ABSTRACT

Electric power infrastructure is rapidly running up against oversized growth, scale and efficiency. Electricity production, distribution and consumption play a critical role in the sustainability of the planet and its natural resources. Smart Grids which enable two-way communication and monitoring between producers and end-users need novel computational algorithms for supporting generation of power from wide range of sources, efficient energy distribution, and sustainable consumption. This paper explores fundamentally distributed approaches with more local flexibility leading to sustainable methodology compared to the traditional centralized frameworks for analyzing and processing data. The paper considers the problems of aggregation and prediction of power generation and consumption trends over a distributed smart grid. The need for more local control, privacy issues, and cost sensitivity for transmission of remote sensory data over the low-bandwidth wireless network is leading toward more distributed approach to data analysis in smart grids. This paper reviews our recent work on more sustainable distributed asynchronous methodology for constructing energy demand prediction models in a smart grid by multivariate linear regression as well as dynamic pricing model built on distributed rank aggregation that will help shape power consumption and optimize the grid.

Categories and Subject Descriptors

H.3.4 [Information Storage and Retrieval]: Systems and Software—Distributed systems; I.6.7 [Simulation and Modeling]: Simulation Support Systems—Environments;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SustKDD 2011 August 2011, San Diego, CA, USA
Copyright 2011 ACM 978-1-4503-0840-3 ...$10.00.

1. INTRODUCTION

Sustainability in Electric Power systems is a very important issue since it usually has a significant impact on the environment and other systems that share the same environment. A system or a process is sustainable when its input and output have little adverse impact on its neighbors and therefore can be accepted as a long term practice. A system that is not sustainable often leads to the failure (sometimes catastrophic) of the system itself or other systems in its environment. Electric power systems are undergoing profound changes driven by a number of needs. Environmental impact, reliability, operational efficiencies in energy generation and distribution along with alternate power generation technologies and “intelligent” appliances are driving the need for developing the new generation of energy networks — the Smart Grids. A sustainable Smart Grid would deliver high performance at the right cost with little impact on the environment. This paper argues that this demands making the “smart” in the Smart Grid really smart by deploying proper adaptive machine learning and data analysis techniques for energy production and distribution.

The paper offers an overview some recently developed distributed data analysis techniques that work in a way very different from traditional centralized approaches. It specifically considers the problem of multivariate regression and rank ordering in a distributed scenario. Multivariate regression model is ubiquitously used for predicting, forecasting and have substantial impact in the area of machine learning theory. Whereas ordering several objects and obtaining a ranking among them like commonly encountered in electoral systems, now has several applications in the field
of information retrieval such as meta search, recommendation systems and combining any general ranking function. Given vast amount of data and least human intervention we are looking to automatically recognize complex patterns and make intelligent decisions. Smart Grid’s physical and operational state along with environmental sensor data are being available as both real time and from archival stores. They contain important correlations, trends, and patterns that can be exploited for optimizing operations with respect to sustainability metrics, such as, energy consumption, carbon footprint and even dynamic pricing. These data mining techniques and analytics will extract prediction rules, which when embedded in distributed, decision support or real time data engines will help shape electric consumption and optimize the grid leading to greater sustenance.

The rest of the paper is organized as follows. Section 2 presents the background and related work. Section 3 describes the problem of predicting demand, the technical approach, and some preliminary results. Section 4 describes the rank aggregation problem, the technical approach, and preliminary results. Finally, Section 5 concludes the paper with prospective future works and directions.

2. BACKGROUND

Most of current methodological thinking in Smart Grid is layered on hardware infrastructure development and fundamentally centralized architecture for information processing. The power companies are primarily in charge of centrally managing the data generated by the different sensors embedded in the smart grid at different points. In this paper the proposed architectures follow decentralized peer-to-peer (P2P) paradigm where power companies can have central control over their own assets; but individual residence and business clients can also participate by creating a network of peers for fundamentally distributed information processing. The approach is based on distributed data mining technology where local hubs use the currently available hardware to process the data locally instead of centralizing everything to a single server. Knowledge discovery, information integration and developing distributed algorithms for extracting information from data sources distributed over a Smart Grid requires a fundamentally different breed of data mining algorithms. It will require distributed data mining techniques that will work in a decentralized distributed manner and scale up to millions of nodes. The main important characteristics of these algorithms include: (i) the ability to efficiently scale-up to peer-to-peer systems that exist today consisting of millions of peers, (ii) the ability to calculate the result “locally” rather than collect all of the data to a central processor (which would quickly exhaust bandwidth in peer-to-peer networks) and (iii) the ability to function correctly in the presence of peer/edge failures and data change. In order to offer scalable asynchronous data mining solutions the paper investigates polynomially bounded “local” computations. Despite the growing availability of distributed data mining algorithms, most of them do not offer local decomposability where the communication-load per node is either constant or a slow growing polynomial with respect to the “size” of network (e.g. number of nodes). Such polynomially bounded communication load per node is very important for scalable performance of distributed data mining algorithms. In this paper we outline our research in developing algorithms for Smart Grid applications and demonstrate some performance experimentally.

Approaches to distributed and P2P data mining have focused on developing some primitive operations as well as complicated data mining algorithms. Jelasity and Eiben [15] developed the “newscast model” showed how to calculate primitive operations on P2P networks. Kempe et al [14] investigated gossip based randomized algorithms and proved that the error goes down to zero in probability if the algorithm runs uninterrupted. Both of the above uses an epidemic model of computation. Local algorithms [3] are also studied which can be used to compute results using information from a handful of nearby neighbors in a peer-to-peer system with definite claims regarding its correctness. Wolff and Schuster [19] has developed local algorithm for computing majority vote over a peer-to-peer network. Monitoring algorithms are also proposed for P2P network by Wolff [19] for monitoring K-means clustering. Distributed multivariate regression has been addressed by many researchers till date. Hershberger et al.[13] considered the problem of solving global regression coefficients in a vertically partitioned data distribution scenario. This is a synchronized algorithm where each node computes the wavelet transformation of the data and exchanges the significant coefficients in a synchronous manner in order to regress the global coefficients of the model. Algorithms presented by Guestrin et al.[12] performs linear regression in a network of sensors using in-network processing of messages where instead of transmitting raw data it transmits only constraints thereby reducing communication complexity. Bhaduri et al[4] addresses the problem by providing a scalable local algorithm for multivariate regression. This paper makes use of a convergecast phase where data is sampled from the network to a central post to compute the coefficients based on the samples followed by a broadcast phase where these coefficients are distributed into the network and the results are monitored in-node. Though the root of the convergecast tree is not pre-specified it inherently doesn’t steer clear of the centralized approach once the tree is built and subsequently during tie resolution during broadcast phase. A survey of challenges related to computational sustainability in general and that of Smart Grids in the area of planning and operating large complex digital ecosystems, controlling and measuring technologies from a producer controlled network to a more decentralized system has been showed by Carla P

Figure 1: Typical architecture of a Smart Grid.¹

---

¹Courtesy U.S Government Accountability Office
Gomes et. al [11]. There exists a body of literature dealing with algorithms and systems-related challenges for information processing over smart grids. Hurdles like standards of interoperability in information obtained from these smart grids are addressed by NIST [2]. The problem of estimation of time between failures in electric grids leading to greater sustenance has been previously modeled in [9]. The problem is particularly related to multiple and distributed failure modes and causes with potential explosion of data. Intelligent techniques for smart grids have been explored elsewhere [20]. Various predictive models for smart grid enabling devices like smart meters are explored elsewhere [8].

In order based aggregation some of the classic methods of determining the top ranked candidates are Borda Count [17] and Ranked Pairs. These methods have a few weaknesses, such as the fact that the Borda Count method fails the Condorcet criterion and the Kemeny-Young method is NP-hard [18], although techniques using integer programming allow fast computation of the full rankings of all votes. The study of the problem of rank aggregation has received fresh impetus due to research in areas such as meta search and collaborative document filtering [6]. Correspondingly several new algorithms for performing rank aggregation have been proposed. In [10], a Markov Chain based algorithm has been proposed. In this algorithm, it has been assumed that there exists a Markov Chain on the candidates and that the order relations between entities in the ranking lists represents transitions in the corresponding Markov Chain. The final ranking is obtained by evaluating the stationary distribution of the Markov Chain.

In the following sections are presented two algorithms which contribute to the computation in distributed and P2P environments. Distributed linear regression and distributed rank aggregation are explained next.

3. PREDICTING DEMAND FOR SUSTAINABLE ENERGY MANAGEMENT

This section looks at the problem of predicting energy demand using distributed energy consumption and production data. It makes the following assumptions about the overall smart grid model of energy management:

1. With the advent of solar panels, smaller wind turbines, and geothermal technology, energy production is likely to be a household process in the near future. Residences are likely to be producing and supplying surplus energy to the community. In addition, traditional energy companies will be producing and selling energy.

2. Eventually, a smart grid will most likely be supporting multiple energy company. Therefore, the household energy consumption data may not belong to a single company. Even if it does, privacy protection of the consumer will be an important issue.

3. A household or a corporate entity may or may not want to allow centralizing data

Forecasting the demand would require building predictive models from the observed demand data and various other features such as consumption behavior, housing and household characteristics, geographical location, season and time of the day. While there exists many techniques for learning predictive models, multivariate regression is a popular well-understood technique for constructing such predictive models.

The proposed solution is abstracted on a distributed environment of the grid infrastructure. The data aggregators like smart meters can be seen as nodes in a network represented by the grid. We can construct a network of N nodes of no specific overlay topology. Each node N_i contains data tuples given by X_i^m where m is the number of features of the data. It is assumed that the data present at each node is homogeneous in nature. These data represent the information that is generated at each node by virtue of the consumption and production characteristics of the grid network. Our goal is to learn a multivariate regression model while we computing the approximate linear coefficients for the global regression model locally at each node in a distributed and asynchronous manner.

3.1 Technical Approach

The approach presented here reduces the linear regression problem into another primitive computation (computing sum) by solving the normal equation using factorization techniques. As noted earlier, the linear regression problem can be written in the form A.w = b. This can be solved using standard methods like LU decomposition and back substitution, Cholesky decomposition or Gauss-Jordan Elimination. In matrix form, this can be written as (A.T .A).w = A.T .b.

Since A.T .A is symmetric and positive-definite, Cholesky decomposition is the most efficient way to solve the normal equations. Cholesky decomposition is about a factor of two faster than alternative methods for solving linear equations. Cholesky decomposition constructs a lower triangular matrix L whose transpose L.T can itself serve as the upper triangular triangular part. In other words we replace A by A = L.L.T

This factorization is sometimes referred to as ‘taking square root’ of a matrix A, though, because of the transpose, it is not literally that. The component of A.T is solved is given by

\[ L_{i1} \]
\[ L_{i1} \rightarrow L_{21} \rightarrow L_{22} \]
\[ L_{31} \rightarrow L_{32} \rightarrow L_{33} \]
\[ \vdots \rightarrow \vdots \rightarrow \vdots \rightarrow \vdots \]

Now we need to solve the equation of the form L.L.T .w = b. Substituting p = L.T .w and solving for p in the equation L.p = b is given by p_i = \frac{1}{L_{i1}} \sum_{j=1}^{m} L_{ij} p_j , where i = 1, 2, ..., m. Finally solving for w, the regression coefficients in the equation L.T .w = p, we have w_i = \frac{1}{L_{i1}} \sum_{j=1}^{m} L_{ij} w_j , where i = m, m-1, ..., 1.

These equations can be computed using an iterative formulation. The computation of L.T using relaxation-type approaches where the value of L_{i(j+1)} at iteration (t + 1) is
The iterative approach to solve normal equations can be extended to fit computations in distributed environments. Assuming there are $v$ nodes in a network with data matrix at each node represented by $D_v$, where $m$ is the dimension of the data and $n$ is the number of data tuples present at the $v^{th}$ node, then data communication among nodes connected with each other over time to compute the regression coefficients can be perceived as an asynchronous distributed problem. All of these assumptions are based on the fact that the data is homogeneous in nature distributed at different nodes or sites. Calculations of linear co-efficients can be iteratively solved and the expressions are additive in nature so a de-centralized calculation of co-efficients are possible over large distributed environments overlayed by a communication strategy or protocols like gossip based computations. In gossip protocol, a peer of a peer-to-peer network exchanges data or statistics with a random peer. Kempe et al’s Push-Sum protocol based on gossip communication for computing sum at the nodes of a network is asymptotically optimal with respect to convergence speed. The experimental setup and some preliminary results are presented in the next subsection.

Figure 2: Simulated hourly electricity demand in the state of New York by County in a typical day of Summer.

3.2 Preliminary Experimental Results

The algorithms were implemented in Distributed Data Mining Toolkit (DDMT)[7] developed by the DIAIDC research lab at UMBC. The topological information generated by the Barabasi Albert (BA) model in BRITE [5] were used since it is often considered a reasonable model for peer-to-peer infrastructure. The data is collected from the consumption section of Residential Energy Consumption Survey (RECS)[1] 2005 which is a national area-probability sample survey that collects energy related data for occupied housing units. The data attributes included housing unit characteristics like mobile homes, single family detached house and apartment buildings etc., the number of people living in each household, the average energy consumption per house by dryer, dishwasher, refrigerator and other electric appliances. The simulated demand model over a period of one day is hosted on web service found elsewhere.

The experiments were carried out for approximately 7 million residential homes for the state of New York grouped by 62 counties. The consumption and production data were

http://geocommons.com/maps/38838

simulated on an hourly basis. Figure 2 shows the geo-spatial power energy demand that was simulated for the counties in the state of New York. In Figure 3 the comparison between communication cost in terms of bytes of data that gets transferred for the centralized approach and distributed approach is shown. As seen from the figures the x axis represents the problem size, here the number of households that are producing the data and the y axis represents the size of the message that are transferred in the network. As seen from the figure the communication cost increases linearly with increase in size for the centralized algorithm whereas communication cost increases logarithmically for the distributed algorithm making it suitable for scalable applications.

The next section presents an approach for effectively using power consumption data from smart grids in dynamic pricing models by distributed rank aggregation.

4. DISTRIBUTED RANKING OF POWER CONSUMPTION DATA

Let us consider the problem of dynamic pricing model in smart grids. A grid injection point is defined as a location in the grid that provides a hook-in for producers of electricity. Retail outlets and industrial units consume electricity via the grid exit points. Computation devices can be installed at such exit points to measure variation in power consumption. Power consumption varies over the day and providers as well as consumers would like a pricing model that reflects this variation. This issue can easily be modeled as a rank aggregation problem. The day (or any length of time) is divided into certain specific time periods that captures the variation in power consumption. Each exit point in the smart grid then uses this information to rank the time periods on the basis of their pattern of power consumption. This ranking can then be aggregated to provide a global ranking that captures the global variation in power consumption. One way of doing this is to collate all the client information in a central database and then apply conventional data mining techniques on it. However for a large number of nodes the overhead of maintaining a central repository becomes a
bottleneck. Centralized systems are also vulnerable to security threats as they provide a single point of attack for cyber criminals. A better approach would be to use a local distributed algorithm where each node communicates with only its local neighborhood. This paper outlines a distributed rank aggregation algorithm that satisfies the Condorcet criterion. The Condorcet criterion states that the Condorcet winner is the candidate who, when compared with every other candidate, is preferred by more voters. Thus the time period that, when compared to every other time period, records the highest power consumption will be ranked at the top in the global aggregate ranking.

Let us first define the problem formally. Referring to the notations in [19] and [16], it is assumed that a group $U$ of peers in a P2P network, denoted by a connected graph $G(U, E)$, would like to determine the top ranked option among a set of $n$ options in $m$ rankings. Each peer $u \in U$ contains a subset $m_u (\sum_{u \in U} m_u = m)$ of such rankings and conveys its preference by initializing a $n \times n$ ranking matrix $P^{\perp}_u$, where $n$ is the total number of options and $P^{\perp}_u$ represents the number of times candidate $i$ wins over $j$ in those $m_u$ rankings. Note that $P^{\perp}_{ij} + P^{\perp}_{ji} = m_u \forall i, j \leq n$. Our aim is to determine the Condorcet winner, the candidate that wins over all other candidates on the basis of a run-off election [21]. Each node $i$ conveys its ranking through a $n \times n$ matrix $P^{\rightarrow}_i$, where $n$ is the number of total number of candidates. $P^{\rightarrow}_i$ represents the total number of rankings in which candidate $i$ wins over candidate $j$. Ranking is assumed to totally ordered. Hence candidate $i$ wins against candidate $j$ if and only if $P^{\rightarrow}_{ij} > P^{\rightarrow}_{ji}$. Our aim is to determine the candidate that wins against all candidates in a run-off election in the global $P^{\rightarrow}$ matrix.

### 4.1 Technical Approach

The DCV or Distributed Condorcet Voting protocol ensures that every peer in the system converges towards the correct Condorcet winner through communication with nodes in its immediate neighborhood. A uni-directional spanning tree overlay is assumed for the P2P system and such a topology can be enforced using existing protocols [16]. It is again assumed that peers are informed of changes in the status of their neighboring nodes and that a failing node gracefully exits the system (fail-stop failure model). Due to the spanning tree topology there exists only one path between two nodes in the system and hence the ranking matrix on each peer is only added once ensuring the duplicate insensitivity of the protocol. The DCV protocol describes the behavior of a peer when it initializes, its data changes, it receives a message and when its neighboring nodes come online or go offline. Additionally, the algorithm also determines when a node should send a message to its neighbors. Messages contain ranking vectors of individual nodes. Each node $u$ records the last message sent to and received from each of its neighbors $v$. These messages are represented by ranking matrices $P^{\rightarrow}$ and $P^{\perp}$ respectively. Then node $u$ calculates certain quantities based on the ranking matrices that it sends and receives as well as its own local ranking matrix $P^{\perp}$. It is shown that all peers will eventually converge to the exact global ranking. Individual knowledges at each node can be represented as $K_i$ where $i = 1, \ldots, t$ for $t$ nodes in the system. We also prove that two peers agreeing on the same rankings will stop sending messages to each other. So the total knowledge of the system can be represented as $\bigcup_{i=1}^{t} K_i$.

An operator $RP$ is defined which gives the Condorcet winner of the $K_i$ that it is applied to. If $RP$ gives the same Condorcet winner for $K_i$ then it will give the same Condorcet winner for its cover i.e., $RP(K_i \cup K_j) = RP(K_i)$ or $RP(K_j)$ iff $RP(K_i) = RP(K_j)$. We also considered the convergence rate of the protocol by examining the second eigenvalue of the random walk induced Markov chain over the spanning tree and is given by $\lambda = 1 - O(\frac{1}{n})$ where $n$ is the number of nodes in the system. Using this result it can be shown that the mixing time $T_{mix}$ of the Markov chain is equal to $O(n \log n)$. Following the approach presented above, the next section offers some preliminary results.

### 4.2 Preliminary Experimental Results

To evaluate the performance of the protocol a discrete event simulator was implemented that allowed us to test our algorithm. The simulator works using a global clock in a lock-step manner. Note that the algorithm itself is not dependent on any global clock but is completely asynchronous. At each step the state of a peer is updated and messages are sent to its neighbors if required. It is assumed that each peer may only process one message at a time. The next updating of the global state occurs after each peer has finished processing all its incoming messages. Essentially the performance characteristics (communication overhead and convergence time) were compared for distributed majority voting DMV [19], distributed plurality voting DPV [16] and DCV. Three different types of networks were used which include grid, power-law graph and random-graph. Power-law and random graphs using BRITE [5] were generated.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Avg. value of $a_u$</th>
<th>Avg. time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCV</td>
<td>3.64</td>
<td>2.87</td>
</tr>
<tr>
<td>DPV</td>
<td>36.78</td>
<td>21.32</td>
</tr>
<tr>
<td>DMV</td>
<td>106.42</td>
<td>124.45</td>
</tr>
</tbody>
</table>

For evaluating our algorithm two performance parameters were used, namely, the total number of messages $a_u$ sent by a peer $u$ which we define as the communication overhead, and the time taken for a state when no messages are sent which we define as the convergence time. It can be observed that the DCV protocol is superior to the DPV and DMV protocol in finding the Condorcet winner both in terms of the communication overhead as well as the time taken. It must be kept in mind though that the messages sent by DCV and DMV contain d entries or the total number of candidates, while those sent by DMV contain only 2 entries. DCV has the added overhead of also transmitting the ordering of the d entries. Thus while DPV and DMV send a larger number of messages, the information density in a message transmitted by DCV is significantly higher.

### 5. CONCLUSIONS

In this paper we presented a look at novel techniques where distributed asynchronous algorithm can be used to learn predictive models and aggregating ranks in a highly distributed peer to peer environment which can be used over smart grids in a scalable manner. Our preliminary ex-
Experiments showed that by distributed approach we arrived at results within small degree of error for prediction and under the assumption of uniform rank distribution it has lower communication overhead as compared to alternative approaches. Also for large number of nodes running the algorithms might prove cost effective keeping in mind as the smart grid grows it would produce large volume of actionable data, has to be resilient and reactive and incorporate plug and play alternate energy producers. Such computational intelligence for predicting, shaping and optimizing the network would not only help reduce carbon footprint but also a sustainable way of life.

6. ACKNOWLEDGMENTS
This research is supported by NASA Grant NNX07AV70G and the AFOSR MURI Grant 2008-11.

7. REFERENCES
http://www.eia.doe.gov/emeu/recs/recspubuse05/pubuse05.html.