Predicting Workplace Incidents with Temporal Graph-guided Fused Lasso

Keerthiram Murugesan[1] and Jaime Carbonell[1]

[1]Language Technologies Institute
Carnegie Mellon University
Pittsburgh, USA
CMU-LTI-15-?? Technical Report

October 25, 2015

Abstract

We consider the problem of learning a sparse structured regression that utilizes both temporal information and structural information in the data. Our goal is to enforce sparsity in both the coefficients and their temporal differences, which is desirable for applications with data that evolves over time. We propose a regularization function based on fused lasso penalty that encourages temporal smoothness and exploits the graph structure over the features to borrow information across correlated features. Using simulated and real data sets, we demonstrate that our methods outperform their non-temporal variants in terms of both minimizing prediction errors and recovery of true sparse underlying patterns.

Introduction

Many real world applications, such as stock market, social network, network traffic, genomics, etc, exhibit a time-varying characteristics. In this paper, we consider one such application, where the data is associated with temporal information (time stamp) and a complex, time-varying dependency structure over the feature set (correlation graph). This paper focuses on an application that involves the prediction of number of workplace incidents based on the safety inspection data. Typically, the data is collected manually from different locations for different projects by safety inspectors, but these data may also be collected from sensors deployed in different locations.

The observation are recorded based on *safety checklist*, a set of simple yes/no questions such as 'Is the person wearing a head protection?' or 'Are there proper ventilation for the job?'. These questions are categorized and they do overlap for different projects

handled in a location. The severity of the incidents ranges from simple sprain to person's death and are collected based on *OSHA* standards [1]. In this year alone, there have been 825 fatalities reported to OSHA from across the country. There is a necessity to predict the number of incidents in the future to prevent the actual injuries.

The varying coefficient model is an important generalization of linear regression model and has been studied extensively over the past decade [4, 2]. The model assumes that the regression coefficients are an unknown function of some other variables, called effect modifiers [3]. In this paper, we consider the regression coefficients as functions of time. For a set of random variables $\mathbf{Y}, \mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_p}$ and $\mathbf{T}$, a time-varying coefficient model has the following form:

$$Y_i = \mathbf{X}_i' \boldsymbol{\beta}(t_i) + \epsilon_i, i = 1, \ldots, N$$

We assume that the regression coefficient function is multidimensional piecewise constant with coefficient values and the dependency structure, i.e, there exists a K-way partition of the sample $\{1, \ldots, N\}$ i.e., $[t_1 = \tau_1 < \tau_2 < \ldots < \tau_K = t_N]$, such that, $\boldsymbol{\beta}(t) = \boldsymbol{\beta}_{\tau_k}$, for $t \in [\tau_{(k-1)}, \tau_k)$ and the correlation graph associated with this interval, G_{τ_k} is same for all $\mathbf{X}_i, i \in [\tau_{(k-1)}, \tau_k)$. Since we are dealing with high dimensional problem, we use the $l1-$sparsity constraint to select the subset of features (or subgraph of the features) relevant to the outcome ([10]).

Our problem involves two unknowns: the K-way partition set and the graph constrained and sparsity constrained regression coefficient function. Note that both the size of the partition set (K) and the actual partition set are unknown. Several papers have discussed this problem (multiple change-point detection, [1]). Most recently, [7] and [11] proposed change-point detection for signal approximation with lasso and group lasso. While the former is proposed for one dimensional signal approximation, the latter can be extended to multidimensional signal approximation. Based on [7], [6] estimated the partition boundaries for the varying coefficient models with fused lasso, which penalize the coefficient values of adjacent temporal difference in each dimension separately. In our application, the model requires that both the coefficient values and their dependency structure changes at the (unknown) partition boundaries. To enforce this constraint, we estimate the multiple change points with fused group lasso, instead of fused lasso penalty. It ensures that all the coefficient values change at the partition boundaries, with which we can estimate correlation graph for each partition separately [11]. From these partitions, we can estimate the regression coefficient with graph and sparsity constraints. In this paper, we use graph-structured fusion penalty to estimate the coefficients with network constraints [8, 5].

The rest of the paper is organized as follows: In the next section, we present temporally-fused lasso and its variants. We provide the parameter estimation for our temporal models. Finally, we demonstrate the performance of our temporal models on simulated and workplace incident dataset, followed by conclusion.

[1] https://www.osha.gov/

The Methodology

Notations

Suppose we have a sample of N observations, each represented by a p-dimensional feature vector. Let $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}) \in \mathbb{R}^{N \times p}$ be the input matrix and $\mathbf{Y} \in \mathbb{R}^N$ represent the outcome variable. We assume a varying coefficient model discussed in the introduction. Let $\boldsymbol{\beta}(t_i) \in \mathbb{R}^p$ be the regression coefficient vector associated with the ith observation. We assume that the feature variables $\mathbf{x_j}$ are centered to have zero mean and unit variance, and the outcome variable $\mathbf{y}$ has mean 0, so that, we consider the model without an intercept.

Temporal Graph-guided Fused Lasso

For our model assumptions, we consider the temporal graph-guided fused lasso penalty to predict workplace incidents. We analyze the following estimator for our application:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{i=1}^{N} (Y_i - \mathbf{X}_i \boldsymbol{\beta}(t_i))^2 + \lambda \sum_i ||\boldsymbol{\beta}(t_i)||_1$$

$$+\gamma_1 \sum_i ||\boldsymbol{\beta}(t_i)' G(\boldsymbol{\beta}(t_i)||^\alpha + \gamma_2 \sum_{i=2}^{N} ||\boldsymbol{\beta}(t_i) - \boldsymbol{\beta}(t_{(i-1)})||_2 \quad (1)$$

where $\{\lambda, \gamma_1, \gamma_2\}$ are regression coefficients that depend on N. The loss function estimates the square loss between the true number of incidents and the estimated number of incidents. The first penalty term promotes sparsity in the regression coefficients, the second penalty induces graph structured constraints over the feature set and the third penalty term identifies the partition boundaries.

For $V = \{1, \ldots, p\}$ and $E = \{(m, l) : \forall (m, l) \in V \times V$ and $w(r_{ml}) \geq \rho\}$, $G(\boldsymbol{\beta}(t_i))$ is a $|V| \times |E|$ matrix with entries: $(m, (m, l)) = \sqrt{(w(r_{ml}))}$ and $(l, (m, l)) = -sign(r_{ml})\sqrt{(w(r_{ml}))}$. $w(r_{ml}) \in \mathbb{R}$ denote the weight of the edge $e = (m, l) \in E$. We choose r_{ml} to represent the strength of correlation between $\mathbf{x_m}$ and $\mathbf{x_l}$. In this paper, we use a simple strategy for constructing the feature graph G by computing a pairwise correlation between $\mathbf{x_m}$ and $\mathbf{x_l}$ and taking $w(r_{ml}) = |r_{ml}|$.

The parameter α takes the value in $\{1, 2\}$. When $\alpha = 1$, the graph constrained penalty forces the features sharing an edge to take the same coefficient values and when $\alpha = 2$, their coefficient values are closer to each other.

Estimation Procedures

When the partition boundaries and the graph structures of each partition are available a priori, equation 1 can be easily optimized. With the partition boundaries unknown (and hence the associated correlation graph), we consider an adaptive procedure to estimate the values of the regression coefficients in equation 1. Due to the bias introduced by the fusion penalty [9], we use a two-stage procedure: (1) estimate the partition boundaries (2) estimate the regression coefficients with *l1*-sparsity and graph constraints.

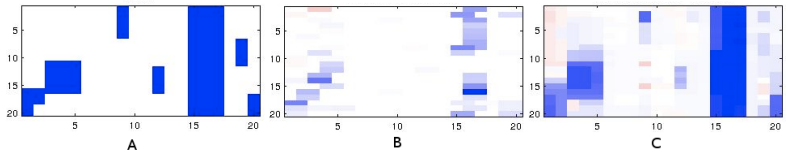


Figure 1: Regression coefficients estimated by different models for a simulated dataset ($p = 20, K = 20$). We used threshold $\rho = 0.6$ for correlation graph and signal strength $b = 0.8$. Blue pixels indicate positive values. (A) True coefficient matrix with each row corresponds to $\boldsymbol{\beta}(t_{\tau_k})$ (B) *TESLA* (C) *Our Procedure*.

Estimating Partition Boundaries

We first estimate the partition boundaries with group fused lasso.

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{i=1}^{N} (Y_i - \mathbf{X}_i \boldsymbol{\beta}(t_i))^2 + \gamma_2 \sum_{i=2}^{N} ||\boldsymbol{\beta}(t_i) - \boldsymbol{\beta}(t_{(i-1)})||_2 \tag{2}$$

Proposition 1. *Given dataset* $(\mathbf{X}, \mathbf{Y})$*, we define*

$$\mathbf{X}^\dagger = (\mathbf{x}_1^\dagger, \mathbf{x}_2^\dagger, \ldots, \mathbf{x}_p^\dagger) \in \mathbb{R}^{N \times Np},$$

$$\mathbf{x}_i^\dagger = \begin{pmatrix} X_1 & 0 & \cdots & 0 \\ X_2 - X_1 & X_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ X_N - X_{N-1} & X_N - X_{N-1} & \cdots & X_N - X_{N-1} \end{pmatrix}$$

(we removed the subscript i for clarity)

$$\boldsymbol{\beta}^\dagger = (\boldsymbol{\beta}_{t_1}^\dagger \boldsymbol{\beta}_{t_2}^\dagger \ldots \boldsymbol{\beta}_{t_N}^\dagger)' \in \mathbb{R}^{Np}$$
$$\boldsymbol{\beta}_{t_i}^\dagger = \boldsymbol{\beta}_{t_i} - \boldsymbol{\beta}_{t_{i-1}}, \text{ for } i = 2 \ldots N$$
$$\text{and } Y_i^\dagger = Y_i - Y_{i-1},\ Y_1^\dagger = Y_1$$

Equation 2 can be written as

$$\min_{\boldsymbol{\beta}^\dagger \in \mathbb{R}^{Np}} \frac{1}{2} ||\mathbf{Y}^\dagger - \mathbf{X}^\dagger \boldsymbol{\beta}^\dagger||_2^2 + \gamma_2 \sum_{i=2}^{N} ||(\boldsymbol{\beta}^\dagger)_i||_2 \tag{3}$$

with ith group containing elements of the vector $(\boldsymbol{\beta}_{t_i} - \boldsymbol{\beta}_{t_{i-1}})$.

Equation 3 is an objective function with group lasso penalty, which has been studied extensively. We use FISTA given in Appendix to solve the above optimization problem. From the estimates of $\boldsymbol{\beta}^\dagger$, we can get partition boundaries where $\boldsymbol{\beta}^\dagger \neq 0$.

Estimating Regression Coefficients Let $\tau_k, k = 1 \dots K$ be the partition boundaries estimated from the previous step. We compute the correlation graph for each partition $G(\beta(\tau_k))$. We construct $X^+ = diag(X_{\tau_1}, X_{\tau_2}, \dots, X_{\tau_k})$, a block diagonal matrix with diagonal elements correspond to the matrix with observations from each partition, estimated in the previous step.

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{i=1}^{N} (Y_i - \mathbf{X}_i \boldsymbol{\beta}(t_i))^2 + \lambda \sum_i ||\boldsymbol{\beta}(t_i)||_1 + \gamma_1 \sum_i ||\boldsymbol{\beta}(t_i)' G(\boldsymbol{\beta}(t_i)||^\alpha \tag{4}$$

When $\alpha = 1$, we can use a coordinate descent algorithm given in Annex A.

Proposition 2. *Given $\alpha = 2$, we write*

$$\mathbf{X}^* = \frac{1}{\sqrt{(1+\gamma_1)}} \begin{pmatrix} \mathbf{X}^+ \\ \sqrt{\gamma_1} G' \end{pmatrix}, \mathbf{Y}^* = \begin{pmatrix} \mathbf{Y} \\ 0 \end{pmatrix} \tag{5}$$

where $G \in \mathbb{R}^{p \times \sum_\tau |E|_\tau}$ is a block diagonal with $|V| \times |E|$ matrices, one for each partition. Let $\gamma = \frac{\lambda}{\sqrt{1+\gamma_1}}$ and $\boldsymbol{\beta}^* = \sqrt{1+\gamma_1}\boldsymbol{\beta}$, then equation 4 is equivalent to:

$$min_{\boldsymbol{\beta}^*} ||\mathbf{Y}^* - \mathbf{X}^* \boldsymbol{\beta}^*||_2^2 + \gamma ||\boldsymbol{\beta}^*||_1 \tag{6}$$

and the solution to equation 4 is $\boldsymbol{\beta} = \frac{1}{\sqrt{1+\gamma_1}} \boldsymbol{\beta}^*$

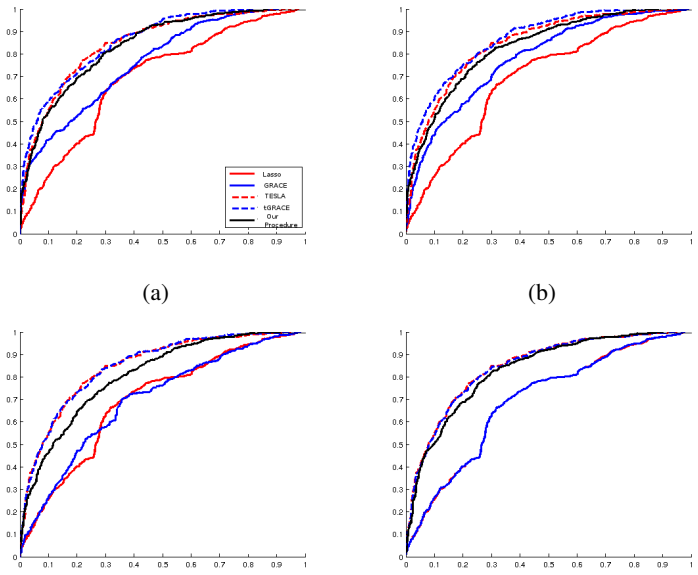
Experiments

In this section, we show the performance of the our procedure discussed in the previous section on simulated and real datasets. We use *Lasso* [10], *GRACE* [8] and *TESLA* [6] as our baselines. In addition to our procedure, we consider a model where $G(\beta(t_1)) = G(\beta(t_2)) = \dots = G(\beta(t_N)$ in equation 4, i.e., we use the same correlation graph estimated on $(\mathbf{X})$ for all partition instead of estimating correlation graph on $(\mathbf{X}_{\tau_k})$ for each partition. We call it *tGRACE* for temporal data. We choose the regularization parameters λ and $\boldsymbol{\gamma}$ by minimizing BIC criterion. We assume that the input matrix $\mathbf{X}$ and the outcome variable $\mathbf{Y}$ are standardized before the experiments.

Simulation Study

We conduct a simulation study to compare the performance of *Lasso*, *GRACE*, *TESLA* and our procedure. We use $K = 20$ partitions of the input matrix with 20 features for generating simulated datasets. We choose the number of observations in each temporal segment τ_k, $N_{\tau_k} \approx 10$, for both training and test data.

We generate the data with both temporal dependencies and feature correlations using the following procedure: we choose subsets $\{\{1\}, \{1,2\}, \{3,4,5\}, \{1,2,3,4,5\}, \{9\}, \{12\}, \{15,16,17\}, \{19\}, \{20\}\}$ of features as groups, and used these groups to sample a covariance matrix Σ for each group separately. We generate the observations for each partition from a



(a) (b)

(c) (d)

Figure 2: Comparison of ROC curves for the recovery of true sparsity patterns with varying threshold ρ for feature correlation graph. (a) $\rho = 0.1$, (b) $\rho = 0.3$, (c) $\rho = 0.5$, (d) $\rho = 0.7$. Results are averaged over 50 simulated datasets. We use $b = 0.8$ for signal strength. X-axis: False positive rate; Y-axis: True positive rate.

multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance Σ, based on the sparsity pattern of $\boldsymbol{\beta}(t_{\tau_k})$ given in Figure (1A). Each non-zero values of $\boldsymbol{\beta}(t_{\tau_k})$ is set to 0.8.

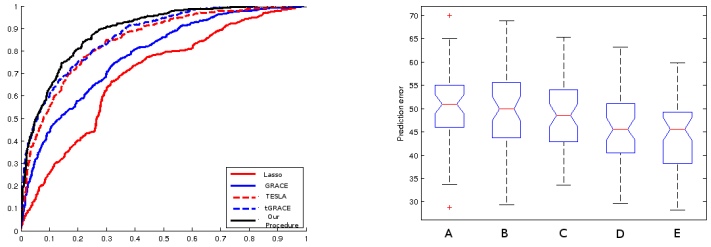
For the first part of the simulated experiment, we restrict our attention to *TESLA* and our procedure for recovering the true sparsity pattern in temporal domain. We estimate the regression coefficients $\boldsymbol{\beta}$ on the simulated training data to compare them. Figure 1(A) shows the true regression coefficients $\boldsymbol{\beta}$ for each partition. Figures (1B) and (1C) show the regression coefficients $\hat{\boldsymbol{\beta}}$ estimated by TESLA and our procedure.

For $alpha = 2$, the coefficient estimates of $\boldsymbol{\beta}$, for example $\{15, 16, 17\}$, will be closer to each other, but may not be same. In order to recover the exact sparsity pattern, we use $\alpha = 1$ in equation 4 for the simulated experiment. Note that there is no need to use $\alpha = 1$ unless the application requires it. We use it in the simulated experiment for the better visualization. We will use $alpha = 2$ for the real data. Clearly we can see that our procedure succeeds in recovering the true model. *TESLA* recovers the temporal smoothness, but without the feature correlation, *TESLA* couldn't recover the exact support of the regression coefficients.

We evaluate the models on test data with the receiver operating characteristics (ROC) curves for the recovery of true sparsity pattern and prediction errors. performance of the our procedure discussed in the previous section on simulated and real datasets. We use *Lasso* , *GRACE* and *TESLA* to compare it against our procedure. We

study the importance of threshold ρ for correlation graph used in *GRACE*, *tGRACE* and our procedure.As we can see in Figure 2, *TESLA*, *tGRACE* and our procedure consistently outperform *Lasso* and *GRACE*, regardless of the threshold ρ, which make sense since we are dealing with temporal data.

For lower values of ρ, more edges are added to the correlation graph that includes edges E with strong correlations and edges with weak correlations due to the added noise in both covariance matrix and the input data. It is worth noting that *tGRACE* exhibits better performances than the other models for $\rho = 0.3$, since it uses the correlation constructed from the entire data. This reduces the number of spurious edges added to the edge set E. When the threshold is higher (say $\rho = 0.7$), there are no edges in the feature graph, and thus removes the graph constrained penalty function. As a result, the performance curves of *TESLA* and *tGRACE* almost entirely overlap. Similar performances can be noticed for *GRACE* and Lasso.



(a) (b)

Figure 3: ROC Curve (left) and prediction error (right) estimated for (A) *Lasso*, (B) *GRACE*, (C) *TESLA* and (D) *tGRACE* with $\rho = 0.3$ and for our procedure (E) with $\rho = 0.85$. The results are averaged over 50 simulated datasets. (Left) X-axis: False positive rate; Y-axis: True positive rate. (Right) Box shows the lower quartile, median, and upper quartile values, and the whiskers show the range of prediction errors in the 50 simulated datasets.

Even though the performances of our procedure are better than *Lasso* and *GRACE*, we can see that our procedure performs worse than *TESLA* and *tGRACE* for all the values of ρ. As mentioned before, since the correlation graph $G(\boldsymbol{\beta}(t_{\tau_k}))$ is estimated on a subset of the observation, we notice that significant number of spurious edges are added to the edge set E_{τ_k}, even when $\rho = 0.7$. To overcome this problem, we choose the threshold value separately for our procedure to filter these noisy edges.

We repeat the experiment with $\rho = 0.85$ for our procedure. For the other models, we choose the threshold that gave the best results in Figure 2 ($\rho = 0.3$). Figure 3 shows the updated ROC curve and prediction errors for the regression models. We can see that our procedure now performs significantly better than all the other models, by utilizing correlation between the features in each partition.

The evaluation results shown in Figures 2 and 3 are averaged over 50 randomly generated datasets.

Location/ Model	Lasso	Elastic Net	GRACE	TESLA	Our Procedure
Complex 0	5.2803	4.7049	3.2828	5.6586	2.2483
Complex 1	14.0825	11.0319	0.5159	2.6186	0.5057
Complex 2	0.3901	0.3891	0.3075	0.2913	0.2932
Complex 3	10.8767	8.3318	0.3132	1.3766	0.3258
Complex 4	4.3131	4.0839	0.5904	0.9435	0.4996
Complex 5	0.9010	0.7970	0.1666	0.6119	0.1429
Complex 6	2.6956	2.6534	0.3143	0.9058	0.2924
All	0.6186	0.6186	0.6173	0.9194	0.5447

Table 1: Mean Squared Error (MSE) estimated for *Lasso*, *Elastic Net*, *GRACE*, *TESLA* and *Our Procedure* on all datasets. Threshold $\rho = 0.3$

Workplace Incident Dataset

Workplace incident dataset contains inspections and the records of incidents collected from different projects at different locations in 2011 and 2012. The dataset consists of $\approx$ 4000 observations ($\approx$ 2000 observations in each year) with 51 features collected from different locations, each associated with a time stamp. Each observation is a weekly summary of safety inspections at a location with number of incidents happened in that week as an outcome.

We conduct a preliminary experiment to test our real data with each model. For the first part of our experiment, we consider naive assumptions about the dataset. We assume that the partition boundaries are given. We split the dataset into $K = 12$ partitions based on months. We also assume that the features with shared edge take same coefficient values i.e., we set $\alpha = 1$. We use the observations from 2012 for training/validation and the observations from the first month of 2012 as test set. We use the regression coefficient $\beta(t_{12})$ to predict the number of incidents in the test set.

We use training set to learn the regression coefficients and test set to measure the performance of regression models. As in the simulation study, we notice that smaller value for threshold ρ hurts the performance of our procedure. We pick the threshold value for our procedure separately. We randomly sample data from each partition to build the validation set. We use this validation set to choose $\rho = 0.5$ for *GRACE*, *tGRACE*, and $\rho = 0.9$ for our procedure.

Figure 4 shows the prediction error for each model. We use sum of absolute errors ($||\mathbf{Y} - \hat{\mathbf{Y}}||_1$) to compare the performance of the models. We removed the *Lasso* results since the estimates were beyond the y-axis scale shown in the figure. As we see in Figure 4 that our procedure performs significantly better than other models. This is good but these results doesn't give any information to explain the model behavior.

Each partition contains weekly observations collected from different locations, so it is hard to interpret the regression coefficient values, as these values might be influenced by the observations from the locations with larger projects. Moreover, we noticed that there are several locations with temporal gap in the observations, either because there were no projects during that period or that location handles small projects. Nonethe-

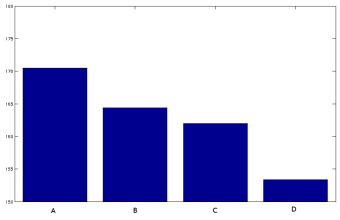


Figure 4: Prediction errors for the workplace incident dataset. X-axis: (left-to-right) (A) *GRACE*, (B) *TESLA*, (C) *tGRACE* and (D) *Our Procedure*; Y-axis: $||\mathbf{Y} - \hat{\mathbf{Y}}||_1$ (Sum of absolute errors).

less, even with the shaky assumptions and unreliable data, *TESLA*, *tGRACE* and *Our Procedure* did better than *GRACE* and *Lasso*.

In the second part of our experiment, we considered the original problem where the partition set size K and the actual partition set are unknown. In addition, we restricted ourselves to 7 locations for which the observations are available for all two years ($Complex0$, $Complex1$, $Complex2$, $Complex3$, $Complex4$, $Complex5$ and $Complex6$). Each dataset now contains 104 observations, one for each week. We used every other observations for test data, i.e., observations with odd week number belong to training data and the observations with even week number belong to test dataset. We used this way to split the dataset so that we can use the regression coefficient $\boldsymbol{\beta}_{\tau_{t_i}}$ of week i in the training data to predict the number of incidents for the week $(i + 1)$ in the test data.

We chose *GRACE* and *TESLA* as our baseline models for *our procedure*. We included *Lasso* and *Elastic Net* to compare the results with [8]. We used Mean Squared Error (MSE: $\frac{1}{N}||\mathbf{Y} - \hat{\mathbf{Y}}||_2^2$) to compare the models. Table 1 shows the results. In addition, we included the result for dataset generated by combining observations from all 7 locations.

Lasso, *Elastic Net* and *GRACE* results are consistent with the results produced in [8]. We can see that our procedure gave good results in most of the locations. Interestingly, *GRACE* outperforms *TESLA* in most of the cases. This explains the importance of using graph constraints to estimate regression coefficients. *TESLA* performs little better than *GRACE* and *Our Procedure* in $Complex2$ due to the limited number of inspections available for that location. This is consistent with [6]'s results for finite sample data. It is worth noting that *Our Procedure* gave much smalled MSE than other models when we use all the datasets. This is due to the availability of better correlation graphs for each partition estimated with additional data from all locations.

We used the regularization path for *TESLA* and *Our Procedure* to study the model behaviors. We noticed that after computing the change points, almost all the locations contain change points around week 40 and week 90. A simple observation revealed that the number of inspections and the number of incidents were significantly low in the last few weeks of an year due to the holidays, compared to the previous week. This shows that change point detection identifies the seasonality which is common in our

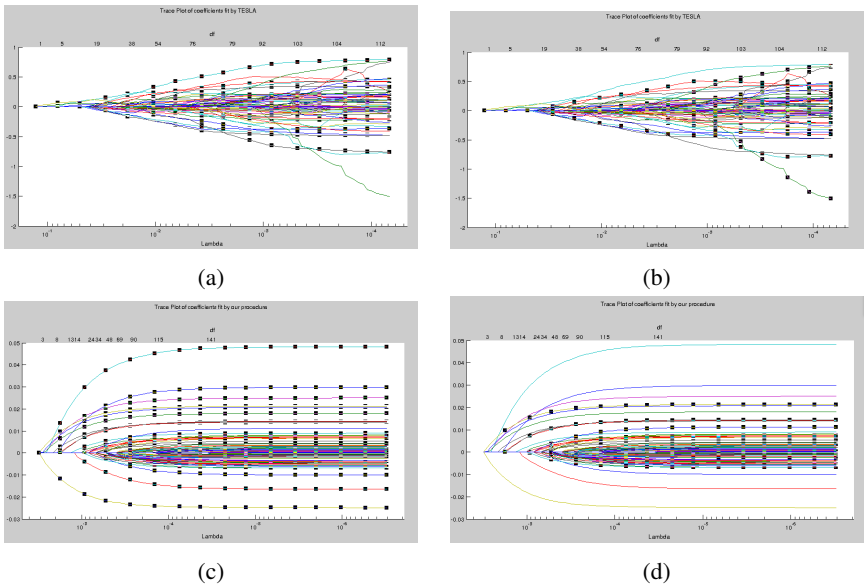


Figure 5: Regularization Path for *TESLA* (Top) and *Our Procedure* (Bottom) showing the last two partitions for the location (Complex 5)

application.

Based on our previous observations, we analyzed the regularization paths for the last two partitions under *TESLA* and *Our Procedure*. Figure 5 shows the regularization paths for *TESLA* and *Our Procedure*. Left column contains the regularization paths for $(K-1)th$ partition and right column contains the regularization paths for Kth partition for $Complex5$. We can see that with graph constraints over feature set, the regularization paths of our procedure are well formed. The correlated features often travel together in the regularization path. Similar behavior can be noticed in regularization path of *GRACE*. The dotted lines in the left and right paths correspond to the features that belong to $(K-1)th$ and Kth partitions respectively. We can clearly see that the correlation between the features changed between the partition boundaries, which is captured by our procedure.

Conclusion

We considered sparse regression models for our application that encourage temporal smoothness and utilize the graph structure over the feature set to learn the true sparsity pattern in the regression coefficients. We demonstrated with simulated and real datasets that using additional information about the feature correlation can improve the prediction performance.

We noticed that the performance of *tGRACE* and *our procedure* depends on the quality of the available feature graph. In this paper, we have used a simple strategy

based on pair-wise correlation to construct the feature graph, more sophisticated methods can be employed for constructing this graph. For example, we can learn both structure and regression coefficients together. This is especially useful when we consider a dynamic network where the structure changes frequently over time. Although this paper focused on one application, this procedure can be easily extended to other applications such as modeling disease progression, survival analysis, financial data and genome analysis, etc.

References

[1] Jushan Bai. Estimation of a change point in multiple regression models. *Review of Economics and Statistics*, 79(4):551–563, 1997.

[2] Jianqing Fan and Wenyang Zhang. Statistical estimation in varying coefficient models. *Annals of Statistics*, pages 1491–1518, 1999.

[3] Jianqing Fan and Wenyang Zhang. Statistical methods with varying coefficient models. *Statistics and its Interface*, 1(1):179, 2008.

[4] Trevor Hastie and Robert Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 757–796, 1993.

[5] Seyoung Kim and Eric P Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS genetics*, 5(8):e1000587, 2009.

[6] Mladen Kolar, Le Song, and Eric P Xing. Sparsistent learning of varying-coefficient models with structural changes. In *Advances in Neural Information Processing Systems*, pages 1006–1014, 2009.

[7] Céline Levy-leduc and Zaïd Harchaoui. Catching change-points with lasso. In *Advances in Neural Information Processing Systems*, pages 617–624, 2008.

[8] Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

[9] Alessandro Rinaldo et al. Properties and refinements of the fused lasso. *The Annals of Statistics*, 37(5B):2922–2952, 2009.

[10] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[11] Jean-Philippe Vert and Kevin Bleakley. Fast detection of multiple change-points shared by many signals using group lars. In *Advances in Neural Information Processing Systems*, pages 2343–2351, 2010.

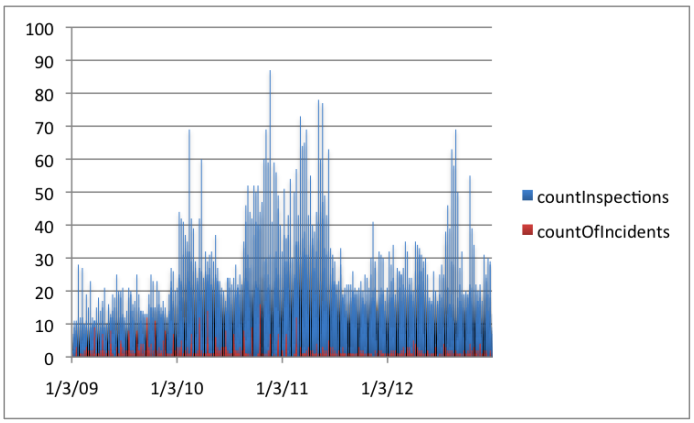


Figure 6: Number of Incidents/Inspections Vs Week for Workplace Incident Dataset.

Supplementary Materials

Additional Plot

The above plot shows the number of incidents & inspections observed over 4 years.

Optimization

Algorithm: Group Lasso with FISTA

Data: $\boldsymbol{\beta}^0, \mathbf{X}, \mathbf{Y}, \gamma_2, L - Lipschitzconstant$
Result: $\hat{\boldsymbol{\beta}}$
$\boldsymbol{w}^0 = \boldsymbol{\beta}^0, \alpha^1 = 1;$
while *Convergence met* **do**
$g^k = -\mathbf{X}'(\mathbf{Y} - \mathbf{X}\boldsymbol{w^k})$
$\boldsymbol{\beta}^k = \left(1 - \frac{\gamma_2}{L||w^k - \frac{1}{L}g^k||_2}\right)_+ (w^k - \frac{1}{L}g^k)$
$\alpha^k = (w^{k+1} - \boldsymbol{\beta}^k)'(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k) > 0?1 : alpha^k$
$\alpha^{k+1} = \frac{1}{2}1 + \sqrt{1 + 4(\alpha^k)^2}$
$w^{k+1} = \boldsymbol{\beta}^k + \frac{\alpha^k - 1}{a^{k+1}}(\boldsymbol{\beta}^k - \boldsymbol{\beta}^{k-1})$
end

Algorithm 1: FISTA for detecting multiple change points with group fused lasso

Optimizing Equation 4

When $\alpha = 1$**,** equation 4 contains a non-smooth function which cannot be optimized. There are efficient methods to solve this equation (Such as FISTA, ADMM, etc.), but we use a simplest approach. We consider a smooth approximation of the non-smooth

function by introducing additional variables $d_{\tau,j}, d'_{\tau,\tau-1,j}, d''_{\tau,m,l}$ that need to be estimated, along with the regression coefficients.

$$\begin{aligned}
\min_{\mathbf{B},\mathbf{d}} \sum_{\tau=1}^{T} ||\mathbf{Y}_\tau - \mathbf{X}_\tau \boldsymbol{\beta}_\tau||_2^2 + \lambda \sum_{\tau=1}^{T}\sum_{j=1}^{p} \frac{\beta_\tau^j}{d_{\tau,j}} \\
+\gamma_1 \sum_{\tau=2}^{T}\sum_{j=1}^{p} \frac{(\beta_\tau^j - \beta_{\tau-1}^j)^2}{d'_{\tau,\tau-1,j}} \\
+\gamma_2 \sum_{\tau=1}^{T} \sum_{(m,l)\in E_\tau} w^2(r_{ml}) \frac{(\beta_\tau^m - sign(r_{ml})\beta_\tau^m)^2}{d''_{\tau,m,l}} \\
\text{s.t} \sum_{\tau=2}^{T}\sum_{j=1}^{p} d'_{\tau,\tau-1,j} = 1, \qquad \sum_{\tau=1}^{T} \sum_{(m,l)\in E_\tau} d''_{\tau,m,l} = 1 \\
\sum_{\tau=1}^{T}\sum_{j=1}^{p} d_{\tau,j} = 1, \\
d'_{\tau,\tau-1,j} > 0, \qquad d''_{\tau,m,l} > 0, \\
d_{\tau,j} > 0, \quad \forall\ \tau, j, m, l
\end{aligned} \tag{7}$$

We solve the above problem by optimizing $d_{\tau,j}, d'_{\tau,\tau-1,j}, d''_{\tau,m,l}$ and $\beta_{\tau,j}$. In each iteration, we fix the value of $\beta_{\tau,j}$ and estimate $d_{\tau,j}, d'_{\tau,\tau-1,j}, d''_{\tau,m,l}$, by taking their derivatives and setting it to 0. We get the following update equations for $d_{\tau,j}, d'_{\tau,\tau-1,j}, d''_{\tau,m,l}$:

$$\begin{aligned}
d_{\tau,j} &= \frac{|\beta_\tau^j|}{\sum_{\tau',j'} |\beta_{\tau'}^{j'}|} \\
d'_{\tau,\tau-1,j} &= \frac{|\beta_\tau^j - \beta_{\tau-1}^j|}{\sum_{\tau',j'} |\beta_{\tau'}^{j'} - \beta_{\tau'-1}^{j'}|} \\
d''_{\tau,m,l} &= \frac{w(r_{ml})|\beta_\tau^m - sign(r_{ml})\beta_\tau^m|}{\sum_{\tau',(m',l')\in E_\tau} w(r_{m'l'})|\beta_{\tau'}^{m'} - sign(r_{m'l'})\beta_{\tau'}^{l'}|}
\end{aligned} \tag{8}$$

Based on the current estimates for $d_{\tau,j}, d'_{\tau,\tau-1,j}, d''_{\tau,m,l}$, we optimize over $\beta_{\tau,j}$ using the following update equation:

$$\begin{aligned}
\beta_\tau^j = \Bigg\{ & \sum_{n=1}^{N_\tau} x_n^j \big(y_n - \sum_{j' \neq j} x_n^{j'} \beta_\tau^{j'}\big) \\
& + \gamma_1 \left(\frac{\beta_{\tau-1}^j}{d'_{\tau,\tau-1,j}} \mathbb{1}(\tau > 1) + \frac{\beta_{\tau+1}^j}{d'_{\tau+1,\tau,j}} \mathbb{1}(\tau < T) \right) \\
& + \gamma_2 \left(\sum_{(m,j) \in E_\tau} w^2(r_{mj}) \frac{sign(r_{mj}) \beta_\tau^m)^2}{d''_{\tau,m,j}} \right) \\
& + \gamma_2 \left(\sum_{(j,l) \in E_\tau} w^2(r_{jl}) \frac{sign(r_{jl}) \beta_\tau^l)^2}{d''_{\tau,j,l}} \right) \Bigg\} \\
\Bigg/ \Bigg\{ & \sum_{n=1}^{N_\tau} x_j^{n\,2} + \frac{\lambda}{d_{\tau,j}} + \gamma_1 \left(\frac{1}{d'_{\tau,\tau-1,j}} + \frac{1}{d'_{\tau+1,\tau,j}} \right) \\
& + \gamma_2 \left(\sum_{(m,j) \in E_\tau} \frac{w^2(r_{mj})}{d''_{\tau,m,j}} + \sum_{(j,l) \in E_\tau} \frac{w^2(r_{jl})}{d''_{\tau,j,l}} \right) \Bigg\}
\end{aligned} \tag{9}$$

We repeat the above two steps iteratively until there is a little improvement in the objective function. Regularization parameters λ and $\boldsymbol{\gamma}$ can be learned automatically using K-fold cross-validations. We can choose the regularization parameters that minimizes the BIC criterion.

Implementation Details

1. We use position independent weights $\sqrt{\frac{N}{i*(N-i)}}$ for group lasso in equation (3) to avoid boundary effects.

2. We can see that the results are highly sensitive to the correlated graph. But we have very limited data in each partition. One approach might be to use the data from other location to estimate the correlation graph. This might be misleading the observations in each location are different.

3. To estimate, we need to use an adaptive version of the estimator in equation 3. One approach might be to estimate $\beta^\dagger$ and use this to minimize $||\beta^\dagger - \tilde{\beta}||_2 + ||\tilde{\beta}||_2$. Alternatively, we used bootstrap samples to estimate change points. Each change point is chosen with probability $\frac{\#ChangePoints\tau_k appearedinbootstrapsamples}{\#Samples}$