Fast Structure Learning in Generalized Stochastic Processes with Latent Factors

Mohammad Taha Bahadori
Univ. of Southern California
Los Angeles, CA 90089
mohammab@usc.edu

Yan Liu
Univ. of Southern California
Los Angeles, CA 90089
yanliu.cs@usc.edu

Eric P. Xing
Computer Science Department
Carnegie Mellon University
Pittsburgh, 15213 PA
epxing@cs.cmu.edu

ABSTRACT

Understanding and quantifying the impact of unobserved processes is one of the major challenges of analyzing multivariate time series data. In this paper, we analyze a flexible stochastic process model, the *generalized linear autoregressive process* (GLARP) and identify the conditions under which the impact of hidden variables appears as an additive term to the evolution matrix estimated with the maximum likelihood. In particular, we examine three examples, including two popular models for count data, i.e, Poisson and Conwey-Maxwell Poisson vector auto-regressive processes, and one powerful model for extreme value data, i.e., Gumbel vector auto-regressive processes. We demonstrate that the impact of hidden factors can be separated out via convex optimization in these three models. We also propose a fast greedy algorithm based on the selection of *composite atoms* in each iteration and provide a performance guarantee for it. Experiments on two synthetic datasets, one social network dataset and one climatology dataset demonstrate the the superior performance of our proposed models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Time Series Analysis

Keywords

Time Series Analysis, Latent Factors, Generalized Linear Models

1. INTRODUCTION

In many applications, an enormous amount of time series data is collected, which requires us to develop faster and more efficient algorithms for analysis and forecasting purposes. A major challenge with which we are confronted in practical applications is the incompleteness of the data, i.e., certain influential time series are missing in the real-world datasets. For example, in social media analysis, the external events influence large clusters of users, while the news propagates through the local connections in the network. In order to identify the true influence patterns among the users, we need to take into consideration the impact of external unobserved events. In climate data analysis, the local terrain characteristics play an important role in the air mass propagation while large weather systems, which are usually not observed in the dataset collected by local weather stations, influence wide areas on the ground.

The traditional approach to capture the impact of unobserved variables is to include them in the graphical models and infer their impact on the model via the EM algorithm [9]. However, this approach has two main weaknesses: (1) often times, the EM algorithm only identifies a local optimum. (2) While several techniques have been developed to speed up the EM algorithm, usually the inference cannot scale to large datasets. Recent progress shows that in the *Gaussian linear* undirected graphical [4] and vector autoregressive [13] models, the impact of hidden variables appears as an additive low rank matrix in the precision and evolution matrices, respectively. Thus, one can use scalable convex optimization algorithms to decompose the parameter matrix into a sparse local dependency and another low-rank global impact matrix which models the impact of hidden variables.

While the convex sparse plus low-rank decomposition in the linear vector auto-regressive models is promising, the model applies to a very limited class of time series data. For example, in social media applications, in which the number of mentions of key words by users is a counting process, the Gaussian linear vector auto-regressive model obviously is not applicable. In many climatology applications the distribution of the data exhibits heavy tails [6, 2]. For e.g. climate change is mostly characterized by increasing probabilities of extreme weather patterns such as temperature or precipitation reaching extremely high values [26]. In search of more general and flexible time series models, we construct several auto-regressive processes and show that the maximum likelihood estimate of their evolution matrices can be decomposed into a sparse and a low-rank matrix with the latter capturing the impact of unobserved processes. For counting processes, we analyze the Poisson [30] and Conway-Maxwell Poisson [32] auto-regressive processes. The latter distribution has recently attracted researchers' attention because of its flexibility in modeling the under-dispersion and over-dispersion of discrete data [25, 23]. For extreme value time

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

series, we propose a novel heavy-tailed auto-regressive time series model, by choosing the distribution of the data to be the Gumbel distribution.

For fast solutions, we develop a scalable greedy sparse plus low-rank decomposition algorithm for maximizing the likelihood functions of GLARP models based on the work in [27, 24]. Providing an upper bound on the convergence rate, we show that the greedy algorithms can be used for *composite atoms*, i.e., vectors that are obtained by concatenating sparse plus low-rank atoms. We also show why the single atom selection per iteration yields slower rate of convergence. To our best knowledge, the composite atoms have not been studied prior to this work. Extensive experiments on two synthetic datasets, one climatology dataset and one social network dataset are shown to demonstrate the superior performance of the proposed algorithms.

2. PRELIMINARIES AND RELATED WORK

Notation.

In this paper, we denote a single random variable with lower-case letters (for e.g. x) and a vector of random variables by bold letters (for e.g. $\mathbf{x}$). We can represent a set of N time series of length T by its elements $x_i(t)$ which represents the value of the i^{th} time series at time t. Using the notation, $\mathbf{x}(t)$ denotes the value of all time series at time t.

Generalized Linear Models.

The Generalized Linear Model [18] describes the connection between the response variables $\mathbf{y}$ and the predictor variables $\mathbf{x}$ via the following linear dependence model:

$$g(\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}]) = A\mathbf{x} + \mathbf{b}, \tag{1}$$

where the strictly monotone function $g(.)$ is called the *link function* and A and $\mathbf{b}$ are linear regression coefficients. Based on generalized linear models, we can define the stochastic process model for time series $\mathbf{x}(t)$ for $t = 1, \ldots, T$ according to the following *Generalized Linear Auto-regressive Processes* (GLARP) model:

$$g(\mathbb{E}_{\mathcal{H}(t)}[\mathbf{x}(t)]) = \sum_{\ell=1}^{K} A^{(\ell)}\mathbf{x}(t-\ell) + \mathbf{b}. \tag{2}$$

where the matrices $A^{(\ell)}$ for $\ell = 1, \ldots, K$, K denoting the maximum lag in time, are called the *Evolution Matrices* and $\mathbb{E}_{\mathcal{H}(t)}$ emphasizes the point that the expectation is performed given the history before time t. The generative process corresponding to the model above can be described as follows: at time t, compute the conditional mean of $\mathbf{x}(t)$ using the outcomes at time $t-K, \ldots, t-1$, i.e. $\mathbf{x}(t-K), \ldots, \mathbf{x}(t-1)$; then generate $\mathbf{x}(t)$ according to the computed mean. Examples of the generalized linear auto-regressive models are vector auto-regressive models that are widely used for jointly modeling multiple continuous time series and Poisson auto-regressive processes for modeling multiple time series of count data.

We build the temporal dependency graph $G(V, E)$ corresponding to the evolution matrices $A^{(\ell)}$ for $\ell = 1, \ldots, K$ by representing every time series x_i by a node $v_i \in V$. We add a *directed* edge $e_{i \to j}$ to the set E if at least one of the entries $A^{(\ell)}_{j,i}$ for $\ell = 1, \ldots, K$ is non-zero.

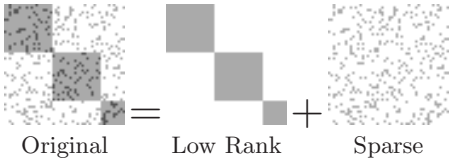


Figure 1: Decomposition of the evolution matrix in Eq. (5) into low-rank and sparse matrices.

Sparse plus Low-rank Decomposition.

In order to achieve a consistent estimate of a high dimensional matrix from a limited number of observations, we are required to impose a low-dimensional structure on the estimated matrix. One of the most popular structures is the sparse plus low-rank structure which assumes that the true value of the matrix is approximately equal to a low-rank part plus a sparse part (Fig. 1). Examples of the applications that exhibit this low-dimensional structure are Robust PCA [5, 3, 20], Robust covariance estimation [1] and Multi-task regression [17, 21].

Learning with Hidden Factors.

In many real world applications, observing all influential quantities can be expensive or even not possible. The hidden time series can be the quantities that are hard to measure or have corrupted measurements; they can also represent immeasurable events such as disease outbreak news and its impact on social networks. Thus, taking into consideration the possible existence of a few hidden variables in the analysis makes the analysis significantly more accurate and realistic. The most common approach to capture the effect of hidden variables is based on the EM algorithm [9]. While the EM framework is quite general, it suffers from getting trapped into the local optima. In this work we are interested in finding a convex programming solution which does not depend on the initialization point.

In many real world datasets there are unobserved variables that impact large groups of observed variables; this phenomenon is called the *global influence*. Examples of this phenomenon include the global impact of airwaves in climatology and the network-wide impact of external news on social networks. At the same time, the observed variables have *local sparse connectivity* with each other. Examples of the local dependency are the users in social networks who share their friends' posts or influence of a region on to another one due to their spatial proximity. [4] shows that in undirected graphs with unobserved variables with global impact, the precision matrix of the joint distribution of observed variables have the sparse plus low-rank structure. [13] shows that in the vector auto-regressive model with unobserved variables with global impact, the evolution matrix estimated via maximizing the likelihood of the observed data only, will result in the sparse plus low-rank structure as well.

3. METHODOLOGY

In this section, after describing the generalized linear auto-regressive processes with latent factors, we introduce and analyze two GLARP models for modeling count data and another one for modeling extreme value time series. Theorem 3.1 shows that in these models, the maximum likelihood estimate of the evolution matrix can be decomposed into a sparse and a low-rank matrix with the latter capturing the impact of unobserved processes. Then, in Section 3.2 we

propose an algorithm to uncover the true evolution matrix and guarantee its convergence to the global optimum of the objective function (Theorem 3.2).

Consider the following model for generalized linear autoregressive processes with hidden factors:

$$g\left(\mathbb{E}_{\mathcal{H}(t)}\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{z}(t)\end{bmatrix}\right) = \sum_{\ell=1}^{K}\begin{bmatrix}A^{(\ell)} & B^{(\ell)}\\ C^{(\ell)} & D^{(\ell)}\end{bmatrix}\begin{bmatrix}\mathbf{x}(t-\ell)\\ \mathbf{z}(t-\ell)\end{bmatrix} + \mathbf{b} \quad (3)$$

for $t = K+1, \ldots, T$, where $\mathbf{x}(t)$, a $p \times 1$ vector, represents the observed variables, $\mathbf{z}(t)$, a $r \times 1$ vector, denotes the unobserved variables and the function g is the link function. The density function of the observations at time t is denoted by $f(\mathbf{x}(t), \boldsymbol{\theta}(t))$ where $\boldsymbol{\theta}(t)$ denotes the set of parameters of the distribution that are functions of the evolution matrices $A^{(\ell)}, B^{(\ell)}, C^{(\ell)}$ and $D^{(\ell)}$ and the past values of time series $\mathbf{x}(t)$ and $\mathbf{z}(t)$.

The maximum likelihood estimation of the model parameters in absence of the time series $\mathbf{z}(t)$ is performed as follows:

$$\{\widehat{A}^{(\ell)}\}_{MLE} = \arg\max_{\{\widehat{A}^{(\ell)}\}}\left\{\prod_{t=K+1}^{T} f(\mathbf{x}(t), \boldsymbol{\theta}(t))\right\}, \quad (4)$$

where $\{\widehat{A}^{(\ell)}\}$ represents the set of evolution matrices $A^{(\ell)}$ for $\ell = 1, \ldots, K$.

3.1 Examples of GLARP

In this section, we define three time series models for two applications: (i) count data obtained from binning of point processes in social networks and (ii) heavy-tailed continuous data in the climate applications. In all of these models the hidden variables create the sparse plus low-rank structure in the evolution matrix.

3.1.1 Count Data

Recently, point processes have been successfully applied to social networks analysis [31, 16, 15]. A popular approach in analysis of temporal dependency among multiple point processes is to count the number of events in regularly spaced intervals and analyze the resulting count time series [30, 15].

The Poisson distribution is one of the most commonly used distributions for modeling count data. According to the Poisson autoregressive point process model [30], the distribution of variables at time t is a Poisson distribution with a rate *conditioned on the history* modeled as follows:

$$\log \boldsymbol{\lambda}(t) = \log(\mathbb{E}_{\mathcal{H}(t)}[\mathbf{x}(t)]) = \sum_{\ell=1}^{K} A^{(\ell)}\mathbf{x}(t-\ell) + \mathbf{b}, \quad (5)$$

where $\boldsymbol{\lambda}(t)$ represents the rate parameter for the Poisson distribution. The negative log-likelihood function for this model is convex and can be efficiently minimized.

Conwey-Maxwell Poisson Distribution.

An important limitation of the Poisson regression is that the variance of a Poisson distributed variable is equal to its mean, i.e., the Poisson model does not allow *over-dispersion* and *under-dispersion* which describe variances above and below the mean, respectively. The Conwey-Maxwell Poisson distribution (in short COM-Poisson) is a two-parameter extension of the Poisson distribution with a parameter for modeling the dispersion. Historically, it was introduced in [8] and recently studied comprehensively in [25]. The COM-Poisson distribution is defined based on the following property:

$$\frac{\mathbb{P}[X = k-1]}{\mathbb{P}[X = k]} = \left(\frac{k}{\mu}\right)^{\nu},$$

where ν is called the dispersion parameter, and $\nu < 1$ modeling over dispersion and $\nu > 1$ modeling underdispersion. The main advantage of the COM-Poisson distribution over other generalizations of the Poisson distribution, such as Double Poisson [10] and Generalized Poisson [7] distributions, is its flexibility in modeling a greater range of dispersion [32]. The COM-Poisson distribution is equivalent to the Poisson distribution when $\nu = 1$, the Geometric distribution when $\nu = 0$ and the Bernoulli distribution as $\nu \to \infty$. The COM-Poisson GLARP is defined as follows [32]:

$$\mathbb{P}[x_i(t)|\mu_i(t), \nu] = \frac{1}{S(\mu_i(t), \nu)}\left(\frac{\mu_i(t)^{x_i(t)}}{x_i(t)!}\right)^{\nu}$$

$$\log\left(\boldsymbol{\mu}(t) + \frac{1}{2\nu} - \frac{1}{2}\right) \approx \log\left(\mathbb{E}_{\mathcal{H}(t)}[\mathbf{x}(t)]\right) = \sum_{\ell=1}^{K} A^{(\ell)}\mathbf{x}(t-\ell) + \mathbf{b}. \quad (6)$$

where $\boldsymbol{\mu}(t)$ is the rate parameter and $S(\mu_i(t), \nu)$ is the normalization term. Given a constant (invariant with time) value for the dispersion parameter ν, the negative log-likelihood function is convex and can be minimized efficiently.

3.1.2 Extreme value data

In many applications, such as climate analysis, time series data usually exhibit a heavy-tailed distribution which is significantly different from the commonly assumed Gaussian distribution. The generalized extreme value theorem states that the maximum of a set of independently and identically distributed random variables asymptotically converges to the Extreme Value Distribution, [6, 2]. Hence, the Generalized Extreme Value distribution and its special case, the Gumbel distribution, are the distributions of choice for modeling the extreme value data. In this paper, we define a Gumbel GLARP model as follows:

$$f(x_i(t)|\mu_i(t), \sigma) = \frac{1}{\sigma}\exp\left(-\frac{x_i(t)-\mu_i(t)}{\sigma} - \exp\left(-\frac{x_i(t)-\mu_i(t)}{\sigma}\right)\right) \quad (7)$$

$$\boldsymbol{\mu}(t) + \sigma\gamma_E = \mathbb{E}_{\mathcal{H}(t)}[\mathbf{x}(t)] = \sum_{\ell=1}^{K} A^{(\ell)}\mathbf{x}(t-\ell) + \mathbf{b},$$

where $\boldsymbol{\mu}(t)$ and σ denote the location and scale parameters of the Gumbel distribution and $\gamma_E \approx 0.5771$ is the Euler constant. Given a constant scale parameter σ, the negative log-likelihood function is convex and can be minimized efficiently. Note that there are other ways to define a Gumbel autoregressive process, [29], however the above novel model is defined to have the sparse and low rank decomposition property for hidden variables.

For all of the GLARP models described above, we have the following theorem:

THEOREM 3.1. *Suppose a generalized linear auto-regressive process $(\mathbf{x}(t), \mathbf{z}(t))$ is defined according to Eq. (5), Eq. (6) and Eq. (7). Suppose the number of unobserved processes r and number of lags K are much smaller than the number of observed ones, i.e. $r, K \ll p$. Then, asymptotically as*

$T \to \infty$, the maximum likelihood estimate of $\{A^{(\ell)}\}$ is sum of two matrices:

$$\lim_{T\to\infty} \widehat{A}^{(\ell)}_{MLE,T} = A^{(\ell)} + L^{(\ell)},$$

where $L^{(\ell)}$ a low-rank matrix with $rank(L^{(\ell)}) \leq r.K$.

Proof sketch.

The solution relies on two main ideas:

1) Asymptotically, the maximum likelihood estimation procedure is equivalent to minimization of the KL-distance between the true model and the observed model. We can write:

$$\widehat{A}_{MLE} = \arg\min_{\widehat{A}} \left\{ \mathbb{E}_{\text{True}} \left[\mathcal{L}_{\text{True}}(\mathbf{x}(t)) - \mathcal{L}_{\text{Obs}}(\mathbf{x}(t)) \right] \right\}, \quad (8)$$

$$= \arg\min_{\widehat{A}} \left\{ \mathbb{E}_{\text{True}} \left[-\mathcal{L}_{\text{Obs}}(\mathbf{x}(t)) \right] \right\} \quad (9)$$

where $\mathcal{L}_{\text{True}}$ and $\mathcal{L}_{\text{Obs}}$ denote the log-likelihood of the true and observed models, respectively.

2) For point processes, suppose we divide the time into small intervals such that the probability of observing more than one event in each interval is small. We can approximate the likelihood of the observed time series for any point process in a unified form given its rate function, as shown in [30]. This allows the computation of $\widehat{A}_{MLE}$ for all point processes in a unified way.

The details of the proof is provided in the Appendix.

3.2 Inference

Using the result of Theorem 3.1 we need to solve the following optimization algorithm to capture the effect of unobserved variables:

$$\min_{A^{(\ell)},L^{(\ell)},\mathbf{b}} \mathcal{L}\left(\mathbf{x}(t), A^{(\ell)}, L^{(\ell)}\right)_{t=1:T}^{\ell=1:K} \quad (10)$$

$$\text{Subject to:} \quad \sum_{\ell=1}^{K} \left\| A^{(\ell)} \right\|_0 \leq \eta_S, \quad \sum_{\ell=1}^{K} \text{rank}\left(L^{(\ell)}\right) \leq \eta_L,$$

where the L_0 norm of the matrices is equal to the number of non-zeros elements of the matrices and $\mathcal{L}$ denotes the likelihood of the stochastic process defined in Eq. (3). There are two main approaches to solve the problem in Eq. (10). The first approach uses a convex relaxation of the L_0 norm with the L_1 norm and the rank constraint with the nuclear norm L_*:

$$\min_{A^{(\ell)},L^{(\ell)},\mathbf{b}} \left\{ \mathcal{L}\left(\mathbf{x}(t), A^{(\ell)}, L^{(\ell)}\right)_{t=1:T}^{\ell=1:K} + \lambda_S \sum_{\ell=1}^{K} \left\| A^{(\ell)} \right\|_1 + \lambda_L \sum_{\ell=1}^{K} \left\| L^{(\ell)} \right\|_* \right\} \quad (11)$$

The optimization problem in Eq. (11) is convex and can be solved via Singular Value Thresholding (SVT) in each iteration of the Accelerated Proximal Gradient algorithm [19] as described in [28]. The second approach is to combine the greedy sparse and greedy low rank [12, 24] matrix learning algorithms in the unified framework provided by [27]. The greedy approach does not rely on the L_* and L_1 heuristics and directly solves Eq. (10); i.e. it iteratively constructs the optimal sparse and low rank matrices along the sparse and low-rank directions. The greedy low-rank learning has been shown to be faster and more scalable than the SVT approach [12, 24]; hence, we develop Algorithm 1 in the greedy framework.

In Algorithm 1, for notation simplicity, we show the parameters in the sparse and low rank matrices by $\mathbf{w} \in \mathbb{R}^{(2K+1)p^2 \times 1}$

Algorithm 1: Greedy Sparse plus Low-Rank Decomposition

Input: $\{\mathbf{x}(t)\}_{t=1,\ldots,T}$, η_S, η_L

Let $\mathbf{w}$ denote concatenation of $L^{(\ell)}$, $A^{(\ell)}$ and $\mathbf{b}$.
Initialize $\mathbf{w}_1 \leftarrow \mathbf{0}$.
for $\tau \leftarrow 1, 2, 3, \ldots$ **do**
  $\mathbf{a}_t^{(L)} \leftarrow \arg\min_{\mathbf{a}\in\mathcal{A}^{(L)}} \left\langle \nabla\mathcal{L}(\mathbf{w}_t), \mathbf{a}^{(L)} \right\rangle$.
  $\mathbf{a}_t^{(S)} \leftarrow \arg\min_{\mathbf{a}\in\mathcal{A}^{(S)}} \left\langle \nabla\mathcal{L}(\mathbf{w}_t), \mathbf{a}^{(S)} \right\rangle$.
  $\alpha_t, \beta_t, \mathbf{b}_t \leftarrow \arg\min_{\alpha,\beta\in[0,1],\mathbf{b}}$
  $\mathcal{L}(\mathbf{w}_t + \alpha(\eta_S \mathbf{a}_t^{(S)} - \mathbf{w}_t^{(S)}) + \beta(\eta_L \mathbf{a}_t^{(L)} - \mathbf{w}_t^{(L)}))$.
  $\mathbf{w}_{t+1}^{(S,L)} \leftarrow$
  $\mathbf{w}_t^{(S,L)} + \alpha_t(\eta_S \mathbf{a}_t^{(S)} - \mathbf{w}_t^{(S)}) + \beta_t(\eta_L \mathbf{a}_t^{(L)} - \mathbf{w}_t^{(L)})$.
end
return $L^{(\ell)}, A^{(\ell)}$, *for* $\ell = 1, \ldots, K$.

where the its first Kp^2 elements $\mathbf{w}^{(S)}$ contain the elements of $A^{(\ell)}$, the second Kp^2 elements $\mathbf{w}^{(L)}$ contain the elements of $L^{(\ell)}$ for $\ell = 1, \ldots, K$ and the last p elements contain $\mathbf{b}$. The algorithm iteratively selects the *atoms* from two sets of atoms: (1) $2Kp^2$ sparse atoms which are created by placing ± 1 in place of first Kp^2 elements of $\mathbf{a}$. This takes $\mathcal{O}(p^2)$ operations. (2) The low-rank atom in the atom identification step can be found via singular value decomposition, as described in [24, 27]. In fact, we only need to find an approximate leading singular vector which can be done in $\mathcal{O}(N_s \log(p))$ where N_s is the number of non-zero elements of the gradient matrix [24]. We update $\mathbf{b}$ after addition of each composite atom.

Following the framework in [27], we can derive the following convergence guarantee for Algorithm 1:

THEOREM 3.2. *The solution of Algorithm 1 at n^{th} iteration is bounded towards the optimal solution $\mathbf{w}^\star$ according to the following equation:*

$$\mathcal{L}(\mathbf{w}_n) - \mathcal{L}(\mathbf{w}^\star) \leq \frac{\mathfrak{B}_S + \mathfrak{B}_L + \mathfrak{B}_b}{n} \quad (12)$$

where the bound constant for the sparsity atom is defined as $\mathfrak{B}_S \triangleq 8L_{||.||}(\mathcal{L})\eta_S^2||\mathcal{A}_S||^2$ in which $L_{||.||}(\mathcal{L})$ is the smoothness constant of the likelihood function as defined in [27] and $||\mathcal{A}_S||^2 = \sup_{\mathbf{a}\in\mathcal{A}_S} \|\mathbf{a}^{(S)}\|$ where $\mathcal{A}_S$ denotes the set of sparse atoms. The bound term for the low-rank atoms $\mathfrak{B}_L$ and $\mathfrak{B}_b$ are defined similarly.

A formal proof is given in the Section 5. Note that the solution always stays inside the constraints, thus the optimization algorithm does not have to deal with the non-differentiability of the Lagrangian in the constraint boundaries. Further analysis in the Appendix shows that similar performance bound for the algorithm that selects only one atom per iteration is larger than the bound in Eq. (12) at least by the ratio of the Lipschitz constant and the restricted smoothness constant of the likelihood function. As discussed in [27], the difference can be very large; hence, the speed up due to composite atom selection can be large, as well.

4. EXPERIMENTS

In this section, we study two types of data (1) point process, including a synthetic dataset and a social networks dataset and (2) heavy-tailed data including a climate science dataset.

4.1 Datasets

Synthetic Datasets.

We created a synthetic dataset according to the Poisson autoregressive point process model in Eq. (3) to study the accuracy of the algorithms in recovering the true underlying temporal dependency graph in the presence of hidden variables. We fix the number of observed variables at 60 and vary the number of hidden variables from $r = 1$ to 5. We also varied the length of observed time series to study the asymptotic behavior of the algorithms. For generation of time series, only one unit of time lag, $K = 1$, is used. The elements of the A matrix in Eq. (5) for the point processes are generated at random, and we choose a sufficiently large negative value for $\mathbf{b}$ to stabilize the time series. The global impact of the hidden variables is modeled in the datasets by setting an edge from the hidden variables to all other observed variables. We generate 10 random datasets of each type and report the average performance on them. Due to space limit, we only report the results on the Poisson point process synthetic datasets.

Social Networking Dataset.

We used a *complete* Twitter dataset to analyze the tweets about "Haiti earthquake" by applying different temporal dependency analysis methods to identify the potential top influencer on this topic (i.e. those Twitter accounts with the highest number of effect to the others). We divided the 17 days after the Haiti Earthquake on Jan. 12, 2010 into 1000 intervals and generated a multivariate time series dataset by counting the number of tweets on this topic for the top 1000 users who tweeted most about it. The resulting time series have on average 0.0225 tweets per user per bin which shows how infrequent the events in the dataset are. For accurate modeling, we removed the users that were highly correlated with each other, most of which were operated by the same users and tweeted exactly the same contents. We also removed robot-like user-accounts who tweeted on very regular intervals, which led to a subset of 100 users.

Wind Speed.

The study of extreme value of wind speed and gust speed is of great interest to the climate scientists and wind power engineers. A collection of wind observations is provided by AWS Convergence Technologies, Inc. of Germantown, MD. It consists of the observations of surface wind speed (mph) and gust speed (mph) every five minutes. We choose 153 weather stations located on a grid laying in the $35N - 50N$ and $70W - 90W$ block. Following the standard practice in this domain, we generated extreme value time series observations, i.e, daily maximum values, at different weather stations. The objective is to examine how the global weather systems impact the local influence patterns at different locations and how well we can make predictions on future precipitation.

4.2 Evaluation Measures

For the synthetic datasets, since we have access to the underlying graph structure we can report the graph learning accuracy. We choose the Area Under the Curve (AUC) accuracy measure as it is a good performance measure for the datasets with unbalanced ratio of positive and negative labels. The value of AUC is the probability that the algorithm assigns a higher value to a randomly chosen positive (existing) edge than a randomly chosen negative (non-existing) edge in the graph. Since we don't have the true underlying influence graph in the wind speed dataset, we only report the prediction accuracy and the visualization of the results. In all the experiments, we tune the penalization parameters via 5 fold cross-validation.

Table 1: The baselines used in evaluations.

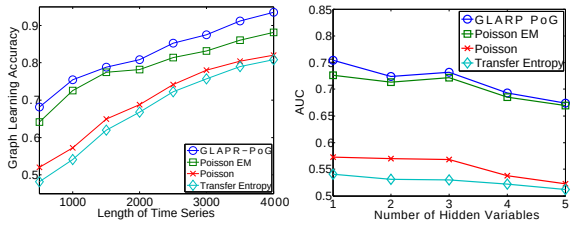
Twitter Dataset	
Algorithm	Description
GLARP-PoG	GLARP with Poisson distribution and Algorithm 1.
Poisson-EM	GLARP with Poisson distribution and EM algorithm inference.
Poisson	GLARP with Poisson distribution without hidden variables.
GLARP-COMG	GLARP with COM-Poisson distribution and Algorithm 1.
COM-P EM	GLARP with COM-Poisson distribution and EM algorithm inference.
COM-P	GLARP with COM-Poisson distribution without hidden variables.
Transfer Entropy	Transfer Entropy, a non-parametric dependency analysis algorithm [22]
Wind Speed Dataset	
Algorithm	Description
GLARP-GumG	GLARP with Gumbel distribution and Algorithm 1.
Gumbel-EM	GLARP with Gumbel distribution and EM algorithm inference.
Gumbel	GLARP with Gumbel distribution without hidden variables.
Gaussian VAR	Gaussian VAR with hidden variables.
Transfer Entropy	Continuous Transfer Entropy [14]

Since we do not have access to the true underlying influence graph in the social networking, we use the retweet network as the ground truth. The retweet network $G_{RT}(n)$ is constructed by adding an edge from user i to user j if user j has retweeted at least n of the tweets of user i, where n is varied from 1 to 5. Clearly, the retweet network is not the actual underlying temporal dependency graph, mainly because there are possible implicit influence patterns as well. However, it is the best possible metric that we could obtain for graph learning accuracy evaluation in our dataset. The retweet network for the 100 selected users is sparse. For e.g., $G_{RT}(1)$ has only 279 out of 10,000 possible edges.

For predictive analysis, in all the datasets, we split them into the training/testing parts with ratio 9/1 based on time and report the root mean square error (RMSE) and normalized RMSE on the test set. In particular, we trained the models with the observations between $t = 1, \ldots, \frac{9}{10}T$ and predicted the observations at $t' = \frac{9}{10}T, \ldots, T$ using K past observations at $t' - K, t' - K + 1, \ldots, t' - 1$. In other words, we evaluate the 1-step prediction performance of the algorithms. We reported the average RMS error on the test samples. The predictive analysis is plausible in our Twitter dataset because it is the full dump of the twitter messages, not a sub-sampled version of it.

Baselines.

To compare the performance of sparse plus low-rank decomposition, we use several state-of-art baselines (see Table 1 for details). Specifically, the EM algorithm solutions use



(a) (b)

Figure 2: Synthetic dataset results on the point process dataset (a) Graph learning accuracy as the length of the time series increases. (b) Graph learning accuracy as the number of hidden variables increases.

the EM algorithm to learn the parameters of the GLARP model in Eq. (3). The parameters in the EM algorithm are initialized to zeros. Transfer Entropy [22, 14] algorithms perform pairwise temporal dependency analysis among time series by measuring the amount of uncertainty resolved in the future of a time series by knowing the past values another time series, given its own past values.

4.3 Experiment Results

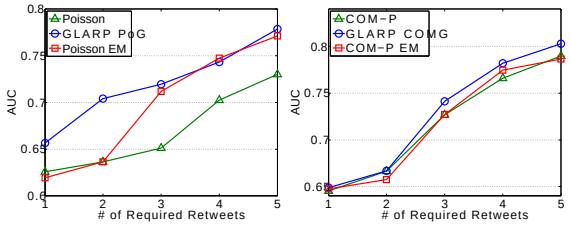
Synthetic Datasets.

The results on the synthetic datasets are shown in Fig. 2. In first set of experiments, we have only one hidden variable and vary the length of time series to measure the graph learning accuracy of the algorithms. As we expect, the performance of the algorithms uniformly increases with the length of the time series. The algorithms which capture the impact of hidden variables outperform the other algorithms by a large margin. Among the hidden variable detection algorithms, the superior performance of our proposed algorithms is because they are convex programming; while the EM-based algorithms can be stuck in some suboptimal local optima. The performance of Transfer Entropy is only comparable to the Poisson process, in Fig. 2, and with large number of samples its performance approaches to the point process.

In the second set of experiments, we fix the time series length at 500 and vary the number of hidden variables. The performance of our algorithms slightly drop, mainly because as we increase the number of hidden variables, the rank of the low-rank matrix L increases and it becomes harder to estimate [24]. With five hidden variables in Fig. 2(b), it reaches to the performance of the EM algorithm which does not rely on the $r \ll p$ assumption.The performance of Transfer Entropy and Poisson degrade too, since the true underlying model deviates more from their assumption about existence of no hidden variables.

Twitter Dataset.

As shown in Fig. 3, the performance of all the algorithms increase as we increase the number of retweets requirement n for the ground truth influence graph $G_{RT}(n)$ (defined in Section 4.2). This means all the algorithms detect the strong influence edges with higher accuracy. In all of the COM-Poisson auto-regressive models, we have set the dispersion parameter ν to a fixed large number to model the large underdispersion in the twitter time series. Capturing underdispersion in the data, all the COM-Poisson based models outperform their Poisson counterparts. As we expected the GLARP-COMG algorithm outperforms the EM counterpart



(a) (b)

Figure 3: The graph learning accuracy when the number of retweets requirement n for the ground truth influence graph $G_{RT}(n)$ is varied. The performance of (a) Poisson and (b) COM-Poisson autoregressive processes confirms that they make better predictions for the stronger influence edges.

Table 2: The RMS prediction error of the algorithms in the Twitter dataset. Results have been normalized by the the mean.

Method	RMSE	Norm-RMSE
GLARP-COMG	0.0059	0.3014
COM-P EM	0.0113	0.5739
COM-P	0.0096	0.4876
GLARP-PoG	0.0017	0.0887
Poisson EM	0.0062	0.3148
Poisson	0.0017	0.0847
Transfer Entropy	0.0030	0.1519

by avoiding the local minima. The prediction performance in Table 2 confirms this trend as well. The inferior performance of the EM algorithm is due to propagation of error; in other words, EM first infers the values of past hidden variables (accruing some error) and then uses them to predict observed time series. The lower prediction performance of COM-Poisson based algorithms is due to the approximation error in estimation of the mean $\mathbb{E}_{\mathcal{H}(t)}[\mathbf{x}(t)]$ in Eq. (6).

The transfer entropy results are (0.5427, 0.5915, 0.5924, 0.5785, 0.5442) for $n = 1, \ldots, 5$. In order to keep the resolution of the graph high, they are not shown in the graph because they were far below the rest of the algorithms. The poor performance of Transfer Entropy can be attributed to the extreme sparsity of the Twitter time series and the fact that, unlike the rest of the parametric algorithms, it does not have any procedure to benefit from sparsity of the underlying data generation model. To evaluate the prediction performance of Transfer Entropy, we used the graph estimated by Transfer Entropy in the Poisson auto-regressive process and measured its prediction performance.

In order to evaluate the speedup of using sparse plus low rank decomposition over the EM solution, we recorded the run time on the Twitter dataset on an i7 2.67 GHz laptop running Windows. The Poisson and GLARP-PoG spent 48 and 98 seconds while each iteration of the EM algorithm took 928 seconds. Given 5 iterations of the EM algorithm, the speedup by sparse plus low-rank decomposition is near 47 fold.

We next examine whether any meaningful hidden homophily can be detected by GLARP-COMG. Using the result in Eq. (28) and because we have identified only one hidden process, we can see that the summation of the $B^{(\ell)}$ matrices for $\ell = 1, \ldots, K$ should be proportional to the value in the left hand side. In other words, we can find the *average* impact of the hidden processes on the observed ones by the $\left(\sum_{\ell=1}^{K} \widehat{L}(\ell)\right) \widehat{\boldsymbol{\lambda}}_{\mathbf{x}}$. Figure 4 shows the results of the

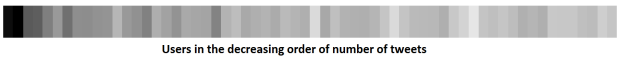
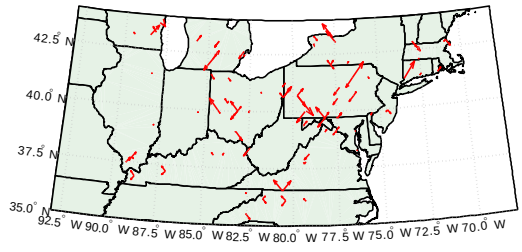
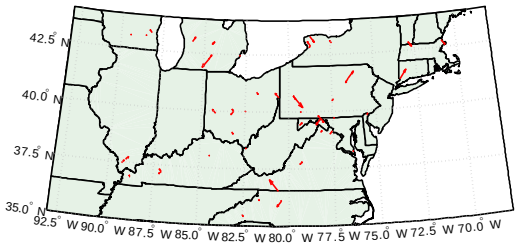


Figure 4: The hidden structure identified by GLARP-COMG approach from the Haiti Dataset. In the Haiti dataset, a single hidden variable is identified by our method. The matrix represents $B^\top$ in Eq. (3) which corresponds to the effects of the hidden variable on the input users; the darker the color, the larger the influence of the hidden variable on the user.



(a)



(b)

Figure 5: (a) The spatial-temporal dependency graph obtained via the Gumbel auto-regressive process. Note the denseness of the graph. (b) The sparse part of the spatial-temporal dependency graph obtained via GLARP-GumG. Removing the low rank global effect leaves only two main local terrain impacts: one is the local impact of the Appalachian mountains along the east coast and the other one is the local impact of the Great Lakes on the weather pattern of their surrounding lands.

GLARP-COMG method on the Haiti dataset. An immediate observation is that the hidden variables mostly impact the users on the left side of the matrix, which corresponds to those Twitter accounts with more tweets. This is reasonable since the users who are more concerned about the topic will get key information more from external news sources, such as TV, radio or personal communications, which act as hidden external variables in the model. When we zoom into the group of users affected by the hidden variable, we can see many of them are organizations or persons with possible close connections to the authority of Haiti, such as `missionmanna` (Mission Manna provides medical care for malnourished children and continuing health care education for adults in and around Montrouis, Haiti), `haitiinfocus` (HCN provides a safe facility in Thomassin Haiti where Haitian students can go to school online) and `pierrecote` (Realtime transmedia strategist, producer, director, writer and advisor to the Prime Minister of Haiti).

Wind Speed Dataset.

The prediction performance of the algorithms is listed in Table 3. The results show that the GLARP-GumG out performs the rest of the algorithms. Two patterns are different in this dataset: first the EM algorithm has lower performance than the simple Gumbel VAR algorithm. The second observation is that due to short length of time series, the Transfer Entropy faces the high dimensionality problem and cannot perform better than the Gaussian model. To evaluate the prediction performance of Transfer Entropy, we used the graph estimated by Transfer Entropy in the Gaussian auto-regressive process and measured its prediction performance.

Table 3: The RMS prediction error of the algorithms in the wind speed dataset.

Method	RMSE	Norm-RMSE
GLARP-GumG	**0.3147**	**0.0349**
Gumbel EM	0.4789	0.0531
Gumbel VAR	0.3233	0.0358
Gaussian VAR	0.8510	0.0943
Transfer Entropy	0.8871	0.0983

The GLARP-GumG algorithm detects only one hidden variable in the wind speed dataset. The impact of the detected hidden variable can be seen in Fig. 5(a) and 5(b) which show the spatial-temporal dependency graph obtained via the Gumbel auto-regressive process and the sparse part of the spatial-temporal dependency graph obtained via GLARP GumP, respectively. Comparing the two graphs, we observe that GLARP-GumG removes the main global weather impact in this season which can be attributed to the summer weather system in the region. Two main local influence patterns are detected by our algorithm: (i) the impact of the Appalachian mountains in the stretch of east coast and (ii) the local impact of the Great Lakes on the weather pattern of their surrounding lands.

5. CONCLUSION

In this paper, we studied three instances of the generalized linear autoregressive processes (GLARPs), in which the impact of hidden variables in time series data appears as an additive low-rank matrix in the maximum likelihood estimation of the evolution matrices. We demonstrated that the convex programming solution solution indeed yields better prediction and graph learning accuracy than the alternative EM-based algorithms, and our model is fast enough for large-scale applications. For future work, we are interested in generalization of the framework and establishing the statistical guarantees.

Acknowledgment

This research was supported by the NSF research grants IIS-1134990, IIS- 1254206 and the U.S. Defense Advanced Research Projects Agency (DARPA) under Social Media in Strategic Communication (SMISC) program, Agreement Number W911NF-12-1-0034. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency.

6. REFERENCES

[1] A. Agarwal, S. Negahban, and M. J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *Ann. Stat.*, 2012.

[2] J. Beirlant, Y. Goegebeur, J. Segers, and J. Teugels. *Statistics of Extremes: Theory and Applications*. Wiley, 2004.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. of the ACM*, 2011.

[4] V. Chandrasekaran, P A. Parrilo, and A S. Willsky. Latent Variable Graphical Model Selection via Convex Optimization. *Ann. Statist.*, 2012.

[5] V. Chandrasekaran, S. Sanghavi, P A. Parrilo, and A S. Willsky. Rank-Sparsity Incoherence for Matrix Decomposition. *SIAM Journal on Optimization*, 2011.
[6] S. Coles. *An introduction to statistical modeling of extreme values*. Springer-Verlag London Ltd., 2001.
[7] P. C. Consul and G. C. Jain. A generalization of the poisson distribution. *Technometrics*, 1973.
[8] R.W. Conway and W.L. Maxwell. A queuing model with state dependent service rates. *Journal of Industrial Engineering*, 1962.
[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. Series B*, 1977.
[10] B. Efron. Double exponential families and their use in generalized linear regression. *JASA*, 1986.
[11] G R. Grimmett and D R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.
[12] M. Jaggi and M. Sulovsky. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.
[13] A. Jalali and S. Sanghavi. Learning the Dependence Graph of Time Series with Latent Factors. In *ICML*, 2011.
[14] A. Kaiser. Information transfer in continuous processes. *Physica D*, 2002.
[15] G. Kim, F-F. Li, and E. P. Xing. Web Image Prediction Using Multivariate Point Processes. In *KDD*, 2012.
[16] A. Myers, S, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*, 2012.
[17] S. Negahban and M. J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. In *ICML*, 2010.
[18] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *J. R. Statist. Soc. A*, 1972.
[19] Y. Nesterov. Gradient methods for minimizing composite objective function. Core discussion papers, Universite catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.
[20] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 2010.
[21] A. Rohde and A. B. Tsybakov. Estimation of high-dimensional low-rank matrices. *Ann. Stat.*, 2011.
[22] T. Schreiber. Measuring Information Transfer. *Physical Review Letters*, 2000.
[23] K F. Sellers and G. Shmueli. A flexible regression model for count data. *Ann. Appl. Stat.*, 2010.
[24] S. Shalev-Shwartz, A. Gonen, and O. Shamir. Large-scale convex minimization with a low-rank constraint. In *ICML*, 2011.
[25] G. Shmueli, T P. Minka, J B. Kadane, S. Borle, and P. Boatwright. A useful distribution for fitting discrete data: Revival of the conway-maxwell-poisson distribution. *J. R. Stat. Soc. S. C*, 2005.
[26] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor, and H.L. Miller, editors. *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2007.
[27] A. Tewari, P D. Ravikumar, and I S. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In *NIPS*, 2011.
[28] KC. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific J. Optim*, 2010.
[29] G. Toulemonde, A. Guillou, P. Naveau, M. Vrac, and F. Chevallier. Autoregressive models for maxima and their applications to CH4 and N2O. *Environmetrics*, 2010.
[30] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 2005.
[31] D. Q. Vu, A. U. Asuncion, D. R. Hunter, and P. Smyth. Continuous-Time Regression Models for Longitudinal Networks. In *NIPS*, 2011.
[32] F. Zhu. Modeling time series of counts with com-poisson ingarch models. *Math Comput Model*, 2012.

Appendix

Proof of Theorem 3.1

Without loss of generality, we prove the case where $K = 1$ and $\mathbf{b} = \mathbf{0}$. The proof for the general case is straightforward, but algebraically more involved, extension of the simpler case.

Proof for the Poisson and COM-Poisson GLARPs

Consider the following *true* model for the time series:

$$\log\left(\mathbb{E}\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{z}(t)\end{bmatrix}\right) = \begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}\mathbf{x}(t-1)\\ \mathbf{z}(t-1)\end{bmatrix} \tag{13}$$

Let $\mathbb{E}[\mathbf{x}(t)] = \boldsymbol{\lambda}(t), \mathbb{E}[\mathbf{z}(t)] = \boldsymbol{\lambda}'(t)$ and $\mathbf{u}(t) = [\mathbf{x}(t)^\top, \mathbf{z}(t)^\top]^\top$ denote the aggregation of the both observed and unobserved variables. In the maximum likelihood solution with unobserved time series $\mathbf{z}(t)$, we fit the data to the following *observed* model:

$$\log\left(\mathbb{E}[\mathbf{x}(t)]\right) = \widehat{A}\mathbf{x}(t-1) \tag{14}$$

First we show how we can derive the likelihood for any point process given its rate function, [30]. Suppose we have divided the time into small enough so that the probability of $x_i(t) = 1$ for $i = 1, \ldots, p$ becomes small in each interval [11] and we have:

$$\mathbb{P}[x_i(t) = 0] \approx 1 - \lambda_i(t), \tag{15}$$
$$\mathbb{P}[x_i(t) = 1] \approx \lambda_i(t), \tag{16}$$
$$\mathbb{P}[x_i(t) \geq 2] \approx 0. \tag{17}$$

The probability of observing $\mathbf{x}(t)$ in the t^{th} interval can be written as following:

$$\mathbb{P}[\mathbf{x}(t)|\mathbf{x}(t-1)] = \prod_{i=1}^{p}(\lambda_i(t))^{x_i(t)}(1-\lambda_i(t))^{1-x_i(t)}. \tag{18}$$

Now we can approximate the negative log-likelihood function as follows using the fact that when $\lambda_i(t)$ is small, we can write $\log(1-\lambda_i(t)) \approx -\lambda_i(t)$ and $\log(\lambda_i(t)[1-\lambda_i(t)]^{-1}) \approx \log(\lambda_i(t))$ [30].

$$\mathcal{L}_{\text{Obs}} \approx \sum_{i=1}^{p} x_i(t)\log(\lambda_i(t)) - \lambda_i(t). \tag{19}$$

Substituting the value of $\lambda_i(t)$ from the observed model in Eq. (14) into Eq. (19), we can see that $\widehat{A}_{MLE}$ is the solution of the following problem:

$$\widehat{A}_{MLE} = \arg\max_{\widehat{A}} \mathbb{E}_{\text{True}}[\mathcal{L}_{\text{Obs}}], \tag{20}$$

$$= \arg\max_{\widehat{A}} \left\{\mathbb{E}_{\text{True}}\left[\mathbf{x}(t)^\top \widehat{A}\mathbf{x}(t-1) - \mathbf{1}^\top \exp(\widehat{A}\mathbf{x}(t-1))\right]\right\}. \tag{21}$$

where we have written the equations in the compact vector format. Differentiation of $\mathcal{L}$ with respect to $\widehat{A}$ and setting it to zero yields the following results:

$$\mathbb{E}_{\text{True}}\left[\mathbf{x}(t-1)\mathbf{x}(t)^\top - \mathbf{x}(t-1)\exp(\widehat{A}\mathbf{x}(t-1))^\top\right] = \mathbf{0}, \tag{22}$$

$$\mathbb{E}_{\mathbf{u}(t-1)}\left[\mathbb{E}_{\mathbf{x}(t)|\mathbf{u}(t-1)}\left[\mathbf{x}(t-1)\mathbf{x}(t)^\top - \mathbf{x}(t-1)\exp(\widehat{A}\mathbf{x}(t-1))^\top\right]\right] = \mathbf{0}, \tag{23}$$

$$\mathbb{E}_{\mathbf{u}(t-1)}\left[\mathbf{x}(t-1)\left(\exp([A\ B]\mathbf{u}(t-1)) - \exp(\widehat{A}\mathbf{x}(t-1))\right)^\top\right] = \mathbf{0}. \tag{24}$$

where A and B are the true values in Eq. (13). Since $u_i \in \{0,1\}$ with high probability, by taking the expectation with respect to each individual u_i we can see that Eq. (24) is satisfied if and only if the following equality holds:

$$\mathbb{E}_{\mathbf{u}(t-1)}\left[\exp([A\ B]\mathbf{u}(t-1)) - \exp(\widehat{A}\mathbf{x}(t-1))\right] = \mathbf{0}. \tag{25}$$

Suppose A, B, and $\widehat{A}$ are bounded. Since $u_i \in \{0,1\}$, the values inside the exponential functions are bounded, and the exponential function is one to one. Thus, Eq. (25) is equivalent to the following equation: (which can also be obtained by Taylor expansion.)

$$\mathbb{E}_{\mathbf{u}(t-1)}\left[[A\ B]\mathbf{u}(t-1) - \widehat{A}\mathbf{x}(t-1)\right] = \mathbf{0}, \tag{26}$$

$$\mathbb{E}_{\mathbf{u}(t-1)}\left[B\mathbf{z}(t-1) - (\widehat{A}-A)\mathbf{x}(t-1)\right] = \mathbf{0}, \tag{27}$$

$$B\boldsymbol{\lambda}'(t-1) - (\widehat{A}-A)\boldsymbol{\lambda}(t-1) = \mathbf{0}, \tag{28}$$

where Eq. (28) is the result of linearity of expectation operator. Since Eq. (28) holds for all values of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$, the column space of $\widehat{A} - A$ is equal to the column space of B. Thus, rank of $L = \widehat{A} - A$ can be at most the rank of column space of B; i.e. $rank(L) \leq r$. This concludes the proof. The proof also holds for Bernoulli and COM-Poisson processes, due to the fact that Eq. (19) holds for them too [30].

Proof for the Gumbel GLARP

Consider the following *true* model for the Gumbel time series:

$$\mathbb{E}\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{z}(t)\end{bmatrix} = \begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}\mathbf{x}(t-1)\\ \mathbf{z}(t-1)\end{bmatrix} \tag{29}$$

In the maximum likelihood solution with unobserved time series $\mathbf{z}(t)$, we fit the data to the following *observed* model:

$$\mathbb{E}[\mathbf{x}(t)] = \widehat{A}\mathbf{x}(t-1) \tag{30}$$

Similar to the previous theorem, our goal is to find the expression for $\widehat{A}_{MLE}$ as in Eq. (20). The key to approximation of $\widehat{A}_{MLE}$ is to assume that $\mathbb{E}[\mathbf{x}(t)] = \mathbf{0}$ and $A\mathbf{x}(t)$ is small; both of these assumptions can be satisfied in the data by pre-processing. Proceeding with the proof, we have:

$$\widehat{A}_{MLE} = \arg\min_{\widehat{A}}\left\{\mathbb{E}_{\text{True}}\left[\sum_{i=1}^{p}\left(\frac{x_i(t)-\mu_i(t)}{\sigma} + \exp\left\{-\frac{x_i(t)-\mu_i(t)}{\sigma}\right\}\right)\right]\right\}, \tag{31}$$

Using the fact that $\mathbb{E}[\mathbf{x}(t)] = \mathbf{0}$ and differentiation with respect to A yields:

$$\mathbb{E}_{\text{True}}\left[\mathbf{x}(t-1)\exp\left\{-\frac{\mathbf{x}(t)-\widehat{A}\mathbf{x}(t-1)}{\sigma}\right\}^\top\right] = \mathbf{0}, \tag{32}$$

$$\mathbb{E}_{\text{True}}\left[\mathbf{x}(t-1)\left\{1-\frac{\mathbf{x}(t)-\widehat{A}\mathbf{x}(t-1)}{\sigma}\right\}^\top\right] \approx \mathbf{0}, \tag{33}$$

$$\mathbb{E}_{\text{True}}\left[\mathbf{x}(t-1)\left\{\mathbf{x}(t)-\widehat{A}\mathbf{x}(t-1)\right\}^\top\right] \approx \mathbf{0}, \tag{34}$$

$$\mathbb{E}_{\mathbf{u}(t-1)}\left[\mathbf{x}(t-1)\left\{A\mathbf{x}(t-1)+B\mathbf{z}(t-1)-\widehat{A}\mathbf{x}(t-1)\right\}^\top\right] \approx \mathbf{0}, \tag{35}$$

$$\widehat{A} \approx A + Q_{\mathbf{xx}}^{-1}Q_{\mathbf{xz}}B, \tag{36}$$

The step from (32) to (33) is due to the Taylor expansion of the exponential function around zero; the step from (33) to (34) is done using the fact that $\mathbb{E}[\mathbf{x}(t)] = \mathbf{0}$; the step from (34) to (35) is done by expectation with respect to conditional distribution of $\mathbf{x}(t)$ given $\mathbf{u}(t)$ under the true model; and the final step is done via the definition of the co-variance matrices.

Proof of Theorem 3.2

Due to space limit, we provide our proof as a continuation of the proof in [27]. Given a set S and a norm $|||$, we define the Restricted Smoothness Property constant of the likelihood function $\mathcal{L}$ as defined in Eq. (3) in [27]. Following the same steps, we have:

$$\begin{aligned}\mathcal{L}(\mathbf{w}_t + \alpha(\eta_S\mathbf{a}_t^{(S)}) + \beta(\eta_L\mathbf{a}_t^{(L)})) \leq\\ \mathcal{L}(\mathbf{w}_t) - \alpha(-\langle\nabla\mathcal{L}(\mathbf{w}_t), \eta_S\mathbf{a}_t^{(S)}\rangle + \langle\nabla_S\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t\rangle)\\ - \beta(-\langle\nabla\mathcal{L}(\mathbf{w}_t), \eta_L\mathbf{a}_t^{(L)}\rangle + \langle\nabla_L\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t\rangle)\\ + 2\alpha^2L_S\eta_SR_S^2 + 2\beta^2L_L\eta_LR_L^2\end{aligned} \tag{37}$$

Similarly, we can define and show that:

$$\begin{aligned}\delta_t &\triangleq \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}^\star)\\ &\leq -\langle\nabla\mathcal{L}(\mathbf{w}_t), \mathbf{w}^{\star,(L)}\rangle + \langle\nabla_S\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t\rangle\\ &\quad - \langle\nabla\mathcal{L}(\mathbf{w}_t), \mathbf{w}^{\star,(S)}\rangle + \langle\nabla_L\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t\rangle \qquad (38)\\ &\leq -\langle\nabla\mathcal{L}(\mathbf{w}_t), \eta_S\mathbf{a}_t^{(S)}\rangle + \langle\nabla_S\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t\rangle\\ &\quad - \langle\nabla\mathcal{L}(\mathbf{w}_t), \eta_L\mathbf{a}_t^{(L)}\rangle + \langle\nabla_L\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t\rangle \qquad (39)\end{aligned}$$

Plugging Eq. (39) into Eq. (37) and following the reasoning in [27], we can show that:

$$\delta_{t+1} \leq \delta_t + \min_{\alpha,\beta\in[0,1]}(-(\alpha+\beta)\delta_t + 2\alpha^2L_S\eta_SR_S^2 + 2\beta^2L_L\eta_LR_L^2)$$

For $t = 0$, choose $\alpha, \beta = 1$ on the right hand side to get $\delta_1 \leq 2(L_S\eta_SR_S^2 + L_L\eta_LR_L^2)$. Since δ_t is decreasing, we can see that $\delta_t \leq 2(L_S\eta_SR_S^2 + L_L\eta_LR_L^2)$ for all t. Thus, choosing $\alpha = 4(L_S\eta_SR_S^2 + L_L\eta_LR_L^2)$ yields for all $t > 1$: $\delta_{t+1} \leq \delta_t - \frac{\delta_t^2}{\mathfrak{B}_S+\mathfrak{B}_L}$ where $\mathfrak{B}_S \triangleq 8L_S\eta_SR_S^2$ and $\mathfrak{B}_L \triangleq 8L_L\eta_LR_L^2$. Solving this yields the desired result. The impact of optimization of $\mathbf{b}$ can be captured similarly. □

In the performance analysis of the greedy algorithm that selects only one sparse or low-rank atom per iteration we should observe that in Eq. (38) either $\langle\nabla\mathcal{L}(\mathbf{w}_t), \mathbf{w}^{\star,(L)}\rangle$ or $\langle\nabla\mathcal{S}(\mathbf{w}_t), \mathbf{w}^{\star,(S)}\rangle$ remains unbounded. Bounding this term introduces the Lipschitz constant of the likelihood function.

Plugging the additional term into Eq. (37) yields $\alpha\mathfrak{L}_S\eta_SR_S^2$ or $\alpha\mathfrak{L}_L\eta_LR_L^2$ instead of $\alpha^2L_S\eta_SR_S^2$ or $\alpha^2L_S\eta_SR_S^2$. The term $\mathfrak{L}_S$ denotes the maximum Lipschitz constant of the likelihood function inside the convex hull of the sparsity norm. Since $\alpha < 1$ and always $\mathfrak{L}_S < L_S$ and $\mathfrak{L}_L < L_L$, we observe that the bound for the single atom selection should be at least greater by the differences of the Lipschitz and the restricted smoothness constant of the likelihood function for one of the atoms.