meta-analysis," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 320–333, 2011.

[22] R. DerSimonian and N. Laird, "Meta-analysis in clinical trials," *Controlled Clinical Trials*, vol. 7, no. 3, pp. 177 – 188, 1986.

[23] B. Efron, "Discussion of Confidence distribution, the frequentist distribution estimator of a parameter: A review," *International Statistical Review*, vol. 81, no. 1, pp. 41–42, 2013.

[24] Kaggle Inc., "Winners' presentation, Personalize Expedia hotel searches competition," 2013. [Online]. Available: https://www.dropbox.com/sh/5kedakjizgrog0y/_LE_DFCA7J/ICDM_2013

[25] E. Parzen and S. Mukhopadhyay, "United statistical algorithm, small and big data: Future of statisticians," *arXiv*, 2013. [Online]. Available: https://arxiv.org/pdf/1308.0641.pdf

Scott Bruce received his B.S. in Industrial and Systems Engineering and M.S. in Statistics from the Georgia Institute of Technology, USA in 2009 and 2010. He is currently a Ph.D. candidate in Statistics at Temple University, USA. His research interests include nonstationary time series analysis, big data learning, Bayesian data analysis, and nonparametric statistics with applications in medicine, public health, finance, and sports.

Zeda Li received his B.S. in Electrical Engineering from Central South University of Forestry and Technology, China in 2010 and M.S. in Biostatistics from Middle Tennessee State University, USA in 2013. He is currently a Ph.D. candidates in Statistics at Temple University, USA. His main research interests are time series analysis, big data analytics, high–dimensional statistics, sufficient dimension reduction methods, and functional data analysis.

Hsiang-Chieh Yang received his B.A. in Public Finance at National Chengchi University, Taiwan and M.S. in Statistics at Temple University, USA. He is currently a Ph.D. student in Accounting at the University of British Columbia, Canada. His major research interests include the relationship between financial reporting transparency and accounting standards, internal controls in financial institutions, and applications of machine learning in accounting research.

Subhadeep Mukhopadhyay received his Ph.D. degree in Statistics from Texas A&M University, USA in 2013. He is currently an Assistant professor of Statistical Science at Fox Business School, Temple University. He is also a member of the Data Science Institute of Temple University. His major research interest lies in developing theory of "United Statistical Algorithms" that could reveal the interconnectedness among different branches of statistics. He is a member of the IEEE.

The Supplementary Appendix

This supplementary document contains five Appendices. Section A provides a data dictionary for representative variables from various categories found in the Expedia dataset. Section B provides a small data example to demonstrate the applicability of the `MetaLP` framework on datasets both big and small. Section C demonstrates how the proper treatment of heterogeneity through the `MetaLP` approach provides new insights and resolutions for two challenging problems: Simpson's paradox and Stein's paradox. Finally, Sections D and E will describe a `MapReduce` computational implementation of the MetaLP inference engine and the τ^2 estimators used in our calculations.

APPENDIX A
EXPEDIA DATA DICTIONARY

See Table 1 for detailed descriptions of representative variables from each category of data found in the Expedia dataset. Data type information is also included to better illustrate the challenges stemming from the mixed data problem.

APPENDIX B
METALP ANALYSIS OF TITANIC DATA

The *Titanic* dataset is utilized as a benchmark to validate the effectiveness, accuracy, and robustness of the `MetaLP` analytical framework. Due to its manageable size, we are able to compute the full data LP estimates and can compare with the `MetaLP` estimates, which operate under a distributed

computing framework. A step-by-step `MetaLP` analysis of Titanic dataset is provided here.

The *Titanic* dataset contains information on 891 of its passengers, including which passengers survived. A key objective in analyzing this dataset is to better understand which factors (e.g. age, gender, class, etc.) significantly influence passenger survival. Complete descriptions of all eight variables can be found in Table 2. We seek to estimate the relationship between various passenger characteristics ($X_i, i = 1, \ldots, 7$) and the binary response variable (Y), passenger survival, by using both our distributed algorithm and traditional aggregated LP statistics to compare their results.

To develop an automatic solution to the mixed data problem, we start by constructing LP score polynomials for each variable. Figure 1 shows the shapes of LP basis functions for two variables from the *Titanic* data. Next, we randomly assign 891 observations to 5 different subpopulations and calculate LP statistics for each variable in each subpopulation, and then combine LP statistics to get a combined LP statistic for each variable. We repeat this process three times to see how much our final `MetaLP` result changes with different random partitions of the full data. Figures 2(a) shows the I^2 statistics for three random partitions on the *Titanic* dataset. Even with the randomly assigned partitions, some variables may exhibit heterogeneity among subpopulations as I^2 statistics move above 40%. For example, random partition 2 results show heterogeneity in variables Embarked and Sex. Thus, we use τ^2 regularization to handle the problem. Figure 2(b) shows the I^2 statistics after τ^2 regularization. The additional τ^2 parameter accounts for the heterogeneity in the subpopulations and adjusts the

Category	Variable	Data Type	Description
User information	`visitor_location_country_id`	Discrete	The ID of the country in which customer is located
	`visitor_hist_starrating`	Continuous	The mean star rating of hotels customer previously purchased
	`visitor_hist_adr_usd`	Continuous	The mean price of hotels customer previously purchased
	`orig_destination_distance`	Continuous	Physical distance between hotel and the customer
Search criteria	`srch_length_of_stay`	Discrete	Number of nights stay searched
	`srch_booking_window`	Discrete	Number of days in the future the hotel stay started
	`srch_adults_count`	Discrete	Number of adults specified in the hotel room
	`srch_children_count`	Discrete	Number of children specified in the hotel room
	`srch_room_count`	Discrete	Number of hotel rooms specified in the search
	`srch_saturday_night_bool`	Binary	Short stay including Saturday night
Static hotel characteristics	`prop_country_id`	Discrete	Country ID where customer is located
	`prop_starrating`	Discrete	Hotel star rating
	`prop_review_score`	Continuous	Mean hotel customer review score
	`prop_location_score1`	Continuous	Desirability of hotel location (1)
	`prop_location_score2`	Continuous	Desirability of hotel location (2)
	`prop_log_historical_price`	Continuous	Mean hotel price over last trading period
	`pprop_brand_bool`	Discrete	Independent or belongs to a hotel chain
Dynamic hotel characteristics	`price_usd`	Continuous	Displayed hotel price for the given search
	`promotion_flag`	Discrete	Hotel sale price promotion available
	`gross_booking_usd`	Continuous	Total transaction value
Competitor information	`comp1_rate_percent_dif`	Continuous	Absolute percentage difference between competitors
	`comp1_inv`	Binary	If competitor 1 has hotel availability
	`comp1_rate`	Discrete	If Expedia has lower/same/higher price than competitor
Other information	`srch_id`	Discrete	Search ID
	`site_id`	Discrete	Expedia Point of Sale ID
Response	`booking_bool`	Binary	Hotel booked or not

TABLE 1: Data dictionary for Expedia dataset.

Variable Name	Type	Description	Values
Survival	Binary	Survival	0 = No; 1 = Yes
Pclass	Categorical	Passenger Class	1 = 1st; 2 = 2nd; 3 = 3rd
Sex	Binary	Sex	Male; Female
Age	Continuous	Age	0 - 80
Sibsp	Discrete	Number of Siblings/Spouses Aboard	0 - 8
Parch	Discrete	Number of Parents/Children Aboard	0 - 6
Fare	Continuous	Passenger Fare	0 - 512.3292
Embarked	Categorical	Port of Embarkation	C = Cherbourg; Q = Queenstown; S = Southampton

TABLE 2: Data dictionary for the *Titanic* dataset.

Fig. 1: (a) Top: first four LP orthonormal score functions for variable `# Siblings/Spouses Aboard`, a discrete random variable taking values $0, \ldots, 8$; (b) Bottom: first four LP orthonormal score functions for continuous variable `Passenger Fare`.

estimators accordingly, resulting in significantly lower I^2 statistics for all variables under this model.
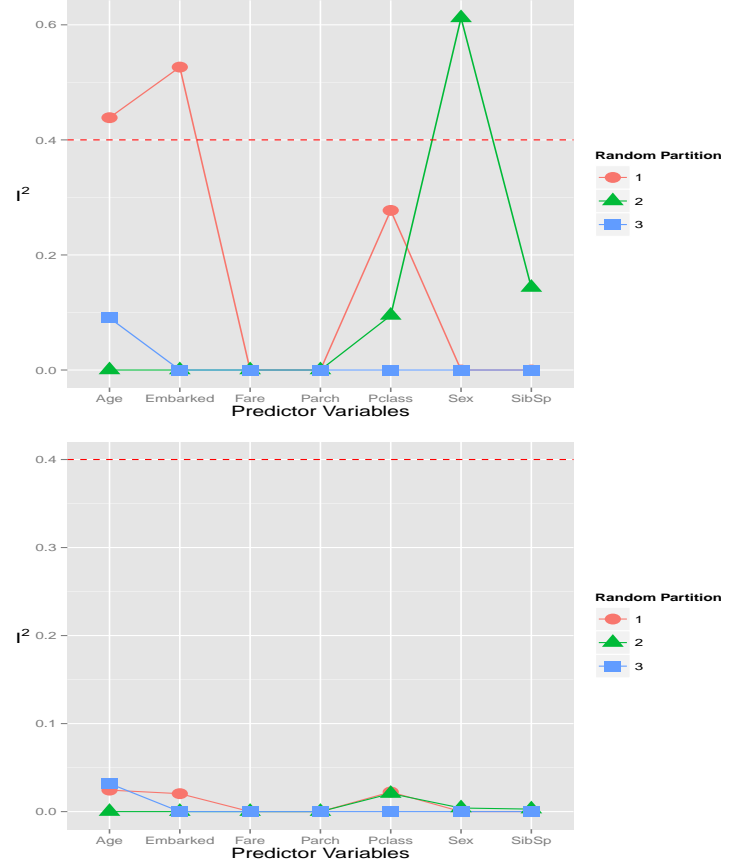

Fig. 2: (a) Top: I^2 diagnostics for three random partitions of the *Titanic* dataset (b) Bottom: I^2 diagnostic with τ^2 regularization on the *Titanic* dataset for three random partitions.

Figure 3 contains the LP statistics and their 95% confidence intervals generated from our algorithm for 3 repetitions of random groupings ($k = 5$) along with the confidence intervals generated using the whole dataset. A *remarkable result of our method* is that the `MetaLP` estimators and the aggregated (full data) LP estimators are almost indistinguishable for *all* variables. In summary, the estimators from our `MetaLP` method produces very similar inference to the estimators using the entire dataset, which means we can obtain accurate and robust statistical inference while taking advantage of the computational efficiency in parallel, distributed processing.
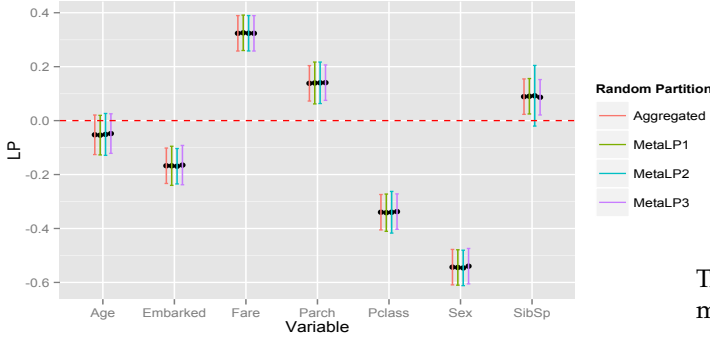
Fig. 3: 95% Confidence Interval of LP Statistic for each variable based on three `MetaLP` repetitions and aggregated full dataset (which is the oracle estimate).

Dept	Male	Female
A	62% (512 / 825)	**82%** (89 / 108)
B	63% (353 / 560)	**68%** (17 / 25)
C	**37%** (120 / 325)	34% (202 / 593)
D	33% (138 / 417)	**35%** (131 / 375)
E	**28%** (53 / 191)	24% (94 / 393)
F	**6%** (22 / 373)	7% (24 / 341)
All	**45%** (1198 / 2691)	30% (557 / 1835)

TABLE 3: UC Berkeley admission rates by gender by department.

APPENDIX C
SIMPSON'S AND STEIN'S PARADOX: A METALP PERSPECTIVE

Heterogeneity is not solely a big data phenomenon; it can easily arise in the small data setup. We show two examples, Simpson's Paradox and Stein's Paradox, where blind aggregation *without paying attention to the underlying heterogeneity* leads to a misleading conclusion.

C.1 Simpson's Paradox

Table 3 shows the UC Berkeley admission data [1] by department and gender. Looking only at the university level admission rates at the bottom of this table, there appears to be a significant difference in admission rates for males at 45% and females at 30%. However, the department level data *does not* appear to support a strong gender bias as in the university level data. The real question at hand is whether *there is a gender bias in university admissions?* We provide a concrete statistical solution to the question put forward by [2] regarding the validity and applicability of traditional statistical tools in answering the real puzzle of Simpson's Paradox: "So in what sense do B-K plots, or ellipsoids, or vectors display, or regressions etc. contribute to the puzzle? They don't. They can't. Why bring them up? Would anyone address the real puzzle? It is a puzzle that cannot be resolved in the language of traditional statistics."

In particular, we will demonstrate how adopting the `MetaLP` modeling and combining strategy (that properly takes the existing heterogeneity into account) can resolve issues pertaining to Simpson's paradox [3]. This simple example teaches us that *simply averaging* as a means of combining effect sizes is *not appropriate* regardless of the size of the data. The calculation for the weights *must* take into account the underlying departure from homogeneity, which is ensured in the `MetaLP` distributed inference framework. Now we explain how this paradoxical reversal can be resolved using the `MetaLP` approach.

As both admission (Y) and gender (X) are binary variables, we can compute at most one LP orthogonal polynomial for each variable $T_1(Y; Y)$ and $T_1(X; X)$; accordingly, we can compute only the first-order linear LP statistics, $\text{LP}[1; Y, X]$, for each department. Following Equation (9),

we derive and estimate the aCD for the LP statistic for each of the 6 departments, $H(\text{LP}_l[1; X, Y])$, $l = 1, \ldots, 6$, and for the aggregated university level dataset, $H(\text{LP}_a[1; X, Y])$. As noted in Section 3.3, the department level aCDs are normally distributed with a mean of $\widehat{\text{LP}}_l[1; X, Y]$ and variance of $1/n_\ell$ where n_ℓ is the number of applicants to department ℓ. Similarly, the aggregated aCD is also normally distributed with a mean of $\widehat{\text{LP}}_a[1; X, Y]$ and variance of $1/n_a$ where n_a is the number of applicants across all departments.

Now we apply the heterogeneity-corrected `MetaLP` algorithm following Theorem 3.5 to estimate the combined aCD across all departments as follows:

$$H^{(c)}(\text{LP}[1; X, Y]) =$$

$$\Phi\left[\left(\sum_{\ell=1}^{6}\frac{1}{\tau^2 + (1/n_\ell)}\right)^{1/2}(\text{LP}[1; X, Y] - \widehat{\text{LP}}^{(c)}[1; X, Y])\right]$$

with

$$\widehat{\text{LP}}^{(c)}[1; X, Y]) = \frac{\sum_{\ell=1}^{6}(\tau^2 + (1/n_\ell))^{-1}\widehat{\text{LP}}_\ell[1; X, Y]}{\sum_{\ell=1}^{6}(\tau^2 + (1/n_\ell))^{-1}}$$

where $\widehat{\text{LP}}^{(c)}[1; X, Y])$ and $\sum_{l=1}^{6}(\tau^2 + (1/n_\ell))^{-1}$ are the mean and variance respectively of the meta-combined aCD for $\text{LP}[1; X, Y]$. Here, the heterogeneity parameter, τ^2, is estimated using the restricted maximum likelihood formulation outlined in Supplementary Section E. Figure 4(a) displays the estimated aCDs for each department, aggregated data, and for the `MetaLP` method. First note that the aggregated data aCD is very different from the department level aCDs, which is characteristic of the Simpson's paradox reversal phenomenon due to naive "aggregation bias". This is why the aggregated data inference suggests a gender bias in admissions, while the department level data does not. Second, note that the aCD from the `MetaLP` method provides an estimate that falls more in line with the department level aCDs. This highlights the advantage of the `MetaLP` meta-analysis framework for combining information in a judicious manner. Also, as mentioned in Section 3.3, all traditional forms of statistical inference (e.g. point and interval estimation, hypothesis testing) can be derived from the aCD above.

For example, we can test $H_0 : \text{LP}^{(c)}[1; X, Y] \leq 0$ (indicating no male preference in admissions) vs. $H_1 : \text{LP}^{(c)}[1; X, Y] > 0$ (indicating a male preference in admissions) using the aCD for $\text{LP}^{(c)}[1; X, Y]$. The corresponding
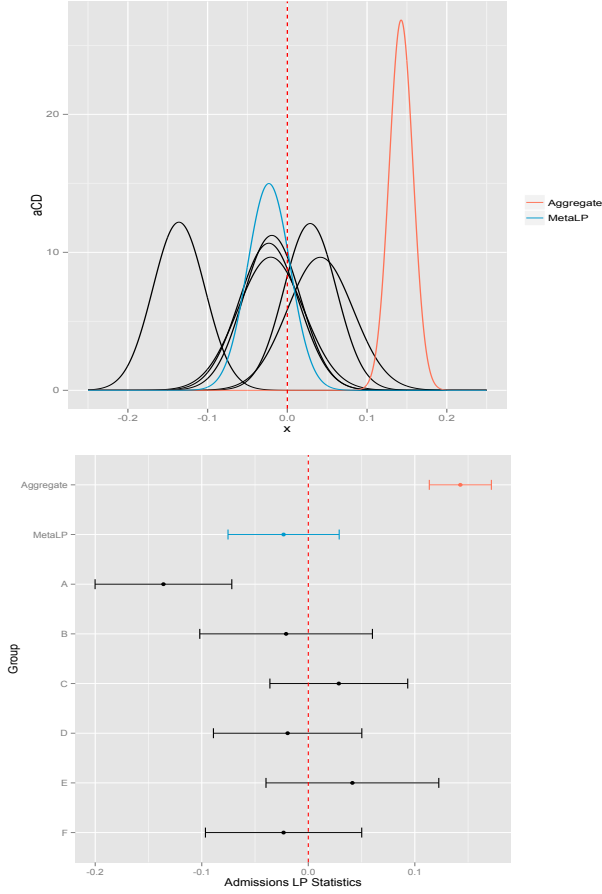
Fig. 4: (a) aCDs (top) and (b) 95% confidence intervals (bottom) for linear LP statistics for UC Berkeley admission rates by gender (department level aCDs and confidence intervals in black).

p-value for the test comes from the probability associated with the support of H_0, $C = (-\infty, 0]$, (i.e. "high" support value for H_0 leads to acceptance) following [4]. Hence, the p-value for the above test is

$$\text{p-value} = H\left(0; \text{LP}^{(c)}[1; X, Y]\right)$$
$$= \Phi\left(\frac{0 - \widehat{\text{LP}}^{(c)}[1; X, Y]}{\sqrt{\sum_{l=1}^{6}(\tau^2 + (1/n_l))^{-1}}}\right) \approx .81.$$

In this case, the support of the LP CD inference (also known as 'belief' in fiducial literature [5]) is .81. Hence, at the 5% level of significance, we fail to reject H_0 and confirm that there is no evidence to support a significant gender bias favoring males in admissions using the MetaLP approach.

In addition, we can also compute the 95% confidence intervals for the LP statistics measuring the significance of the relationship between gender and admissions as shown in Figure 4(b). Note the paradoxical reversal as 5 out of the 6 departments show no significant gender bias at the 5% level of significance (confidence intervals include positive and negative values), while the confidence interval for the aggregated dataset indicates a significantly higher admission rate for males. On the other hand, note that the MetaLP

approach resolves the paradox (which arises *due to the failure of recognizing* the presence of heterogeneity among department admission patterns) and correctly concludes that no significant gender bias exists (as the confidence interval for the MetaLP-based LP statistic includes the null value 0).

C.2 Stein's Paradox

Perhaps the most popular and classical dataset for Stein's paradox is given in Table 4, which shows the batting averages of 18 major league players through their first 45 official at-bats of the 1970 season. The goal is to predict each player's batting average over the remainder of the season (comprising about 370 more at bats each) using only the data of the first 45 at-bats. Stein's shrinkage estimator [6], which can be interpreted as an empirical Bayes estimator [7] turns out to be more than 3 times more efficient than the MLE estimator. Here we provide a MetaLP approach to this problem by recognizing the "parallel" structure (18 parallel sub-populations) of baseball data, which fits nicely into the "decentralized" MetaLP modeling framework.

Name	hits/AB	$\hat{\mu}_i^{(\text{MLE})}$	μ_i	$\hat{\mu}_i^{(JS)}$	$\hat{\mu}_i^{(\text{LP})}$
Clemente	18/45	.400	.346	**.294**	.276
F Robinson	17/45	.378	.298	**.289**	.274
F Howard	16/45	.356	.276	.285	**.272**
Johnstone	15/45	.333	.222	.280	**.270**
Berry	14/45	.311	.273	**.275**	.268
Spencer	14/45	.311	.270	.275	**.268**
Kessinger	13/45	.289	.263	.270	**.265**
L Alvarado	12/45	.267	.210	.266	**.263**
Santo	11/45	.244	.269	**.261**	.261
Swoboda	11/45	.244	.230	**.261**	.261
Unser	10/45	.222	.264	.256	**.258**
Williams	10/45	.222	.256	**.256**	.258
Scott	10/45	.222	.303	.256	**.258**
Petrocelli	10/45	.222	.264	.256	**.258**
E Rodriguez	10/45	.222	.226	**.256**	.258
Campaneris	9/45	.200	.286	.252	**.256**
Munson	8/45	.178	.316	.247	**.253**
Alvis	7/45	.156	.200	**.242**	.251

TABLE 4: Batting averages $\hat{\mu}_i^{(\text{MLE})}$ for 18 major league players early in the 1970 season; μ_i values are averages over the remainder of the season. The James-Stein estimates $\hat{\mu}_i^{(JS)}$ and MetaLP estimates $\hat{\mu}_i^{(\text{LP})}$ provide much more accurate overall predictions for the μ_i values compared to MLE. MSE ratio for $\hat{\mu}_i^{(JS)}$ to $\hat{\mu}_i^{(\text{MLE})}$ is 0.283 and MSE ratio for $\hat{\mu}_i^{(\text{LP})}$ to $\hat{\mu}_i^{(\text{MLE})}$ is 0.293 showing comparable efficiency.

We start by defining the variance-stabilized effect-size estimates for each group

$$\widehat{\theta}_i = \sin^{-1}(2\hat{\mu}_i^{(\text{MLE})} - 1), \quad i = 1, \ldots, k$$

whose asymptotic distribution is normal with mean θ_i and variance $1/n_i$ where $n_i = 45$ (for all i) is the number of at-bats for each player and $\hat{\mu}_i^{(\text{MLE})}$ is the individual batting average for player i. Figure 5 provides some visual evidence of the heterogeneity between the studies.

We apply a MetaLP procedure that incorporates inter-study variations and is applicable for unequal variance/sample size scenarios with no further adjustment.

First, we estimate the weighted mean, $\hat{\theta}_\mu$, of the transformed batting averages with weights for each study $(\hat{\tau}_{DL}^2 + n_i^{-1})^{-1}$, where $\hat{\tau}_{DL}^2$ denotes the DerSimonian and Laird data-driven estimate given in Supplementary Section E. The `MetaLP` estimators, $\hat{\theta}_i^{(LP)}$, are represented as weighted averages between the transformed batting averages and $\hat{\theta}_\mu$ as follows:

$$\hat{\theta}_i^{(LP)} = \lambda \hat{\theta}_\mu + (1 - \lambda)\widehat{\theta}_i, \quad (i = 1, \ldots, 18),$$

where $\lambda = (n_i^{-1})/(\hat{\tau}_{DL}^2 + n_i^{-1})$. Table 4 shows that `MetaLP`-based estimators are as good as James-Stein empirical Bayes estimators for the baseball data. This stems from the simple fact that random-effect meta-analysis and the Stein formulation are mathematically equivalent. But nevertheless, the framework of understanding and interpretations are different. Additionally, `MetaLP` is much more flexible and automatic in the sense that it works for 'any' estimators (such as mean, regression function, classification probability) beyond mean and Gaussianity assumptions. We feel the `MetaLP` viewpoint is also less mysterious and clearly highlights the core issue of heterogeneity. Our analysis indicates an exciting frontier of future research at the interface of `MetaLP`, Empirical Bayes, and Stein's Paradox to develop new theory of distributed massive data modeling.
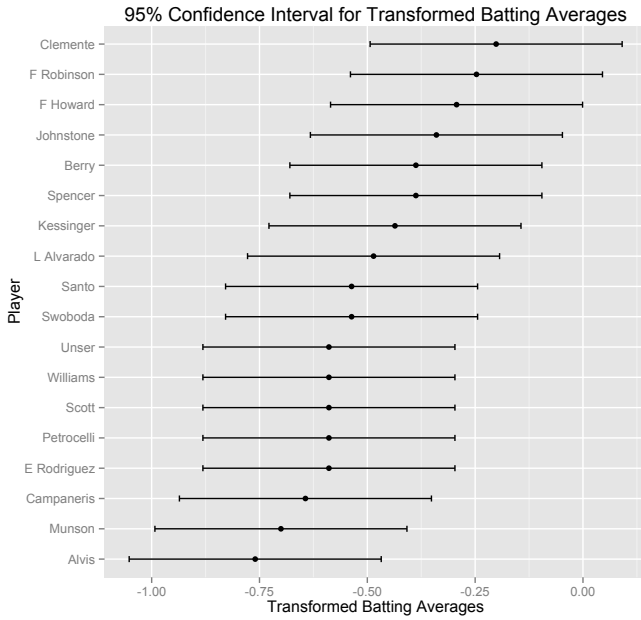

Fig. 5: 95% confidence intervals for transformed batting averages, θ_i, for each player, indicating the heterogeneity of the effect sizes estimates.

APPENDIX D
MAPREDUCE COMPUTATION FRAMEWORK AND R FUNCTIONS

In this note, we describe how the proposed `MetaLP` statistical algorithmic framework for big data analysis can easily be integrated with the `MapReduce` computational framework, along with the required R code. MapReduce implementation of `MetaLP` allows efficient parallel processing of large amounts of data to achieve scalability.

D.1 LP.Mapper

We apply the following `LP.Mapper` function to each subpopulation. This function computes $LP[j; X, Y]$ for $j = 1, \ldots, m$ (where user selects m, which should be less than the number of distinct values of the given random sample). The first step is to design the data-adaptive orthogonal LP polynomial transformation of the given random variable X. This is implemented using the function `LP.Score.fun`. The second step uses the LP inner product to calculate the LP variable selection statistic using the function `LP.VarStat` (see Section 3.1 for details).

Inputs of `LP.Mapper`. Y is binary (or discrete multinomial) and X is a mixed (discrete or continuous type) predictor variable.

Outputs of `LP.Mapper`. It returns the estimated $\widehat{LP}[j; X, Y]$ and the corresponding (asymptotic) sample variance. Note that the sample LP statistic converges to $\mathcal{N}(0, \sigma_\ell^2 = 1/n_\ell)$, where n_ℓ is the effective sample size of the ℓth subpopulation. By effective size we mean $n_\ell - M_\ell(X)$, where $M_\ell(X)$ denotes the number of missing observations for variable X in the ℓth partition. `LP.Mapper` returns only $\{\widehat{LP}[1; X, Y], \ldots, \widehat{LP}[m; X, Y]\}$ and n_ℓ, from which we can easily reconstruct the CD of the LP statistics.

```
LP.Mapper <- function (Y,x,m=1) {
LP.Score.fun <- function(x,m){
    u <- (rank(x,ties.method = c("average"))-.5)/length(x);
    m <- min(length(unique(u ))-1, m);
    S.mat <- as.matrix(poly(u,df=m));
    return(as.matrix(scale(S.mat)))
  }
  LP.VarStat <- function(Y,x,m){
    x <- ifelse (x=="NULL",NA,x);
    x <- na.omit (x);
    if (length (unique(x)) <=1 ){
      r.lp=0;
            n=0;
    }else{
      which <- na.action(x);
      if (length(which)>0) Y <- Y[-which];
      if (length(unique(Y))<=1){
        r.lp=0;
                n=0;
      }else{
        x <- as.numeric (x);
        S <- LP.Score.fun(x,m);
        r.lp <- cor(Y,S);n=length(Y);
      }
    }
    return(c(r.lp,n))
}
    temp <- LP.VarStat(Y,x,m);
    output.LP <- temp[1:length(temp)-1];
    output.n <- temp [length(temp)]
    logic <- ifelse (length(temp)-1==m,"NA",
    "m is not less than the number of distinct value of x")
return (list(LP=output.LP,n=output.n,Warning=logic))
  }
```

D.2 Meta.Reducer

`LP.Mapper` computes the sample LP statistics and the corresponding sample variance. Now at the 'Reduce' step, our goal is to judiciously combine these estimates from k subpopulations to produce the statistical inference for the original large data. Here we implement the MetaReduce strategy to combine the inference from all the subpopulations, implemented in the function `Meta.Reducer`.

Before performing the `Meta.Reducer` step, we run the 'combiner' operation that gathers the outputs of the

`LP.Mapper` function for all the subpopulations and organizes them in the form of a list, which has two components: (i) a matrix `L.value` of order $k \times p$, where k is the number of subpopulations and p is the number of predictor variables (the (ℓ, i)th element of that matrix stores the jth LP statistic $LP[j; X_i, Y]$ for ℓth partition); (ii) a matrix `P.size` of size $k \times p$ ((ℓ, i)th element stores the effective size of the subpopulation for the variable ℓ).

Inputs of `Meta.Reducer`

1) `L.value` and `P.size`
2) `fix`: a binary argument (TRUE or FALSE), indicating whether to ignore the τ^2 regularization. If it equals to FALSE, then the model with τ^2 regularization is applied.
3) `method`: It's valid only if `fix` equals FALSE, and can equal to either `"DL"` or `"REML"`, indicating the estimation method of τ^2.
4) `"DL"` stands for the method proposed in [8], and `"REML"` is the restricted maximum likelihood method, which was proposed in [9]. We include the calculation methods of these two τ^2 estimators in the next section.

Outputs of `Meta.Reducer`

1) Meta-analysis combined LP statistic estimators
2) Standard errors of meta-analysis combined LP statistic estimators
3) I^2 heterogeneity diagnostic
4) τ^2 estimate only if `fix` equals to FALSE

```
Meta.Reducer <- function(L.value, P.size, fix, method){
th_c <- NA;
  sd_th_c <- NA;
  for (i in 1:ncol(L.value)){
    th_c[i] <- sum(L.value[,i]*P.size[,i])/sum(P.size[,i]);
    sd_th_c[i] <- sqrt(1/sum(P.size[,i]));
  }
  Q <- matrix (,ncol(L.value),1);
  for (i in 1:ncol(L.value)){
    Q[i,] <- sum ( P.size[,i]*(L.value [,i] - th_c[i])^2);
  }
  K<-NA;
  for (i in 1:ncol(L.value)){
    A <- P.size[,i];
    K[i] <- length (A[A!=0]);
  }
  if (fix==T){
    I_sq.f <- ifelse ((Q-(K-1))/Q>0, (Q-(K-1))/Q,0);
    return (list(LP.c=th_c, SE.LP.c=sd_th_c,I_sq.f=I_sq.f))
  }else{
   if (method=="DL"){
    tau.sq <- NA;
    for (i in 1:ncol(L.value)){
     tau.sq[i] <- (Q[i]-(K[i]-1)) /
       (sum(P.size[,i])
       - sum((P.size[,i])^2)/sum(P.size[,i]));
    }
    tau.sq <- ifelse(tau.sq>0,tau.sq,0);
    w_i <- matrix(NA,nrow(P.size), ncol(P.size));
    for (i in 1:ncol(L.value)){
     w_i[,i] <- (1/P.size[,i]+tau.sq[i])^-1;
    }
    mu.hat <- NA;
    SE_mu.hat <- NA;
    for (i in 1:ncol(L.value)){
     mu.hat[i] <- sum(L.value[,i]*w_i[,i])/sum(w_i[,i]);
     SE_mu.hat[i] <- sqrt(1/sum(w_i[,i]));
    }
    lam_i <- matrix (NA,nrow(P.size),ncol(P.size));
    for (i in 1:ncol(L.value)){
     lam_i[,i] <- (1/P.size[,i])/(1/P.size[,i]+tau.sq[i]);
    }
    th.tilde <- matrix(NA,nrow(L.value), ncol(L.value))
    for (i in 1:ncol(L.value)){
     th.tilde[,i] <- lam_i[,i] * mu.hat [i] +
```

```
      (1-lam_i[,i])*L.value[,i];
    }
    th.tilde <- ifelse(is.nan(th.tilde)==T,0,th.tilde);
    Q <- matrix (NA,ncol(L.value),1);
    for (i in 1:ncol(L.value)){
     Q[i,] <- sum ( w_i[,i]*(th.tilde [,i] - mu.hat[i])^2);
    }
    I_sq.r <- ifelse ((Q-(K-1))/Q>0, (Q-(K-1))/Q,0);
    return (list (LP.c=mu.hat,
    SE.LP.c=SE_mu.hat,I_sq.r=I_sq.r,tau.sq=tau.sq))
  }
  if (method=="REML"){
   tau.sq <- NA;
   for (i in 1:ncol(L.value)){
    tau.sq[i] <- (Q[i]-(K[i]-1)) /
     (sum(P.size[,i]) -
     sum((P.size[,i])^2/sum(P.size[,i])))
   }
   tau.sq <- ifelse(tau.sq>0,tau.sq,0);
   for (i in 1:ncol(L.value)){
    if (sum(P.size[,i]==0)>0){
     n <- P.size[,i][-which(P.size[,i]==0)];
     thh <- L.value[,i][-which(P.size[,i]==0)];
    }else{
     n <- P.size[,i];
     thh <- L.value[,i];
    }
    nloop <- 0;
    absch <- 1;
    while (absch > 10^(-10)){
     nloop <- nloop + 1;
     if (nloop > 10^5){
      tau.sq[i] <- NA ;
     }
     else{
      tau.sq.old <- tau.sq[i]
      # update thetaR, wR
      wR <- 1/(1/n + tau.sq.old);
      thetaR <- sum(wR*thh) / sum(wR);
      # update tauR
      tau.sq[i] <- sum(wR^2*(K[i]/(K[i]-1)*
      (thh- thetaR)^2 - 1/n) ) / sum(wR^2);
      absch <- abs(tau.sq[i] - tau.sq.old);
     }
    }
   }
   tau.sq <- ifelse(tau.sq>0, tau.sq, 0);
   w_i <- matrix(NA,nrow(P.size),ncol(P.size));
   for (i in 1:ncol(L.value)){
    w_i[,i] <- (1/P.size[,i]+tau.sq[i])^-1;
   }
   mu.hat <- NA;
   SE_mu.hat <- NA;
   for (i in 1:ncol(L.value)){
    mu.hat[i] <- sum(L.value[,i]*w_i[,i])/sum(w_i[,i]);
    SE_mu.hat[i] <- sqrt(1/sum(w_i[,i]));
   }
   lam_i <- matrix(NA,nrow(P.size), ncol(P.size));
   for (i in 1:ncol(L.value)){
    lam_i[,i] <- (1/P.size[,i])/(1/P.size[,i]+tau.sq[i]);
   }
   th.tilde <- matrix (NA,nrow(L.value),ncol(L.value));
   for (i in 1:ncol(L.value)){
    th.tilde [,i] <- lam_i[,i] * mu.hat [i] +
    (1-lam_i[,i])*L.value[,i];
   }
   th.tilde <- ifelse(is.nan(th.tilde)==T,0,th.tilde);
   Q <- matrix(NA,ncol(L.value),1);
   for (i in 1:ncol(L.value)){
    Q[i,] <- sum(w_i[,i]*(th.tilde [,i] - mu.hat[i])^2);
   }
   I_sq.r <- ifelse ((Q-(K-1))/Q>0,(Q-(K-1))/Q,0);
   return(list(LP.c=mu.hat,SE.LP.c=SE_mu.hat,
   I_sq.r=I_sq.r,tau.sq=tau.sq))
   }
  }
}
```

APPENDIX E
τ^2 ESTIMATOR

There are many different proposed estimators for the τ^2 parameter. We consider the DerSimonion and Laird estimator [8], $\hat{\tau}^2_{\mathrm{DL}}$, and the restricted maximum likelihood estimator

[9], $\hat{\tau}^2_{\text{REML}}$, for our analysis. $\hat{\tau}^2_{\text{DL}}$ can be found from the following equation:

$$\hat{\tau}^2_{DL} = \max\left\{0, \frac{Q-(k-1)}{\sum_\ell s_\ell^{-2} - \sum_\ell s_\ell^{-4}/\sum_\ell s_\ell^{-2}}\right\};$$

where

$$Q = \sum_{\ell=1}^k \left(\widehat{\text{LP}}_\ell[j;X,Y] - \widehat{\text{LP}}^{(c)}[j;X,Y]\right)^2 s_\ell^{-2}.$$

However, $\hat{\tau}^2_{\text{REML}}$ should be calculated in an iterative fashion to maximize the restricted likelihood following these steps:

Step 1: Obtain the initial value, $\hat{\tau}^2_0$. We use $\hat{\tau}^2_{\text{DL}}$ as the initial value:

$$\hat{\tau}^2_0 = \hat{\tau}^2_{\text{DL}}.$$

Step 2: Obtain $\widehat{\text{LP}}^{(c)}_\tau[j;X,Y]$ (τ-corrected combined LP statistics).

$$\widehat{\text{LP}}^{(c)}_\tau[j;X,Y] = \frac{\sum_\ell w_\ell(\tau_0^2)\widehat{\text{LP}}_\ell[j;X,Y]}{\sum_\ell w_\ell(\hat{\tau}_0^2)};$$
$$w_\ell(\hat{\tau}_0^2) = (s_\ell^2 + \hat{\tau}_0^2)^{-1}.$$

Step 3: Obtain the REML estimate.

$$\hat{\tau}^2_{\text{REML}} =$$
$$\frac{\sum_\ell w_\ell^2(\hat{\tau}_0^2)\left(\frac{k}{k-1}\left(\widehat{\text{LP}}_\ell[j;X,Y] - \widehat{\text{LP}}^{(c)}_\tau[j;X,Y]\right) - s_\ell^2\right)}{\sum_\ell w_\ell^2(\hat{\tau}_0^2)}.$$

Step 4: Compute new $\widehat{\text{LP}}^{(c)}_\tau[j;X,Y]$ by plugging $\hat{\tau}^2_{\text{REML}}$ obtained in Step 3 into formula from Step 2.

Step 5: Repeat Step 2 and Step 3 until $\hat{\tau}^2_{\text{REML}}$ converges.

Convergence can be measured as the absolute difference between $\hat{\tau}^2_{\text{REML}}$ from the latest iteration and the previous iteration reaching a threshold close to zero.

REFERENCES

[1] P. J. Bickel, E. A. Hammel, and J. W. O'Connell, "Sex bias in graduate admissions: Data from berkeley," *Science*, vol. 187, no. 4175, pp. 398–404, 1975.

[2] J. Pearl, "Comment: Understanding simpson's paradox using a graph." 2014. [Online]. Available: http://andrewgelman.com/2014/04/08/understanding-simpsons-paradox-using-graph/

[3] E. H. Simpson, "The interpretation of interaction in contingency tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 13, no. 2, pp. 238–241, 1951.

[4] M. Xie and K. Singh, "Confidence distribution, the frequentist distribution estimator of a parameter: A review," *International Statistical Review*, vol. 81, no. 1, pp. 3–39, 2013.

[5] M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics*, 3rd ed. London: Griffin, 1974, vol. 2.

[6] W. James and C. Stein, "Estimation with quadratic loss," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Berkeley, Calif.: University of California Press, 1961, pp. 361–379.

[7] B. Efron and C. Morris, "Data analysis using stein's estimator and its generalizations," *Journal of the American Statistical Association*, vol. 70, no. 350, pp. 311–319, 1975.

[8] "Meta-analysis in clinical trials," *Controlled Clinical Trials*, vol. 7, no. 3, pp. 177 – 188, 1986.

[9] S.-L. T. Normand, "Meta-analysis: formulating, evaluating, combining, and reporting," *Statistics in Medicine*, vol. 18, no. 3, pp. 321–359, 1999.