Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Continuous Conditional Dependency Network for Structured Regression

Chao Han
Center for Data Analytics and Biomedical Informatics, Temple University
Philadelphia, PA 19122 USA
chao.han@temple.edu

Mohamed Ghalwash
Center for Computational Health
IBM T.J. Watson Research Center
Cambridge, MA, USA
Temple University, PA, USA
Ain Shams University, Egypt, Cairo
mohamed.ghalwash@temple.edu

Zoran Obradovic
Center for Data Analytics and Biomedical Informatics, Temple University
Philadelphia, PA 19122 USA
zoran.obradovic@temple.edu

Abstract

Structured regression on graphs aims to predict response variables from multiple nodes by discovering and exploiting the dependency structure among response variables. This problem is challenging since dependencies among response variables are always unknown, and the associated prior knowledge is non-symmetric. In previous studies, various promising solutions were proposed to improve structured regression by utilizing symmetric prior knowledge, learning sparse dependency structure among response variables, or learning representations of attributes of multiple nodes. However, none of them are capable of efficiently learning dependency structure while incorporating non-symmetric prior knowledge. To achieve these objectives, we proposed Continuous Conditional Dependency Network (CCDN) for structured regression. The intuitive idea behind this model is that each response variable is not only dependent on attributes from the same node, but also on response variables from all other nodes. This results in a joint modeling of local conditional probabilities. The parameter learning is formulated as a convex optimization problem and an effective sampling algorithm is proposed for inference. CCDN is flexible in absorbing non-symmetric prior knowledge. The performance of CCDN on multiple datasets provides evidence of its structure recovery ability and superior effectiveness and efficiency as compared to the state-of-the-art alternatives.

Introduction

In many applications, like climate science and power engineering, multiple continuous variables are observed over time, which can be regarded as many independent observations of graph instances.Structured regression is proposed to predict the values of response variables of multiple nodes given their attributes, where it has been shown that discovering and exploiting the graph structure do improve the prediction. However, this problem is challenging due to the fact that the dependency among nodes is unknown, and the prior knowledge about structure is non-symmetric.

Many works have been proposed to improve regression on graphs. A simple approach is to solve regression problem for each response variable independently, where each regression problem can be solved using either linear or nonlinear model. This approach is efficient but it does not utilize the structure among nodes.

Sparse inverse covariance estimation (Banerjee, El Ghaoui, and d'Aspremont 2008), known as Graphical Lasso (Friedman, Hastie, and Tibshirani 2008), learns a sparse inverse covariance matrix of residuals by modeling the joint distribution of response variables as a multivariate Gaussian distribution. However, this generative model is not applicable to regression. This limitation is addressed by Gaussian Conditional Random Fields (GCRF) (Qin et al. 2009; Radosavljevic, Vucetic, and Obradovic 2010), which enables structured regression on graphs by modeling the conditional distribution of response variables given node attributes as a multivariate Gaussian distribution. Neural GCRF(NGCRF) (Baltrušaitis, Robinson, and Morency 2014; Radosavljevic, Vucetic, and Obradovic 2014) models nonlinear relationships between inputs and response variables by learning hidden variables via neural network. Both GCRF and NGCRF incorporate prior knowledge about graph structure in terms of a symmetric similarity matrix and, therefore, do not learn the graph structure.

Sparse Gaussian Conditional Random Fields (SGCRF) (Sohn and Kim 2012; Wytock and Kolter 2013; Yuan and Zhang 2014), as a discriminative variant of Graphical Lasso, is capable of conducting regression *and* learning sparse precision matrix simultaneously. SGCRF assumes that the relation between attributes and response variables are linear. This assumption has been relaxed in a recently developed model called Representation Learning based Structured Regression (RLSR) (Han et al. 2016), which iteratively learns the precision matrix and representation over attributes jointly to model complex relations between attributes and response. Although both SGCRF and RSLR can learn structure among response variables, they are suffering from demanding computational cost and are not able to incorporate any prior knowledge.

The dependency network, which is modeled as a cyclic Bayesian network (Heckerman et al. 2001), approximates the joint distribution of response variables as a product of multiple local conditional distributions. This allows directed graph learning and approximating local distributions separately. Unlike the Bayesian network, the graphical structure in dependency network models is not required to be a di-

rected acyclic graph. A generalization of dependency network is proposed for infering graph structure of undirected graphical models by decomposing the joint distribution as the product of multiple univariate exponential family distributions (Yang et al. 2013). Another conditional extension of dependency network, which models the conditional distribution of labels given node attributes, is successfully applied to multi-label classification problems (Guo and Gu 2011; Guo and Xue 2013). Structured Output-Associative Regression (Bo and Sminchisescu 2009) also models both input-dependency and self-dependency of outputs, but it is costly in solving its non-convex structure learning. Motivated from these aspects, in this work, we propose a flexible, effective and efficient model for structured regression with structure learning, called Continuous Conditional Dependency Network (CCDN).

The proposed CCDN model assumes that the response variable at each node is not only dependent on attributes of the same node, but is also dependent on response variables from other nodes. Then, the conditional distribution of the response variables is modeled as the product of all local univariate conditional (Gaussian) distributions, which allows regression.

Example. The intuition of CCDN is illustrated in Figure 1. The response variable y_1 is assumed to be dependent on response variables from all other nodes $\{y_2, y_3, y_4\}$ and attributes $\boldsymbol{x}_1$ at the corresponding node by modeling $P(y_1|\boldsymbol{x}_1, y_2, y_3, y_4)$, where these directed dependencies are marked as red edges.

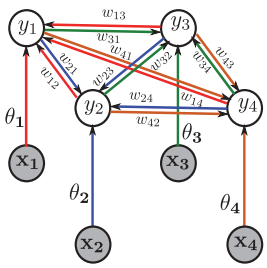


Figure 1: Graphical representation of CCDN. The directed dependencies toward y_1, y_2, y_3 and y_4 are marked as red, blue, green and orange edges, respectively.

The key contribution of this work is summarized as the following characteristics of the proposed CCDN model:

- **Flexibility:** CCDN is flexible in incorporating different types of prior knowledge, e.g., symmetric and non-symmetric, which can significantly improve the accuracy of models with insufficient training data.
- **Effectiveness:** Parameter learning is formulated as a convex optimization problem, which leads to a global optimal solution. An effective sampling algorithm is proposed as an inference algorithm.
- **Efficiency:** The joint modeling of local (independent) univariate conditional distributions and incorporation of prior knowledge makes CCDN efficient and scalable.
- **Structure Recovery:** CCDN is able to learn directed dependencies among nodes, which is a good approximation to the underlying precision matrix.

Furthermore, the effectiveness and efficiency of CCDN are demonstrated in 3 real-world applications as compared to state-of-the-art methods for structured regression.

Continuous Conditional Dependency Network

Suppose we are given m i.i.d. graphs, where each graph has p nodes. Each node has one response variable y_i and r attributes, which are denoted as $\boldsymbol{x}_i \in \mathcal{R}^r$. The response variables from all other nodes $[y_1, \cdots, y_{i-1}, y_{i+1}, \cdots, y_p]^T$ are denoted as $\bar{\boldsymbol{y}}_{\boldsymbol{i}} \in \mathcal{R}^{p-1}$. The goal of structured regression is to (1) predict response variables over all nodes, denoted as $\boldsymbol{y} \in \mathcal{R}^p$, and (2) discover the dependency among nodes. We assume graph instances are i.i.d and the structure does not change over time.

Modeling

In each graph, CCDN models the conditional probability of each response variable given its attributes and all other response variables as a univariate Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i^2)$. The pdf of the conditional probability of the i^{th} response variable is given by

$$P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i) = (2\pi\sigma_i^2)^{-1/2} \exp\{-\frac{(y_i - \mu_i)^2}{2\sigma_i^2}\}, \quad (1)$$

where $\mu_i \in \mathcal{R}$ is the expectation and $\sigma_i^2 \in \mathcal{R}^+$ is the variance.

In the Gaussian graphical model(Friedman, Hastie, and Tibshirani 2008; Banerjee, El Ghaoui, and d'Aspremont 2008), the joint distribution of $\boldsymbol{y}$ and $\boldsymbol{x}$ on one graph is assumed to be a multivariate Gaussian distribution

$$P(\boldsymbol{y}, \boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{0}, \begin{bmatrix} \Lambda_{yy} & \Lambda_{yx} \\ \Lambda_{xy} & \Lambda_{xx} \end{bmatrix}^{-1}). \quad (2)$$

The conditional Gaussian graphical model(Sohn and Kim 2012; Wytock and Kolter 2013; Yuan and Zhang 2014) was proposed based on joint Gaussian assumption from Gaussian graphical model. The conditional distribution $P(\boldsymbol{y}|\boldsymbol{x})$ is hence modeled as

$$P(\boldsymbol{y}|\boldsymbol{x}) \sim \mathcal{N}(-\Lambda_{yy}^{-1}\Lambda_{yx}\boldsymbol{x}, \Lambda_{yy}^{-1}). \quad (3)$$

Inspired by this modeling, we come up with the following theorem about local objective in CCDN.

Theorem 1. *Given the assumption from the Gaussian graphical model that the joint distribution is modeled as*

$$P(y_i, \bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i) \sim \mathcal{N}(\boldsymbol{0}, \begin{bmatrix} \Lambda_{yy} & \Lambda_{y\bar{y}} & \Lambda_{yx} \\ \Lambda_{\bar{y}y} & \Lambda_{\bar{y}\bar{y}} & \Lambda_{\bar{y}x} \\ \Lambda_{x\bar{y}} & \Lambda_{xy} & \Lambda_{xx} \end{bmatrix}^{-1})$$

the conditional distribution can be modeled as

$$P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i) \sim \mathcal{N}(-\Lambda_{yy}^{-1}(\boldsymbol{x}_i^T\Lambda_{yx} + \bar{\boldsymbol{y}}_{\boldsymbol{i}}{}^T\Lambda_{y\bar{y}}), \Lambda_{yy}^{-1})$$

Proof Sketch. By utilizing properties of multivariate Gaussian and applying simple linear algebra, the derivation

can be reached. For more details see the supplementary material. □

Theorem 1 is essential in showing how the local conditional probability of CCDN is formulated as an univariate Gaussian. Based on this theorem, $P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i)$ can be rewritten as

$$P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i) \sim \mathcal{N}(-\Lambda_i^{-1}(\boldsymbol{x}_i^T\boldsymbol{\theta}_i + \bar{\boldsymbol{y}}_{\boldsymbol{i}}{}^T\boldsymbol{w}_i), \Lambda_i^{-1}) \quad (4)$$

where $\boldsymbol{\theta}_i = \Lambda_{yx} \in \mathcal{R}^r$ models the dependency between y_i and $\boldsymbol{x}_i$, $\boldsymbol{w}_i = \Lambda_{y\bar{y}} \in \mathcal{R}^{p-1}$ models the dependency between y_i and its complement $\bar{\boldsymbol{y}}_{\boldsymbol{i}}$, and $\Lambda_i = \Lambda_{yy}$. Hence, the negative log-likelihood l_i for $P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i)$ is given by

$$\begin{aligned} l_i &= -\log P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i) \\ &= (y_i + \Lambda_i^{-1}(\boldsymbol{x}_i^T\boldsymbol{\theta}_i + \bar{\boldsymbol{y}}_{\boldsymbol{i}}{}^T\boldsymbol{w}_i))^2\Lambda_i - \log\Lambda_i + c, \end{aligned} \quad (5)$$

where c is a constant term.

Unlike GCRF (Radosavljevic, Vucetic, and Obradovic 2010) and SGCRF (Wytock and Kolter 2013), CCDN does not model the joint conditional distribution of response variables, but minimizes the product of negative log-likelihood of conditional probability for each node in the graph. The global learning objective of CCDN can be expressed as

$$\min_{\Lambda,\Theta,W} \sum_{i=1}^{p} l_i = \sum_{i=1}^{p} -\log P(y_i|\bar{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{x}_i), \quad (6)$$

where $\Lambda = [\Lambda_1, \ldots, \Lambda_p]$, $\Theta = [\boldsymbol{\theta}_1; \cdots; \boldsymbol{\theta}_p]$ and $W = [\boldsymbol{w}_1; \cdots; \boldsymbol{w}_p]$.

Example. An example of CCDN global learning objective is illustrated in Figure 1. Rather than modeling the joint conditional probability $P(y_1, y_2, y_3, y_4|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$, CCDN maximizes the product of local conditional probabilities $P(y_1|\bar{\boldsymbol{y}}_{\boldsymbol{1}}, \boldsymbol{x}_1)$, which are marked with red edges, $P(y_2|\bar{\boldsymbol{y}}_{\boldsymbol{2}}, \boldsymbol{x}_2)$ which are marked with blue edges, $P(y_3|\bar{\boldsymbol{y}}_{\boldsymbol{3}}, \boldsymbol{x}_3)$ which are marked with green edges, and $P(y_4|\bar{\boldsymbol{y}}_{\boldsymbol{4}}, \boldsymbol{x}_4)$ which are marked with orange edges. In this example, the probabilistic dependencies of each node are presented using directed edges with its corresponding parameters, i.e., $\boldsymbol{\theta}_1$ models the relation between y_1 and $\boldsymbol{x}_1$, and $\boldsymbol{w}_1 = [w_{12}, w_{13}, w_{14}]$ models the dependency between y_1 and $\bar{\boldsymbol{y}}_{\boldsymbol{1}} = [y_2, y_3, y_4]$.

Let $Y_i = [y_i^1, y_i^2, \ldots, y_i^m] \in \mathcal{R}^m$ denotes the i^{th} response variable observed over m graphs, and $\bar{Y}_i \in \mathcal{R}^{m\times(p-1)}$ and $X_i \in \mathcal{R}^{m\times r}$ denote the complementary response variables and the attributes of node i, respectively. The global learning objective of CCDN on m graphs is given by

$$\min_{\Lambda,\Theta,W} \sum_{i=1}^{p} L_i = \sum_{i=1}^{p} -\log P(Y_i|\bar{Y}_i, X_i). \quad (7)$$

Learning

Since $P(Y_i|\bar{Y}_i, X_i)$ is modeled as an univariate Gaussian distribution with positive variance Λ_i^{-1}, the parameter learning of CCDN is formulated as the a constrained optimization problem

$$\begin{aligned} &\arg\min_{\Lambda,\Theta,W} \; L = \sum_{i=1}^{p} -\log P(Y_i|\bar{Y}_i, X_i), \\ &\text{subject to} \quad \Lambda_i > 0, \text{ for } i = \{1, \cdots, p\}. \end{aligned} \quad (8)$$

Given the constrained optimization problem of CCDN in equation (8), we have the following theorem.

Theorem 2. *The parameter learning of CCDN in (8) is a convex optimization problem.*

Lemma 1. *For any symmetric matrix, M, of the form*

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad (9)$$

if C is invertible, then we have $M \succ 0$ iff $C \succ 0$ and $A - BC^{-1}B^T \succ 0$. In our problem, M is specified with

$$A = \begin{bmatrix} \frac{\partial^2 l}{\partial \boldsymbol{w}\partial \boldsymbol{w}} & \frac{\partial^2 l}{\partial \boldsymbol{w}\partial \boldsymbol{\Theta}} \\ \frac{\partial^2 l}{\partial \boldsymbol{\Theta}\partial \boldsymbol{w}} & \frac{\partial^2 l}{\partial \boldsymbol{\Theta}\partial \boldsymbol{\Theta}} \end{bmatrix}, \; B = \begin{bmatrix} \frac{\partial^2 l}{\partial \boldsymbol{W}\partial \Lambda} \\ \frac{\partial^2 l}{\partial \boldsymbol{\Theta}\partial \Lambda} \end{bmatrix} \text{ and } C = \frac{\partial^2 l}{\partial \Lambda^2}$$

Proof Sketch. By utilizing the Schur complement lemma (Gallier 2010) in Lemma 1 and the positiveness constraint in equation (8), the Hessian matrix w.r.t. parameters in each local objective of CCDN can be proved to be positive definite. Therefore the constrained optimization problem formulated in equation (8) is convex. For more details see the supplementary material. □

Given the convexity of parameter learning as stated in Theorem 2, any gradient based method can lead to a global solution. We chose to apply Quasi-Newton as our learning algorithm. The first derivatives of the local objective of the i^{th} subtask L_i with respect to Λ_i, $\boldsymbol{\theta}_i$ and $\boldsymbol{w}_i$ are given by

$$\begin{aligned} \frac{\partial L_i}{\partial \Lambda_i} &= \frac{1}{m}\{Y_i^T Y_i - \frac{(X_i\boldsymbol{\theta}_i + \bar{Y}_i\boldsymbol{w}_i)^T(X_i\boldsymbol{\theta}_i + \bar{Y}_i\boldsymbol{w}_i)}{\Lambda_i^2}\} - \frac{1}{\Lambda_i}, \\ \frac{\partial L_i}{\partial \boldsymbol{\theta}_i} &= \frac{1}{m}\{2X_i^T(X_i\boldsymbol{\theta}_i + \bar{Y}_i\boldsymbol{w}_i)\Lambda_i^{-1} + 2Y_i^T X_i\}, \\ \frac{\partial L_i}{\partial \boldsymbol{w}_i} &= \frac{1}{m}\{2\bar{Y}_i{}^T(X_i\boldsymbol{\theta}_i + \bar{Y}_i\boldsymbol{w}_i)\Lambda_i^{-1} + 2Y_i^T\bar{Y}_i\}. \end{aligned} \quad (10)$$

Time Complexity. As there are p nodes in each graph, there are $q = p^2$ pairwise dependencies among nodes to learn. If all dependencies are learned simultaneously as in most existed works, the time complexity of one iteration of a Quasi-Newton method is $O(q^2) = Q(p^4)$. However, time can be reduced to $O(p * p^2) = O(p^3)$ in CCDN because of the joint modeling of p independent local objectives. That's why CCDN is efficient in model learning. Additional evidence about efficiency is provided in experiments.

Inference

The aim of inference is to find the estimation of Y such that the product of conditional probabilities can be maximized. This goal is expressed as

$$\hat{Y} = \operatorname*{argmax}_{Y} \prod_{i=1}^{p} P(Y_i|\bar{Y}_i, X_i) \quad (11)$$

The inference is challenging in several aspects. First, $\bar{Y}_i$ is unknown in testing phase. Therefore, it is impossible to do exact inference over Y_i through $P(Y_i|\bar{Y}_i, X_i)$. Second, it is neither possible to infer Y from X since there is no closed-form expression of $P(Y|X)$ from CCDN; however, $P(Y|X)$ can be approximated via Markov Chain Monte

Algorithm 1 Iterated Conditional Mode

Input: $\{X_1, \cdots, X_p\}$, Θ, Λ and W.
Output: Estimation of response variables in the test data: $\hat{Y}$

Initialization: $t = 0$, Y^0
for $t = 1 : T^*$ **do** ▷ Stopping Criteria
  Generating a semi-random ordering $\boldsymbol{\tau}$ ▷ S.R.O
  for i in $\boldsymbol{\tau}$ **do**
    Composing $\bar{Y_i}^t$

$$\bar{Y_i}^t = \{Y_1^{T(1)}, \cdots, Y_{i-1}^{T(i-1)}, Y_{i+1}^{T(i+1)}, \cdots, Y_p^{T(p)}\}$$

where $T(j) = t$**, if** $\boldsymbol{\tau}(j) < i$**;** $T(j) = t - 1$**, otherwise.**

    Drawing Y_i^t

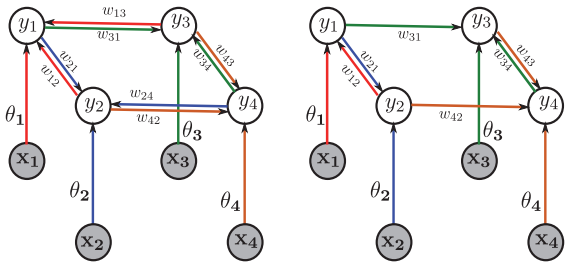
$$\begin{aligned} Y_i^t &= \underset{Y_i}{\operatorname{argmax}}\, P(Y_i|\bar{Y_i}^t, X_i, \boldsymbol{\theta}_i, \Lambda_i, \boldsymbol{w}_i) \\ &= -\Lambda_i^{-1}(X_i\boldsymbol{\theta}_i + \bar{Y_i}^t\boldsymbol{w}_i)) \end{aligned}$$

  end for
end for
$\hat{Y} = Y^{T^*}$

Carlo algorithms if the posterior probabilities for each variable are known, i.e. $P(Y|X)$ is unknown but $P(Y_i|\bar{Y}_i, X_i)$ is known. Gibbs sampling is hence a natural choice in this case. However, Gibbs Sampling requires extensive iterations to achieve a good estimation due to the uncertainties, which makes it inefficient. In this paper, we adapt Iterated Conditional Mode(Monaco, Viswanath, and Madabhushi 2009) as a simpler and more efficient substitute for Gibbs sampling, based on the fact that univariate Gaussian distribution has closed form expression for its only mode. In each iteration of Iterated Conditional Mode, rather than sampling instances from a univariate Gaussian distribution, each instance is assigned exactly as the expectation of corresponding Gaussian distribution $E(Y_i|\bar{Y}_i, X_i)$. In this way, the uncertainties in sampling are avoided, and the times of simulation can be saved. The product of probabilities $\prod_{i=1}^{p} P(Y_i|\bar{Y}_i, X_i)$ is also guaranteed to be maximized. A detailed description of Iterated Conditional Mode is presented in Algorithm 1.

Stopping Criteria The goal is to find the index of iteration with draws that are closest to the ground truth when testing, which is not easy. Thus, we find the best draw on validation dataset. Then, the index of iteration with the best draw is denoted as T^*.

Semi-randomized Ordering (S.R.O) Semi-randomized ordering is a strategy used to generate the ordering for sampling Y in each iteration of Iterated Conditional Mode. When there is no prior knowledge (see next section) available or the prior knowledge is symmetric, semi-randomized ordering generates a fully randomized permutation over variables Y. For example, the ordering generated for symmetric prior knowledge, which is illustrated at Figure 2a, is $\{randperm(y_1, y_2, y_3, y_4)\}$. When the prior knowledge is non-symmetric, there might exists directed dependency between different subsets of response variables. For example, the ordering for non-symmetric dependency illustrated at Figure 2b is $\{randperm(y_1, y_2)\}, \{randperm(y_3, y_4)\}$, where variables of the same group are permuted randomly, but groups are permuted in order.



(a) CCDN-S (b) CCDN-N

Figure 2: Incorporation of prior knowledge: (a) CCDN with symmetric prior knowledge (CCDN-S) and (b) CCDN with non-symmetric prior knowledge (CCDN-N).

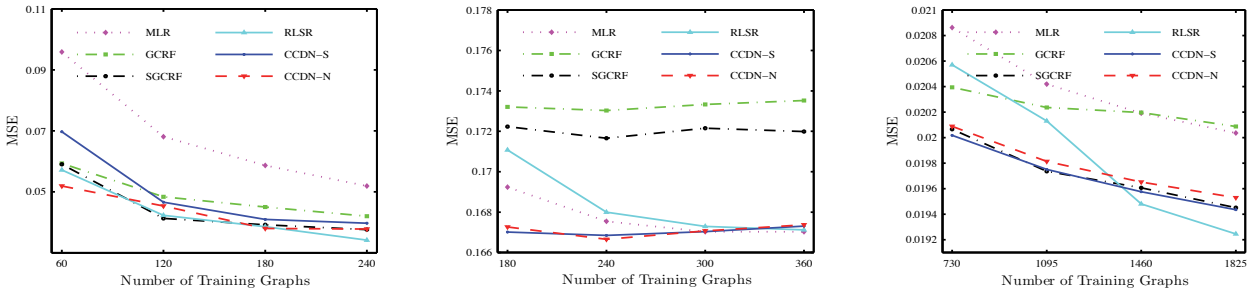
Incorporation of Prior Knowledge

Prior knowledge about graph structure is important to structured regression because the valuable information within it is always available but not easy to utilize. Effective exploitation of prior knowledge can help the model to generalize well, even with limited data for training. **In this section, we explain how CCDN is flexible in incorporating various types of prior knowledge.**

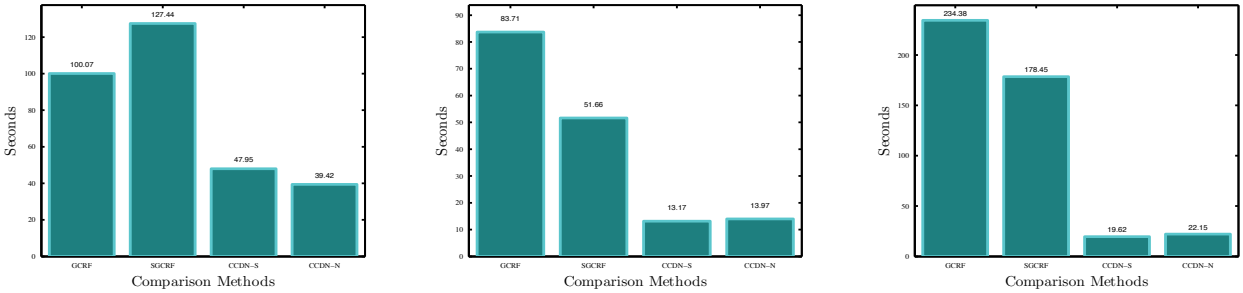
In this work, prior knowledge is defined as the presence of directed dependencies among response variables in a single graph. In a graph with p nodes, the prior knowledge about y_i toward all other response variables $\bar{\boldsymbol{y}}_{\boldsymbol{i}}$ is encoded as a binary column vector $\boldsymbol{s}_i \in \{0, 1\}^{p-1}$. Particularly, if y_i is independent to y_j in the prior knowledge, then $s_{ij} = 0$. Otherwise $s_{ij} = 1$. The prior knowledge $\boldsymbol{s}$ is incorporated in CCDN by $\boldsymbol{w}_i = \boldsymbol{s}_i \circ \boldsymbol{w}_i$, where the $\circ$ operator is the element-wise product. Generally, the prior knowledge can be represented as real numbers, which weight the belief of directed dependencies. But in this study, we only consider the binary representation.

As we know, the prior knowledge about dependency structure may existed as either a symmetric matrix or a non-symmetric matrix. The symmetric dependency is defined as, $\forall i, j, i \neq j, s_{ij} = 1 \Rightarrow s_{ji} = 1$. The non-symmetric dependency is defined as $\forall i, j, i \neq j, s_{ij} = 1 \nRightarrow s_{ji} = 1$.

Example. An example of symmetric prior knowledge is illustrated in Figure 2a, where directed dependencies with opposite directions always exist together. For example, we have $s_{21} = s_{12} = 1$, $s_{31} = s_{13} = 1$, $s_{24} = s_{42} = 1$ and $s_{34} = s_{43} = 1$. The CCDN model with symmetric dependency prior knowledge is referred as **CCDN-S**. The non-symmetric prior knowledge is illustrated in Figure 2b, where we assume y_3 is dependent on y_1, and y_4 is dependent on y_2, but not vice versa. Namely, directed dependencies with opposite directions are not necessary to exist simultaneously. This is why we have $s_{31} = 1, s_{42} = 1$, but $s_{13} = 0$ and $s_{24} = 0$. The CCDN model with non-symmetric dependency is referred as **CCDN-N**.



(a) Effectiveness evaluation on Wind Data (b) Effectiveness evaluation on Precipitation Data (c) Effectiveness evaluation on Energy Data



(d) Efficiency comparison on Wind Data. RLSR spent more than 5000 sec. (e) Efficiency comparison on Precipitation Data. RLSR spent more than 3000 sec. (f) Efficiency comparison on Energy Data. RLSR spent more than 4500 sec.

Figure 3: Performance comparison of all baselines on 3 datasets.

Experimental Results

In this section, we empirically demonstrate the characteristics of CCDN, including **effectiveness**, **efficiency** and **structure recovery**. The benefits brought by **flexibility** are validated in experiments about **efficiency** and **effectiveness** .

Real datasets

We used 3 real-world datasets in our experiments for structured regression. The brief description of datasets are as following. For more detailed description see the supplementary material. **Wind Power Forecasting.** Wind power data is obtained from the Global Energy Forecasting 2012 competition[1]. The task is to predict hourly wind power at 7 nearby wind farms for the next 24 hours. Each graph has 168 nodes and each node has 4 attributes. Both symmetric and non-symmetric prior knowledge assume that each farm is dependent on other farms at same time point. Symmetric prior knowledge also assumes that each farm is dependent on same farm from neighbouring time points. But non-symmetric prior knowledge assumes that each farm is only dependent on the farm of previous time point.**Precipitation Forecasting.** The task is to forecast daily precipitation across multiple locations based on several climatological features. It can be downloaded from NOAA's NCDC website[2]. The observation from each month is modeled as a graph with 124 nodes and each node has 9 attributes. **Solar Energy Forecasting.** The task is to predict the daily solar energy income at 98 Oklahoma Mesonet sites. The data is available on kaggle [3]. Each day is modeled as a graph with 98 nodes with each node having 19 features. On both precipitation and energy datasets, symmetric prior knowledge connects two stations if they are among 4 nearest neighbors of each other. Non-symmetric prior knowledge connect each station to its 4 nearest neighbors.

Comparison Methods. We compare our proposed models (CCDN-S and CCDN-N) to the following structured regression models:

- **MLR (Multiple Linear Regression).** Building independent linear regression models for different response variables.
- **GCRF:** Gaussian Conditional Random Fields that is developed in (Radosavljevic, Vucetic, and Obradovic 2010), which exploits prior knowledge about similarity among response variables for structured regression without learning structure.
- **SGCRF:** Sparse Gaussian Conditional Random Fields that are developed in (Sohn and Kim 2012; Wytock and Kolter 2013; Yuan and Zhang 2014). We only compare to the most recent work (Wytock and Kolter 2013). The λ of SGCRF is picked from $\{1e-4, 1e-3, 1e-2, 0.1, 1\}$ via 8 folds cross-validation.
- **RLSR:** A structured regression model that jointly learns representation over attributes and structure among re-

[1]https://www.kaggle.com/c/GEF2012-wind-forecasting
[2]http://www.ncdc.noaa.gov/
[3]https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest

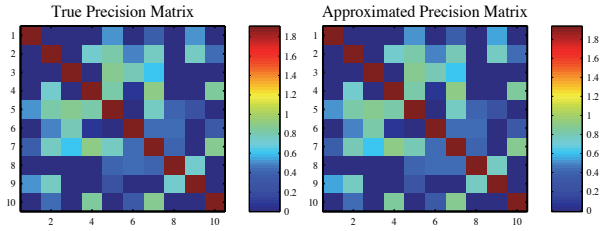
sponse variables. (Han et al. 2016). The λ of RLSR is tuned the same way as in SGCRF. The neural network is set with one hidden layer with 20 hidden neurons.

Effectiveness Evaluation

In the experiments, we evaluated the effect of increasing the training size on the predictive accuracy for all methods. Each experiment is evaluated in terms of mean square error (MSE) over 8 consecutive windows. The result on **Wind Data** is presented in Figure 3a. We considered 4 different training sizes $r = \{60, 120, 180, 240\}$, and fix validation and testing sizes as 60. We noticed that CCDN-N performs the best when training data is limited. This is because incorporating prior knowledge removes useless information, and thus helps the model to generalize better. RLSR and SGCRF perform comparable to CCDN-N when more training data is available, but with the sacrifice of efficiency. The better performance of CCDN-N over CCDN-S, also reflects that the non-symmetric prior knowledge on wind data makes more sense. The results on **Precipitation Data** are shown in Figure 3b. We set both validation and testing size as 24. Our proposed methods are still performing the best with limited data. In addition, the prior knowledge imposed on this data is sparse. It may explain why structure models don't work well, but independent models perform better. The results of **Energy data** are presented in Figure 3c, where we show the effect of training with large amount of data. In the experiment, both validation size and testing size are set as 365. As we can see, CCDN-S slightly outperforms SGCRF with less data, but it is outperformed by RLSR with more and more data added. It still meets our expectation, because all models are supposed to fully converge with sufficient training data. Therefore, we can conclude that proposed models with prior knowledge can generalize better with limited data, and they are also comparable with the best alternatives when data is sufficient. Most importantly, we will see that proposed model is simultaneously effective and efficient, while other models always need demanding computational cost.

Efficiency Evaluation

In the experiments, we demonstrated the efficiency of structure learning by evaluating the learning time of all methods when they achieved best performance. MLR is not compared, because it is fast but not effective as evident in previous experiments. RLSR is not reported, because it spent at least 3000 seconds on all dataset. The experiments on different data were running on different nodes of cluster. The settings are same as previous experiment. In **Wind Data** (Figure 3d), both CCDN-S and CCDN-N are the fastest models. GCRF is slow because the running time increases linearly with the increment of training sizes. The efficiency evaluation on **Precipitation Data** is shown Figure 3e. CCDN-N and CCDN-S are more than 4 times faster than other models. The results on **Energy Data** are shown in Figure 3f. In this experiment, CCDN-N and CCDN-S also outperforms other models. SGCRF takes more time because the learned structure is dense. Although SGCRF performed comparable to proposed models in terms of effectiveness, it is about 9 times slower. The results showed that our proposed models are significantly faster than other alternatives. The benefit in terms of efficiency, brought by joint modeling and incorporation of prior knowledge, is hence demonstrated. The efficiency of proposed models can be further improved if all local probabilistic models are learned in parallel. The time consumption of inference is not reported, because it converges within less than 0.1 second in each repetition in average.



(a) (b)

Figure 4: Visualization of (a) true precision matrix and (b) learned precision matrix from CCDN.

Structure Recovery on synthetic data

Precisely recovering structure is challenging when structures are learned from multiple independent models. In this section, we want to demonstrate that CCDN is also able to discover structure. From Theorem 1, we know that learned w is actually an approximation of Λ_{yy}. However, as w_{ij} and w_{ji} are learned from different local distributions, how to guarantee that they have similar values and be close to $[\Lambda_{yy}]_{ij}$ and $[\Lambda_{yy}]_{ji}$ of true precision matrix? Therefore we designed an experiment on synthetic data to answer this question. In this experiment, we generated 1600 independent graphs based on equation (3), with 1000 used for training, 300 for validation and the 300 for testing. For the simplicity of visualization. We generated 10 response variables per graph, and 3 attributes per node. $\Lambda_{yy} \in \mathcal{R}^{10\times10}$ and $\Lambda_{yx} \in \mathcal{R}^{10\times30}$ are generated with random structure. We applied CCDN-S with symmetric prior knowledge as fully connected graph for learning. The true precision matrix is visualized in Figure 4a, and the learned precision matrix is visualized in Figure 4b. Obviously, we can see that the learned precision matrix (w) is close to the true precision matrix (Λ_{yy}). The Euclidean distance between two matrices is as small as $9e-4$. Therefore, we can conclude that: (1) any pair of directed dependencies with opposite directions, e.g., w_{ij} and w_{ji}, are similar to each other. (2) The estimated precision matrix, even though they are learned from different local models, can still accurately approximate the true precision matrix

Conclusion

We proposed continuous conditional dependency network for structured regression from theory to practice. It is flexible in incorporating different types of prior knowledge, efficient in model learning and inference, effective on structured regression and able to recover dependency structure. All experiments provide evidence that the proposed method have

better or at least comparable performance than other structure regression models in terms of MSE, but our proposed models is superior as it is flexible, always efficient and able to precisely discover structure.

Acknowledgments. The authors gratefully acknowledge the support of Defense Advanced Research Project Agency (DARPA)GRAPHS program under Air Force Research Laboratory (AFRL) prime contract no. FA9550-12-1-0406, National Science Foundation BIGDATA grant 14476570 and Office of Naval Research Mathematics of Data Science Project N00014-15-1-2729.

References

Baltrušaitis, T.; Robinson, P.; and Morency, L.-P. 2014. Continuous conditional neural fields for structured regression. In *European Conference on Computer Vision*, 593–608. Springer.

Banerjee, O.; El Ghaoui, L.; and d'Aspremont, A. 2008. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research* 9:485–516.

Bo, L., and Sminchisescu, C. 2009. Structured output-associative regression. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2403–2410. IEEE.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3):432–441.

Gallier, J. 2010. The schur complement and symmetric positive semidefinite (and definite) matrices. *Technical report, Penn Engineering*.

Guo, Y., and Gu, S. 2011. Multi-label classification using conditional dependency networks. In *IJCAI Proc. International Joint Conf. on Artificial Intelligence*.

Guo, Y., and Xue, W. 2013. Probabilistic multi-label classification with sparse feature learning. In *IJCAI Proc. International Joint Conf. on Artificial Intelligence*.

Han, C.; Zhang, S.; Ghalwash, M.; Vucetic, S.; and Obradovic, Z. 2016. Joint learning of representation and structure for sparse regression on graphs. In *SDM Proc. SIAM Intl Conf. Data Mining*.

Heckerman, D.; Chickering, D. M.; Meek, C.; Rounthwaite, R.; and Kadie, C. 2001. Dependency networks for inference, collaborative filtering, and data visualization. *The Journal of Machine Learning Research* 1:49–75.

Monaco, J.; Viswanath, S.; and Madabhushi, A. 2009. Weighted iterated conditional modes for random fields: Application to prostate cancer detection. *Program Committee John Ashburner (University College London) Sylvain Bouix (Harvard Medical School) Tim Cootes (University of Manchester)* 209.

Qin, T.; Liu, T.-Y.; Zhang, X.-D.; Wang, D.-S.; and Li, H. 2009. Global ranking using continuous conditional random fields. In *Advances in Neural Information Processing Systems*, 1281–1288.

Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2010. Continuous conditional random fields for regression in remote sensing. In *ECAI*, 809–814.

Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2014. Neural gaussian conditional random fields. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 614–629. Springer.

Sohn, K.-A., and Kim, S. 2012. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *International Conf. on Artificial Intelligence and Statistics*, 1081–1089.

Wytock, M., and Kolter, Z. 2013. Sparse gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *Proc. International Conf. on Machine Learning (ICML-13)*, 1265–1273.

Yang, E.; Ravikumar, P. K.; Allen, G. I.; and Liu, Z. 2013. Conditional random fields via univariate exponential families. In *Advances in Neural Information Processing Systems*, 683–691.

Yuan, X.-T., and Zhang, T. 2014. Partial gaussian graphical model estimation. *Information Theory, IEEE Transactions on* 60(3):1673–1687.