Distributed Data Mining for Astronomy Catalogs

Chris Giannella* Haimonti Dutta† Kirk Borne‡ Ran Wolff§ Hillol Kargupta¶

Abstract

The design, implementation, and archiving of very large sky surveys is playing an increasingly important role in today's astronomy research. However, these data archives will necessarily be geographically distributed. To fully exploit the potential of this data, we believe that capabilities ought to be provided allowing users a more communication-efficient alternative to multiple archive data analysis than first down-loading the archives fully to a centralized site.

In this paper, we describe the architecture of a system, DEMAC, for the distributed mining of massive astronomical catalogs. The system is designed to sit on top of the existing national virtual observatory environment and provide tools for distributed data mining (as web services) without requiring datasets to be fully down-loaded to a centralized server. To illustrate the potential effectiveness of DEMAC, we carry out a case study using distributed principal component analysis (PCA) for detecting fundamental planes of astronomical parameters. In particular, PCA enables dimensionality reduction within a set of correlated physical parameters, such as a reduction of a 3-dimensional data distribution (in astronomer's observed units) to a planar data distribution (in fundamental physical units). Fundamental physical insights are thereby enabled through efficient access to distributed multi-dimensional data sets.
Keywords: distributed data mining, astronomy catalogs, principal component analysis, fundamental plane.

1 Introduction

The design, implementation, and archiving of very large sky surveys is playing an increasingly important role in today's astronomy research. Many projects today (*e.g.* GALEX All-Sky Survey), and many more projects in the near future (*e.g.* WISE All-Sky Survey) are destined to produce enormous catalogs (tables) of astronomical sources (tuples). These catalogs will necessarily be geographically distributed. It is this virtual collection of gigabyte, terabyte, and (eventually) petabyte catalogs that will significantly increase science return and enable remarkable new scientific discoveries through the integration and cross-correlation of data across these multiple survey dimensions [21]. Astronomers will be unable to fully tap the riches of this data without a new paradigm for *astro-informatics* that involves distributed database queries and data mining across distributed virtual tables of joined and integrated sky survey catalogs [4, 5].

The development and deployment of a National Virtual Observatory (NVO) [25] is a step toward a solution of this problem. However, processing, mining, and analyzing these distributed and vast data collections are fundamentally challenging tasks since most off-the-shelf data mining systems require the data to be down-loaded to a single location before further analysis. This imposes serious scalability constraints on the data mining system and fundamentally hinders the scientific discovery process. Figure 1 further illustrates this technical problem. The left part depicts the current data flow in the NVO. Through web services, data are selected and down-loaded from multiple sky-surveys.

If distributed data repositories are to be really accessible by a larger community, then technology ought to be developed for supporting distributed data analysis that can reduce, as much as possible, communication requirements among the data servers and the client machines. Communication-efficient distributed data mining (DDM) techniques will allow a large number of users simultaneously to perform advanced data analysis without necessarily down-loading large volumes of data to their respective client machines.

In this paper, we describe the architecture of a system, *DEMAC*, for the distributed exploration of massive astronomical catalogs. The primary purpose of DEMAC is to provide a collection of data mining tools based on various DDM algorithms. DEMAC is designed to reside on top of the existing NVO environment and provide tools for data mining (as

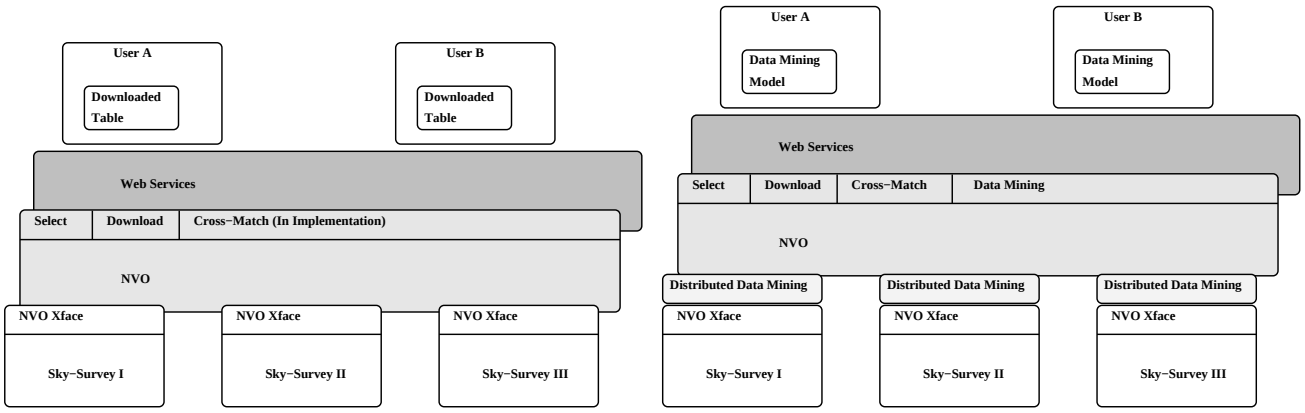
*CSEE Department, University of Maryland, Baltimore County, USA cgiannel@cs.umbc.edu

†CSEE Department, University of Maryland, Baltimore County, USA hdutta1@cs.umbc.edu

‡School of Computational Sciences, George Mason University, USA kborne@gmu.edu

§Computer Science Department, Technion, Israel ran.wolff@gmail.com

¶CSEE Department, University of Maryland Baltimore County, USA, hillol@cs.umbc.edu. Also affiliated with AGNIK LLC, Columbia, MD USA.



(a) Current data flow is restricted because of data ownership and bandwidth considerations.

(b) Distributed data mining algorithms can process large amounts of data using a small amount of communication. The users get the data mining output rather than raw data.

Figure 1: Data flow for distributed data mining embedded in the NVO.

web services) without requiring datasets to be down-loaded to a centralized server. Consider again Figure 1. DEMAC requires a relatively simple modification – the addition a distributed data mining functionality in the sky servers. This allows DDM to be carried out without having to down-load large tables to the users' desktop or some other remote machine. Instead, the users will only down-load the output of the data mining process (a data mining model); the actual data mining from multiple data servers will be performed using communication-efficient DDM algorithms. The algorithms we develop sacrifice perfect accuracy for communication savings. They offer approximate results at a considerably lower communication cost than that of exact results through centralization. As such, we see DEMAC as serving the role of an exploratory "browser". Users can quickly get (generally quite accurate) results for their distributed queries at low communication cost. Armed with these results, users can focus in on a specific query or portion of the datasets, and down-load for more intricate analysis.

At present, we are in the design phase of the project. We have not begun implementation of the full system. The purpose of this paper is to describe the basic arcitecture and to present a case study designed to illustrate the potential effectiveness of DEMAC. The case study uses one of the DDM algorithms that will be included in DEMAC: distributed principal component analysis (PCA). In a simulated environment, the study investigates the accuracy of the distributed PCA algorithm with respect to its communication savings (measured relative to the communication needed to centralized all the data).

PCA is important because it is used to detect fundamental planes of astronomical parameters. Astronomers have previously discovered cases where the observed parameters measured for a particular class of astronomical objects (such as elliptical galaxies) are strongly correlated, as a result of universal astrophysical processes (such as gravity). PCA will find such correlations in the form of principal components. An example of this is the reduction of a 3-dimensional scatter plot of elliptical galaxy parameters to a planar data distribution. The explanation of this plane follows from fundamental astrophysical processes within galaxies, and thus the resulting data distribution is labeled the Fundamental Plane of Elliptical Galaxies. The important physical insights that astronomers have derived from this fundamental plane suggest that similar new physical insights and scientific discoveries may come from new analysis of combinations of other astronomical parameters. Since these multiple parameters are now necessarily distributed across geographically dispersed data archives, it is scientifically valuable to explore distributed PCA on larger astronomical data collections and for greater numbers of astrophysical parameters. The application of communication-efficient distributed PCA and other DDM algorithms will likely enable the discovery of new fundamental planes, and thus produce new scientific insights into our Universe.

2 Related Work

2.1 Analysis of Large Data Collections There are several instances in the astronomy and space sciences research communities where data mining is being applied to large data collections [23]. Some dedicated data mining projects include F-MASS [12], Class-X [10], the Auton Astrostatistics Project [3]. In essentially none of these cases does the project involve truly DDM [22]. Through a past NASA-funded project, K. Borne applied some very basic DDM concepts to astronomical data mining [8]. However, the primary accomplishments focused only on centralized co-location of the data sources [6, 7].

One of the first large-scale attempts at grid data mining for astronomy is the U.S. National Science Foundation (NSF) funded GRIST [14], project. The GRIST goals include application of grid computing and web services (service-oriented architectures) to mining large distributed data collections. GRIST is focused on one particular data modality: images. Hence, GRIST aims to deliver mining on the pixel planes within multiple distributed astronomical image collections. The project that we are proposing here is aimed at another data modality: catalogs (tables) of astronomical source attributes. GRIST and other projects also strive for exact results, which usually requires data centralization and co-location, which further requires significant computational and communications resources. DEMAC (our system) will produce approximate results without requiring data centralization (low communication overhead). Users can quickly get (generally quite accurate) results for their distributed queries at low communication cost. Armed with these results, users can focus in on a specific query or portion of the datasets, and down-load for more intricate analysis.

The U.S. National Virtual Observatory (NVO) [25] is a large scale effort funded by the NSF to develop a information technology infrastructure enabling easy and robust access to distributed astronomical archives. It will provide services for users to search and gather data across multiple archives and some basic statistical analysis and visualization functions. It will also provide a framework for new services to be made available by outside parties. These services can provide, among other things, specialized data analysis capabilities. As such, we envision DEMAC to fit nicely into the NVO as a new service.

2.2 Distributed Data Mining DDM is a relatively new technology that has been enjoying considerable interest in the recent past [18]. DDM algorithms strive to analyze the data in a distributed manner without down-loading all of the data to a single site (which is usually necessary for a regular centralized data mining system). DDM algorithm naturally fall into two categories according to whether the data is distributed horizontally (with each site having some of the tuples) or vertically (with each site having some of the attributes for all tuples). In the latter case, it is assumed that the sites have an associated unique id used for matching. In other words, consider a tuple t and assume site A has a part of this tuple, t_A, and B has the remaining part t_B. Then, the id associated with t_A equals the id associated with t_B.[1]

The NVO can be seen as a case of vertically distributed data, assuming ids have been generated by a cross-matching service. With this assumption, DDM algorithms for vertically partitioned data can be applied. These include algorithms for principal component analysis (PCA), Bayesian network learning, clustering, and supervised classification (see [18] for references).

Some DDM frameworks and systems have been developed. The JAM framework for meta-learning based classification over homogeneously distributed data was developed by Stolfo *et al.* [27]. The Collective Data Mining framework for data mining over heterogeneously distributed data was developed by Kargupta *et al.* [15]. In this framework an algorithm for distributed PCA was developed. We leave the comparison of this algorithm to the one we give later for future work. A client-server architecture for a data mining system, the Kensington System, we developed by Chattratichat *et al.* [9].

3 Data Analysis Problem: Analyzing Distributed Virtual Catalogs

We illustrate the problem with two archives: the Sloan Digital Sky Survey (SDSS) [26] and the 2-Micron All-Sky Survey (2MASS) [1]. Each of these has a simplified catalog containing records for a large number of astronomical point sources, upward of 100 million for SDSS and 470 million for 2MASS. Each record contains sky coordinates (ra,dec) identifying the sources' position in the celestial sphere as well as many other attributes (460+ for SDSS; 420+ for 2MASS). While each of these catalogs individually provides valuable data for scientific exploration, together their value increases significantly. In particular, efficient analysis of the *virtual catalog* formed by joining these catalogs would enhance their scientific value significantly. Henceforth, we use "virtual catalog" and "virtual table", interchangeably.

To form the virtual catalog, records in each catalog must first be matched based on their position in the

[1] Each id is unique to the site at which it resides; no two tuples at site A have the same id.

celestial sphere. Consider record t from SDSS and s from 2MASS with sky coordinates $t[ra, dec]$ and $s[ra, dec]$. Each record represents a set of observations about an astronomical object *e.g.* a galaxy. The sky coordinates are used to determine if t and s match, *i.e.* are close enough that t and s represents the same astronomical object. The issue of how matching is done will be discussed later. For each match (t, s), the result is a record $t \bowtie s$ in the virtual catalog with all of the attributes of t and s. As described earlier, the virtual catalog provides valuable data that neither SDSS or 2MASS alone can provide.

DEMAC addresses the data analysis problem of developing communication-efficient algorithms for analyzing user-defined subsets of virtual catalogs. The algorithms allow the user to specify a region R in the sky and a virtual catalog, then efficiently analyze the subset of tuples from that catalog with sky coordinates in R. Importantly, the algorithms we develop do not require that the base catalogs first be centralized and the virtual catalog explicitly realized. Moreover, the algorithms are not intended to be a substitute for exact, centralization-based methods currently being developed as part of the NVO. Rather, they are intended to complement these methods by providing, quick, communication-efficient approximate results to allow browsing. Such browsing will allow the user to better focus their exact, communication-expensive, queries.

EXAMPLE 1. *The all data release of 2MASS contains attribute, "K band means surface brightness" (Kmsb). Data release four of SDSS contains galaxy attributes "redshift" (rs), "petrosian I band angular effective radius" (Iaer) and "velocity dispersion" (vd). To produce a physical variable, consider composite attribute "petrosian I band effective radius" (Ier) formed by the product of Iaer and rs. Note, since Iaer and rs are both at the same repository (SDSS), then, from the standpoint of distributed computation, we may assume Ier is contained in SDSS.*

A principal component analysis over a region of sky R on the virtual table with columns log(Ier), log(vd), and Kmsb is interesting in that it can allow the identification of a "fundamental plane" (the logarithms are used to place all variables on the same scale). Indeed, if the first two principal components capture most of the variance, then these two variables define a fundamental plane. The existence of such things points to interesting astrophysical behaviors. We develop a communication-efficient distributed algorithm for approximating the principal components of a virtual table.

4 DEMAC - A System for Distributed Exploration of Massive Astronomical Catalogs

This section describes the high level architecture design of the DEMAC system. DEMAC is designed as an additional web-service which seamlessly integrates into the NVO. It consists of two basic services. The main one is a web-service providing DDM capabilities for vertically distributed sky surveys (*WS-DDM*). The second one, which is intensively used by WS-DDM, is a web-service providing cross-matching capabilities for vertically distributed sky surveys (*WS-CM*). Cross-matching of sky surveys is a complex topic which is dealt with, in itself, under other NASA funded projects. Thus, our implementation of this web-service is designed to supply bare minimum capabilities which are required in order to provide distributed data mining capabilities.

To provide a distributed data mining service, DEMAC relies on other services of the NVO such as the ability to select and down-load from a sky survey in an SQL-like fashion. Key to our approach is that these services be used not over the web, through the NVO, but rather by local agents which are co-located with the respective sky survey. In this way, the DDM service avoids bandwidth and storage bottlenecks, and overcomes restrictions which are due to data ownership concerns. Agents, in turn, take part in executing efficient distributed data mining algorithms, which are highly communication-efficient. It is the outcome of the data mining algorithm, rather than the selected data table, that is provided to the end-user. With the removal of the network bandwidth bottleneck, the main factor limiting the scalability of the distributed data mining service would be database access. For database access, DEMAC uses the SQL-like interface provided to the different sky-surveys to the NVO.

We outline here the architecture for the two web-services.

4.1 WS-DDM – DDM for Heterogeneously Distributed Sky-Surveys This web-service allows running a DDM algorithm (one will be discussed later) on a selection of sky-surveys. The user applies existing NVO services to locate sky-surveys and define the portion of the sky to be data mined. The user then applies WS-CM to select a cross-matching scheme for those sky-surveys. This specifies how the tuples are matched across surveys to define the virtual table to be analyzed. Following these two preliminary phases the user submits the data mining task.

Execution of the data mining task is scheduled according to resource availability. Specifically, the size of the virtual table selected by the user dictates

scheduling. Having allocated the required resources, the data mining algorithm is carried on by agents which are co-located with the selected sky-surveys. Those agents access the sky-survey through the SQL-like interface it exposes to the NVO and communicate with each other directly, over the Internet. When the algorithm has terminated, results are provided to the user using a web-interface.

4.2 WS-CM – Cross-Matching for Heterogeneously Distributed Sky-Surveys Central to the DDM algorithms we develop is that the virtual table can be treated as vertically partitioned (see Section 2 for the definition). To achieve this, *match indices* are created and co-located with each sky survey. Specifically, for each pair of surveys (tables) T and S, a distinct pair of match indices must be kept, one at each survey. Each index is a list of pointers; both indices have the same number of entries. The i^{th} entry in $T's$ list points to a tuple t_i and the i^{th} entry in $S's$ list points to s_i such that t_i and s_i match. Tuples in T and S which do not have a match, do not have a corresponding entry in either index. Clearly, algorithms assuming a vertically partitioned virtual table can be implemented on top of these indices.

Creating these indices is not an easy job. Indeed, cross-matching sources is a complex problem for which no single best solution exists. The WS-CM web-service is not intended to address this problem. Instead it uses already existing solutions (*e.g.*, the cross-matching service already provided by the NVO), and is designed to allow other solutions to be plugged in easily. Moreover, cross-matching the entirety of two large surveys is a very time-consuming job and requires centralizing (at least) the ra, dec coordinates of all tuples from both.

Importantly, the indices do not need to be created each time a data mining task is run. Instead, provided sky survey data is static (it generally is), each pair of indices only need be created *once*. Then any data mining task can use them. In particular the DDM tasks we develop can use them. The net result is the ability to mine virtual tables at low communication cost.

Note that for each group of surveys and each cross-matching scheme of interest, there need be an separate index held at each survey. However, again, these indeces are computed off-line (not at query time). Thus, keeping different indeces for each group and scheme does not affect query-processing communication cost.

4.3 Definitions and Notation In the next section we describe a DDM algorithm to be used as part of the WS-DDM web service. It assumes that the participating sites have the appropriate alignment indices. Hence, for simplicity, we describe the algorithms under the assumption that the data in each site is perfectly aligned – the i^{th} tuple of each site match (sites have exactly the same number of tuples). This assumption can be emulated without problem using the matching indices.

Let M denote an $n \times m$ matrix with real-valued entries. This matrix represents a dataset of n tuples from $\mathbb{R}^m$. Let M^j denote the j^{th} column and $M^j(i)$ denote the i^{th} entry of this column. Let $\mu(M^j)$ denote the *sample mean* of this column, $\frac{\sum_{i=1}^{n} M_j(i)}{n}$. Let $Var(M^j)$ denote the *sample variance* of this column, $\frac{\sum_{i=1}^{n}[\mu(M^j)-M^j(i)]^2}{n-1}$. Let $Cov(M^j, M^k)$ denote the *sample covariance* of the j^{th} and k^{th} columns, $\frac{\sum_{i=1}^{n}[\mu(M^j)-M^j(i)][\mu(M^k)-M^k(i)]}{n-1}$. Note, $Var(M^j) = Cov(M^j, M^j)$). Finally, let $Cov(M)$ denote the *covariance matrix* of M *i.e.* the $m \times m$ matrix whose $(j,k)^{th}$ entry is $Cov(M^j, M^k)$.

Assume this dataset has been vertically distributed over two sites S_A and S_B. Since we are assuming that the data at the sites is perfectly aligned, then S_A has the first p attributes and S_B has the last q attributes $(p+q=m)$. Let A denote the $n \times p$ matrix representing the dataset held by S_A, and B denote the $n \times q$ matrix representing the dataset held by S_B. Let $A:B$ denote the concatenation of the datasets *i.e.* $M = A:B$. The j^{th} column of $A:B$ is denoted $[A:B]^j$.

Next we describe a communication-efficient algorithm for PCA on M vertically distributed over two sites. The algorithm easily extends to more than two sites, but, for simplicity, we only discuss the two site scenario. Later we examine its effectiveness through a case study. We have also developed a distributed algorithm for decision tree induction (supervised classification) [13] and are in the process of developing a distributed algorithm for outlier detection.

Following a standard practise in applied statistics, we pre-process M by normalizing so that $\mu(M^j) = 0$ and $Var(M^j) = 1$. This is achieved by replacing each entry $M^j(i)$ with $\frac{M^j(i)-\mu(M^j)}{\sqrt{Var(M^j)}}$. Since both $\mu(M^j)$ and $Var(M^j)$ can be computed without any communication, then normalizing can be performed without any communication. Henceforth, we assume $\mu(M^j) = 0$ and $Var(M^j) = 1$.

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$ denote the eigenvalues of $Cov(M)$ and $v_1, v_2, \ldots, v_m$ the associated eigenvectors[2] (pairwise orthonormal). The j^{th} *principal direction* of M is v_j. The j^{th} *principal component* is denoted z_j and equals Mv_j (the projection of M along the j^{th} direction).

[2]We assume the eigenvectors are column vectors *i.e.* $m \times 1$ matrices.

5 Virtual Catalog Principal Component Analysis

PCA is a well-established data analysis technique used in a large number of disciplines: astronomy, computer science, biology, chemistry, climatology, geology, *etc.* Quoting [16] page 1: "The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the dataset." Next we provide a very brief overview of PCA, for a more detailed treatment, the reader is referred to [16].

The j^{th} principal component, z_j, is, by definition, a linear combination of the columns of M – the k^{th} column has coefficient $v_j(k)$. The sample variance of z_j equals λ_j. The principal components are all uncorrelated *i.e.* have zero pairwise sample covariances. Let Z_r ($1 \leq r \leq m$) denote the $n \times r$ matrix with columns $z_1, \ldots, z_r$. This is the dataset projected onto the subspace defined by the first r principal directions. If $r = m$, then Z_m is simply a different way of representing exactly the same dataset, because M can be recovered completely as $M = Z_m V^T$ where T denotes matrix transpose.[3]

However, if $r < m$, then Z_r is a lossy lower dimensional representation of M. The amount of loss is typically quantified as, $\frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{m} \lambda_j}$, the "proportion of variance" captured by the lower dimensional representation. If r is chosen so that a large amount of the variance is captured, then, intuitively, Z_r, captures many of the important features of M. So, subsequent analysis on Z_r can be quite fruitful at revealing structure not easily found by examination of M directly. Our case study employs this idea.

To our knowledge, the problem of vertically distributed PCA computation was first addressed by Kargupta *et al.* [19] based on sampling and communication of dominant eigenvectors. Later, Kargupta and Puttagunta [17] developed a technique based on random projections. Our method is a slightly revised version of this work. We describe a distributed algorithm for approximating $Cov(A : B)$. Clearly, PCA can be performed from $Cov(A : B)$ without any further communication.

Recall that $A : B$ is normalized to have zero column sample mean and unit sample variance. As a result, $Cov([A : B]^j, [A : B]^k) = \frac{\sum_{i=1}^{n} [A:B]^j(i)[A:B]^k(i)}{n-1}$ which is the *inner product* between $[A : B]^j$ and $[A : B]^k$ divided by $n - 1$. Clearly this inner product can be computed without communication when $[A : B]^j$ and $[A : B]^k$ are at the same site (*i.e.* $1 \leq j, k \leq p$ or $p + 1 \leq j, k \leq p + q$). It suffices to show how the inner product can be approximated across different sites, in effect, how $A^T B$ can be approximated. The key idea is based on the following fact echoing the observation made in [24] that high-dimensional random vectors are nearly orthogonal. A similar result was proved elsewhere [2].

FACT 1. *Let R be an $\ell \times n$ matrix each of whose entries is drawn independently from a distribution with variance one and mean zero. It follows that $E[R^T R] = \ell I_n$ where I_n is the $n \times n$ identity matrix.*

We will use the Algorithm 1 for computing $A^T B$. The result is obtained at both sites.[4] The algorithm has a user-defined parameter ℓ (the number of rows in the random matrix).

Algorithm 1 Distributed Covariance Matrix Algorithm

1. S_A sends S_B a random number generator seed. [**1 message**]
2. S_A and S_B generate a $\ell \times n$ random matrix R where ℓ. Each entry is generated independently and identically from any distribution with mean zero and variance one.
3. S_A sends RA to S_B; S_B sends RB to S_A. [$4m\ell$ **messages**]
4. S_A and S_B compute $D = \frac{(RA)^T (RB)}{\ell}$.

From Fact 1, it can be seen that $E[D] = \frac{A^T E[R^T R] B}{\ell} = A^T B$. Hence, on expectation, the algorithm is correct. However, its communication cost (bytes) divided by the cost of the centralization-based algorithm, $\frac{\ell}{n} + \frac{1}{nm}$, is small if $\ell << n$. Indeed ℓ provides a "knob" for tuning the trade-off between communication-efficiency and accuracy. Later we present experiments measuring this trade-off.

6 Case Study: Finding Galactic Fundamental Planes

The identification of certain correlations among parameters has lead to important discoveries in astronomy. For example, the class of elliptical and spiral galaxies (including dwarfs) have been found to occupy a two dimensional space inside a three dimensional space of observed parameters, radius, mean surface brightness and velocity dispersion (described earlier in Example 1). This plane has been referred to as the *Fundamental Plane*([11, 20]).

[3] Since V is a square matrix with orthonormal columns, then basic linear algebra shows that VV^T equals the $m \times m$ identity matrix.

[4] In the communication cost calculations, we assume a message requires 4 bytes of transmission.

This section presents a case study involving the detection of a fundamental plane among galaxy parameters distributed across two catalogs: 2MASS and SDSS (the problem was described earlier in Example 1). Our goal in this paper is to demonstrate that, using our distributed covariance matrix algorithm to approximate the principal components, we can find a very similar fundamental plane as that obtained by applying a centralized PCA. Our ultimate goal is to enable new discoveries in astronomy through our DDM algorithms and DEMAC system. Therefore, we argue that DEMAC could provide a valuable tool for astronomers wishing to explore many parameter spaces across different catalogs for fundamental planes.

In our study we measure the accuracy of our distributed algorithm in terms of the similarity between its results and those of a centralized approach. We examine accuracy at various amounts of communication allowed the distributed algorithm in order to assess the tradeoff described at the end of Section 5. For each amount of communication allowed, we ran the distributed algorithm 100 times with a different random matrix and report the average result (except where otherwise noted). For the purposes of our study, a real distributed environment is not necessary. Thus, for simplicity, we used a single machine and simulated a distributed environment.

Comments: We acknowledge that this simplified experimental environment does not take into account overhead imposed by web services (DEMAC is designed to reside on top of the NVO as a web service), *e.g.* variable resource availability. However, the purpose of this study is to examine the basic accuracy versus communication tradef-off of our algorithm in order to justify its potential use. A more detailed, real-world study is left to future work. We also acknowledge that a straight-forward uniform sampling technique could also be used to approximate PCA. The sites choose a uniform sample of entries in their allignment indeces (the same ids on each site), and centralize the corresponding tuples. These tuples are then joined to form a unifrom sample of the virtual catalog of interest. We to future work a comparison between this sampling technique and our PCA algorithm.

We prepare our test data as follows. Using the web interfaces of 2MASS[5], SDSS[6] and and the SDSS object cross id tool, we obtained an aggregate dataset involving attributes from 2MASS and SDSS lying in the sky region between right ascension (ra) 150 and 200, declination (dec) 0 and 15. The aggregated dataset had the following attributes from SDSS: Petrosian I band angular effective radius (*Iaer*), redshift (*rs*), and velocity dispersion (*vd*);[7] and had the following attribute from 2MASS: K band mean surface brightness (*Kmsb*).[8] After removing tuples with missing attributes, we had a 1307 tuple dataset with four attributes. We produced a new attribute, logarithm Petrosian I band effective radius (*log(Ier)*), as log(Iaer*rs) and a new attribute, logarithm velocity dispersion (*log(vd)*), by applying the logarithm to vd. We dropped all attributes except those to obtain the three attribute dataset, log(Ier), log(vd), Kmsb. Finally, we normalized each column by subtracting its mean from each entry and dividing by its sample standard deviation (as described in Section 4.3).

We applied PCA directly to this dataset to obtain the centralization based results. Then we treated this dataset as if it were distributed (assuming cross match indices have been created as described earlier).[9] This data can be thought of as a virtual table with attributes log(Ier) and log(vd) located at one site and attribute Kmsb at another. Finally, we applied our distributed covariance matrix algorithm and computed the principal components from the resulting matrix. Note, our dataset is somewhat small and not necessarily indicative of a scenario where DEMAC would be used in practice. However, for the purposes of our study (accuracy with respect to communication) it suffices. Note that producing even a centralized, cross-matched data set of this size was a tedious job. This is because of the communication limitations currently imposed by the NVO (the cross-matching tool has an upper-limit on the number of tuples in its output). This fact only bolsters support for our system as it aims to reduce communication.

Figure 2 shows the percentage of variance captured as a function of communication percentage (*i.e.* at 15%, the distributed algorithm uses $0.15(29638)(3) = 589$ bytes). Error bars indicate standard deviation – recall the percentage of variance captured numbers are averages over 100 trials. First observe that the percentage captured by the centralized approach, 90.5%, replicates the known result that a fundamental plane exists among these parameters. Indeed the dataset fits fairly nicely on the plane formed by the first two PCs. Also observe that

[5] http://irsa.ipac.caltech.edu/applications/Gator/

[6] http://cas.sdss.org/astro/en/tools/crossid/upload.asp

[7] *petroRad_i* (galaxy view), z (SpecObj view) and velDisp (SpecObj view) in SDSS DR4

[8] *k_mnsurfb_eff* in the extended source catalog in the All Sky Data Release, http://www.ipac.caltech.edu/2mass/releases/allsky/index.html

[9] All of the preprocessing steps described above could have been carried out without any distributed computation, thus, need not enter into our simulation.

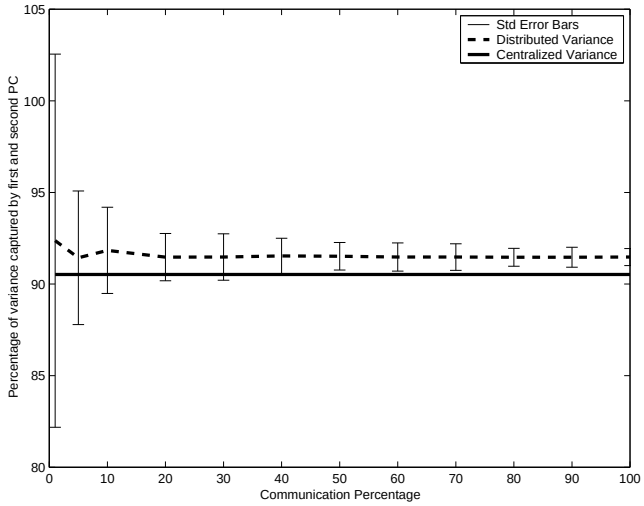


Figure 2: Communication percentage vs. percent of variance captured, (log(Ier), log(vd), Kmsb) dataset.

the percentage of variance captured by the distributed algorithm (including one standard deviation) using as little as 10% communication never strays more than 5 percent from 90.5%. This is a reasonibly accurate result indicating that the distributed algorithm identifies the existence of a plane using 90% less communication. As such, this provides evidence that the distributed algorithm would serve as a good browser allowing the user to get decent approximate results at a sharply reduced communication cost. If this piques the user's interest, she can go through the trouble of centralizing the data and carrying out an exact analysis.

Interestingly, the average percentage captured by the distributed algorithm appears to approach the true percentage captured, 90.5%, very slowly (as the communication percentage approaches infinity, the average percentage captured must approach 90.5%). At the present we don't have an explanation for the slow approach. However, as the communication increases, the standard deviation decreases substantially (as expected).

To analyze the accuracy of the actual principal components computed by the distributed algorithm, we consider the data projected onto each pair of PCs. The projection onto the true first and second PC ought to appear with much scatter in both directions as it represents the view of the data perpendicular to the plane. And, the projections onto the first, third and second, third PCs ought to appear more "flattened" as they represent the view of the data perpendicular to the edge of the plane. Figure 3 displays the results. The left column depicts the projections onto the PCs computed by the centralized analysis (true PCs). Here we see the fundamental plane. The right column depicts the projections onto the PCs computed by our distributed algorithm at 15% communication (for one random matrix and *not* the average over 100 trials). We see a similar pattern, indicating that the PCs computed by the distributed algorithm are quite accurate in the sense that they produce very similar projections as those produced by the true PCs.

Furthermore we quantitatively compared the projections onto the true PCs ("true projections") and those onto the PCs computed by the distributed algorithm ("distributed projections") as follows. Let $x_1, \ldots, x_n$ denote a true projection (say onto the first and second PCs) and $y_1, \ldots, y_n$ the corresponding distributed projection. Here x_i is the i^{th} data tuple projected onto the true PCs and y_i is the projection of the i^{th} data tuple onto the corresponding PCs computed by the distributed algorithm. The normalized, square distance (NSD) between two tuples is $\frac{||x_i - y_i||}{||x_i||}$ (where $||.||$ denotes the 2-norm). For each of the three pairs of PCs, we computed the average NSD and its standard deviation. For PCs 1 and 2, 1 and 3, 2 and 3, the averages and standard deviations are 0.007 and 0.00069, 0.0048 and 0.00056, 0.0036 and 0.00022. These are quite small indicating very good accuracy (in all cases the average plus four standard deviations is less than 0.01).

In closing, it is important to stress that we are not claiming that the actual projections can be computed in a communication-efficient fashion (they can't). Rather, that the PCs computed in a distributed fashion are accurate as measured by the projection similarity with the true PCs.

7 Conclusions

We described the arcitecture of a system, *DEMAC*, for the distributed exploration of massive astronomical catalogs. DEMAC is deisgned to reside on top of the existing U.S. national virtual observatory environment and provide tools for data mining (as web services) without requiring datasets to be down-loaded to a centralized server. Instead, the users only down-load the output of the data mining process (a data mining model); the actual data mining from multiple data servers are performed using communication-efficient DDM algorithms. The distributed algorithms we have developed sacrifice perfect accuracy for communication savings. They offer approximate results at a considerably lower communication cost than that of exact results through centralization. As such, we see DEMAC as serving the role of an exploratory "browser". Users can quickly get (generally quite accurate) results for their distributed queries at low communication cost. Armed with these results, users can focus in on a specific query or portion of the

datasets, and down-load for more intricate analysis.

To illustrate the potential effectiveness of our system, we carried out a case study using distributed principal component analysis (PCA) for detecting fundamental planes of astronomical parameters. We observed our distributed algorithm to identify a fundamental plane (observed through centralized analysis) at reduced communication cost.

In closing, we envision our system to increase the ease with which large, geographically distributed astronomy catalogs can be explored, by providing quick, low-communication solutions. Such benefit will allow astronomers to better tap the riches of distributed virtual tables formed from joined and integrated sky survey catalogs.

Acknowledgments

H. Dutta, C. Giannella, H. Kargupta, R. Wolff: Support for this work was provided by the U.S. National Science Foundation (NSF) through grants IIS-0329143, IIS-0093353, IIS-0203958 and by the U.S. National Aeronautics and Space Agency (NASA) through grant NAS2-37143.

K. Borne: Support for this work was provided in part by the NSF through Cooperative Agreement AST0122449 to the Johns Hopkins University.

References

[1] 2-Micron All Sky Survey. http://pegasus.phast.umass.edu.

[2] Arriaga R. and Vempala S. An Algorithmic Theory of Learning: Robust Concepts and Random Projection. In *Proceedings of the 40th Foundations of Computer Science*, 1999.

[3] The AUTON Project. http://www.autonlab.org/autonweb/showProject/3/.

[4] Borne K. Science User Scenarios for a VO Design Reference Mission: Science Requirements for Data Mining. In *Virtual Observatories of the Future, San Francisco: Astronomical Society of the Pacific*, page 333, 2000.

[5] Borne K. Data Mining in Astronomical Databases. In *Proceedings of Mining the Sky, Heidelberg: Springer-Verlag*, page 671, 2001.

[6] Borne K. Distributed Data Mining in the National Virtual Observatory. In *Proceedings of the SPIE Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology V, Vol. 5098*, page 211, 2003.

[7] Borne K., Arribas S., Bushouse H., Colina L., and Lucas R. A National Virtual Observatory (NVO) Science Case. In *Proceedings of the Emergence of Cosmic Structure, New York: AIP*, page 307, 2003.

[8] Distributed Data Mining Techniques for Object Discovery in the National Virtual Observatory. http://is.arc.nasa.gov/IDU/tasks/NVODDM.html.

[9] Chattratichat J., Darlington J., Guo Y., Hedvall S., Koler M., and Syed J. An Architecture for Distributed Enterprise Data Mining. In *Lecture Notes in Computer Science*, volume 1593, pages 573–582. Springer-Verlag, 1999.

[10] The ClassX Project: Classifying the High-Energy Universe. http://heasarc.gsfc.nasa.gov/classx/.

[11] Elliptical Galaxies: Merger Simulations and the Fundamental Plane. http://irs.ub.rug.nl/ppn/244277443.

[12] Framework for Mining and Analysis of Space Science Data. http://www.itsc.uah.edu/f-mass/.

[13] Giannella C., Liu K., Olsen T., and Kargupta H. Communication Efficient Construction of Decision Trees Over Heterogeneously Distributed Data. In *Proceedings of the The Fourth IEEE International Conference on Data Mining (ICDM)*, 2004.

[14] GRIST: Grid Data Mining for Astronomy. http://grist.caltech.edu.

[15] Hargupta H., Park B., Hershberger D., and Johnson E. Collective Data Mining: A New Perspective Toward Distributed Data Mining. In Kargupta H. and Chan P., editors, *Advances in Distributed and Parallel Knowledge Discovery*, pages 133–184. MIT/AAAI Press, 2000.

[16] Jolliffe I. *Principal Component Analysis*. Springer-Verlag, 2002.

[17] Kargupta H. and Puttagunta V. An Efficient Randomized Algorithm for Distributed Principal Component Analysis from Heterogeneous Data. In *Proceedings of the Workshop on High Performance Data Mining in conjunction with the Fourth SIAM International Conference on Data Mining*, 2004.

[18] Kargupta H. and Sivakumar K. Existential Pleasures of Distributed Data Mining. In *Data Mining: Next Generation Challenges and Future Directions, MIT/AAAI Press*, pages 3–26, 2004.

[19] Kargupta H., Huang W., Sivakumar K., and Johnson E. Distributed clustering using collective principal component analysis. *Knowledge and Information Systems Journal*, 3:422–448, 2001.

[20] Jones L. and Couch W. A Statistical Comparison of Line Strength Variations in Coma and Cluster Galaxies at $z \sim 0.3$. *Astronomical Society, Australia*, 15:309–317, 1998.

[21] McDowell J. Downloading the Sky. *IEEE Spectrum*, 41:35, 2004.

[22] Distributed Data Mining in Astrophysics and Astronomy. http://www.cs.queensu.ca/home/mcconell/DDMAstro.html.

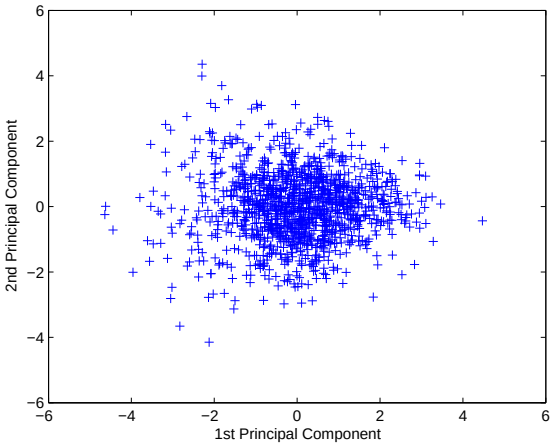
[23] NASA's Data Mining Resources for Space Science. http://rings.gsfc.nasa.gov/~borne/nvo_datamining.html.

[24] Nielsen R. Context Vectors: General Purpose Approximate Meaning Representations Self-organized From Raw Data. In *Computational Intelligence: Imitating Life*, pages 43–56. IEEE Press, 1994.

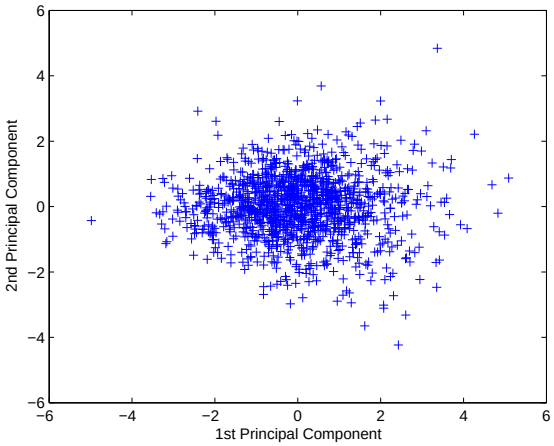
[25] US National Virtual Observatory. http://www.us-vo.org/.

[26] Sloan Digital Sky Survey. http://www.sdss.org.

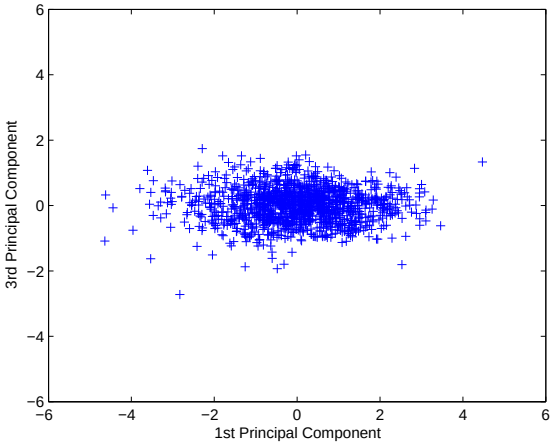
[27] Stolfo S., Prodromidis A., Tselepis S., Lee W., Fan D. JAM: Java Agents for Meta-Learning Over Distributed Databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 74–81, 1997.



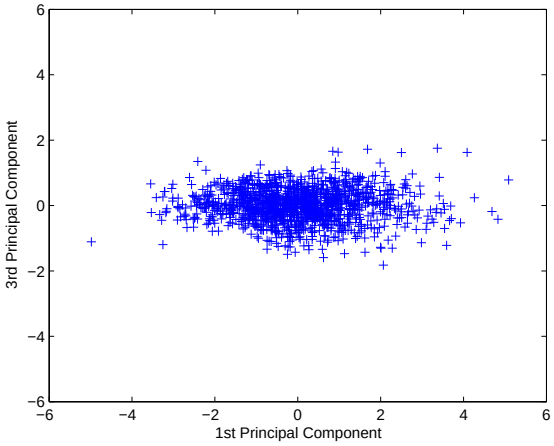
(a) PCs from centralized analysis



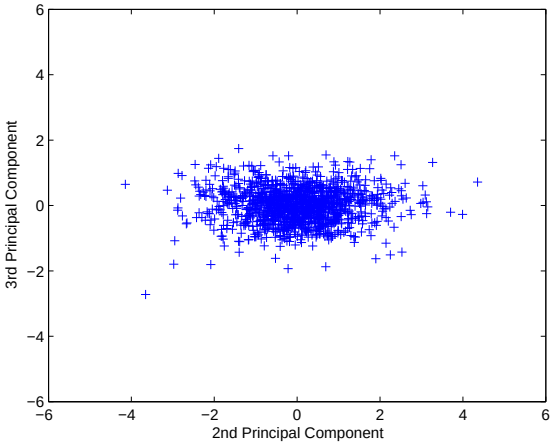
(b) PCs from distributed algorithm



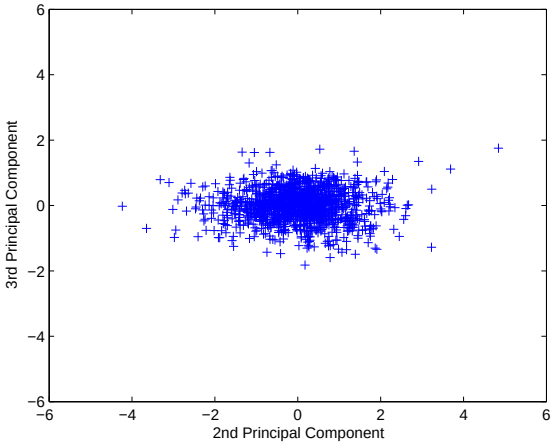
(c) PCs from centralized analysis



(d) PCs from distributed algorithm



(e) PCs from centralized analysis



(f) PCs from distributed algorithm

Figure 3: Projections onto all pairs of PCs; communication percentage 15%, (log(Ier), log(vd), Kmsb) dataset.