
Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30005212

Every reasonable effort has been made to ensure that permission has been obtained for items included in Deakin Research Online. If you believe that your rights have been infringed by this repository, please contact drosupport@deakin.edu.au

Copyright : 2003, InTech
Ensemble Parameter Estimation for Graphical Models

Yiqing Tu, Gang Li, and Honghua Dai

School of Information Technology, Deakin University,
221 Burwood Highway, Vic 3125, Australia
{ytu,gangli,hdai}@deakin.edu.au

Abstract. Parameter Estimation is one of the key issues involved in the discovery of graphical models from data. Current state of the art methods have demonstrated their abilities in different kind of graphical models. In this paper, we introduce ensemble learning into the process of parameter estimation, and examine ensemble parameter estimation methods for different kind of graphical models under complete data set and incomplete data set. We provide experimental results which show that ensemble method can achieve an improved result over the base parameter estimation method in terms of accuracy. In addition, the method is amenable to parallel or distributed processing, which is an important characteristic for data mining in large data sets.

1 Introduction

Graphical Model is a graphical representation for probability distributions of the domain, and it is arguably the representation of choice for uncertainty in artificial intelligence [1]. Due to its compact and concise semantic, it has been successfully applied in many fields such as diagnosis, expert systems, and decision making systems. However, the manual construction of Graphical Model is usually time-consuming and subject to mistakes. Therefore, algorithms for automatic construction, that occasionally use the information provided by an expert, can be of great help [2]. As Graphical Model can often be plausibly understood as describing causal relations, the automatic induction of Graphical Model is usually referred as Causal Discovery.

A graphical model consists of two components — the model structure and the parameters. Therefore, the task of learning graphical models from data can usually be formulated as a search over the space of candidate structures, and the subsequent estimation of parameter given the fixed structure. The first task is often referred as structure learning, while the second task as parameter learning or parameter estimation.

In a number of real world applications the model structure is easier to obtain than the parameters, particularly when working with a domain expert. This is because human experts often have the ability to describe the qualitative correlations in a domain but they usually find it harder to give the exact parameter values.
In this paper, our focus area is the parameter learning task of directed graphical models. Suppose we are given a model structure and a training data set (complete or incomplete) in which the training examples are independent and identically distributed, our goal is to use the data set to estimate the optimal model parameters.

The rest of this paper is organized as follows. In section 2 we briefly introduce related concepts and corresponding standard parameter estimation algorithms. In section 3 we present the ensemble strategies for parameter estimation, under complete data set and incomplete data set. In section 4 those ensemble parameter estimation methods are compared with the base parameter estimation algorithms, and experimental results are given and analyzed. Finally, we conclude this paper in section 5.

2 Background

2.1 Graphical Model and its Parameters

In many knowledge systems, random variables are used to represent events or objects. By making various instantiations to these variables, the current state of the system can be modeled. Thus, this will involve computing joint probabilities of all these random variables. The key idea of graphical model is the explicit representation of conditional (in)dependencies among random variables by a graph, such that the joint distribution on the domain can be represented in a more concise form.

Let a finite set $U = \{V_1, V_2, \ldots, V_n\}$ be a domain with $n$ random variables, and lowercase letters such as $v_1, \ldots, v_n$ denote specific values taken by those variables. Formally, a Graphical Model is the pair $GM = \langle S, \Theta_S \rangle$, where

Model Structure $S$ is a graph whose nodes correspond to the random variables in $U$, and whose edges represent (in)dependencies among these random variables $^1$. Model structure represents the assertions that: When all the parents of a variable $V_i$ are given, $V_i$ is conditionally independent $^2$ of the remaining variables which are not descendants of $V$. Thus, the full joint distribution on $U$ is given by the following factorization:

$$p(V_1, \cdots, V_n) = \prod_{i=1}^{n} p(V_i | Pa_S^i) \quad (1)$$

where $Pa_S^i$ denote the parents of $V_i$ in structure $S$. So, if given the model structure $S$ in order to completely specify the joint distribution of the domain, we only need to specify the conditional probability distributions at each node.

$^1$ In this paper, we limit our model to the Directed Graphical Models, whose structure $S$ consists of nodes and directed edges connected together to form a Directed Acyclic Graph (DAG).

$^2$ A more general criterion on this is the notation of $d$-Separation [1]
**Model Parameters** \( \Theta_S \) represents the set of parameters that quantifies the graphical model with a structure \( S \), and it is the set of parameters at each node, i.e. \( \Theta_S = \{ \Theta_1, \ldots, \Theta_n \} \). According to the nature of the domain, the model parameter can have different forms, and correspondingly there exists at least two kinds of graphical models:

1. In social sciences, there is a class of limited graphical model, usually referred as **Linear Causal Models** \([3, 4]\), in which all random variables are continuous. In **Linear Causal Models**, effect variables are strictly linear functions of exogenous variables. **Linear Causal Model** permits the modelling of complex multivariate phenomena, whereby measurement errors are accounted for both the dependent and explanatory variables. The relation between \( V_i \) and its parents \( Pa^S_i \) can be described by a linear function,

\[
V_i = \sum_{k=1}^{K_i} \alpha_k \times Pa^S_{ik} + R_i
\]  

(2)

Where \( K_i \) is the number of parents for node \( V_i \), \( \{ \alpha_1, \ldots, \alpha_{K_i} \} \) are path coefficients, and \( R_i \) is assumed to be identically distributed following a **Gaussian** distribution with zero mean and a standard deviation that will also be treated as an adjustable parameters, that is \( R_i \sim N(0, \sigma_i^2) \), so the set of local parameters \( \Theta_i \) for a variable \( V_i \) with continuous parents is \( \{ \sigma_i^2, \alpha_1, \ldots, \alpha_{K_i} \} \). This linear function can be written as a **Gaussian** distribution of variable \( V_i \), conditioned on its parents:

\[
p(V_i | Pa^S(V_i)) = N(\sum_{k=1}^{K_i} \alpha_k \times Pa^S_{ik}, \sigma_i^2)
\]

(3)

On the other hand, if the continuous variable \( V_i \) has no parents, we assume it as a random sample from a **Gaussian** distribution,

\[
p(V_i) = N(\mu_i, \sigma_i^2)
\]

(4)

where \( \mu_i \) is the mean value of node \( V_i \), so the set of parameters \( \Theta_i \) for a continuous variable \( V_i \) without parents is \( \{ \mu_i, \sigma_i^2 \} \).

2. When all the random variables are discrete, the graphical models are often called **Bayesian Networks**, in which the local parameters are given by \( \Theta_i = (\theta_{ijk})^q_{i=1,j=1} \), where the parameter \( \theta_{ijk} \) represents the conditional probability of variable \( V_i \) being in its \( k \)-th state, given the set of its parent variables is in its \( j \)-th state. In a **Bayesian Network**, the local parameters are often represented as a set of Conditional Probability Tables.

Thus, we can summary different kind of parameters in A summary of the local parameters is shown in Table 1.
Table 1. Summary of Local Parameters

<table>
<thead>
<tr>
<th>Local Distribution</th>
<th>Local Parameter $\theta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian without parents</td>
<td>${\mu_i, \sigma_i}$</td>
</tr>
<tr>
<td>Gaussian with $k_i$ parents</td>
<td>${\alpha_0, \alpha_1, \ldots, \alpha_{k_i}, \sigma_i}$</td>
</tr>
<tr>
<td>Multinomial</td>
<td>${\theta_{ij1}, \ldots, \theta_{ijr_i}}^*$</td>
</tr>
</tbody>
</table>

* means for each configuration $j$ of discrete parents

2.2 Related Parameter Estimation Algorithms

A graphical model can be parameterized by specifying all the local Conditional Probability Distributions at each node, i.e. deciding the value of local parameters at each node. Given the model structure $S$, how can parameters be learned from data? There are several variants of this question. The involved model can be linear causal model or Bayesian network, and the training data set can be complete or incomplete.

Complete Data Set The case of complete data set is fairly straightforward.

ML & MAP estimation For Bayesian network, a common approach is Maximum Likelihood (ML) estimation, which in the case of no hidden variables, reduces to a function of the relative frequencies of occurrence of the values of the variable [5].

Simple Bayesian updating extends Maximum Likelihood estimation by regarding the parameters as random variables, whose prior distribution represents the observer’s belief about the parameters before observing any data. Given the training data set, the prior density $p(\theta)$ is updated in the posterior density by Bayes Theorem:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)}$$

(5)

The estimate of $\theta$ is then the posterior expectation of $\theta$, and this kind of parameter learning is known as Maximum A Posterior (MAP) estimation. If mutually independencies among parameters, and Dirichlet prior distribution for parameter $\theta$ are assumed, the posterior distribution for $\theta$ can be easily calculated [6].

MML estimation For a linear causal model, a standard parameter estimation algorithm is the Minimum Message Length (MML) based estimation [7, 8]. For a node without parents, by minimizing the total encoding length, the estimate of its parameters will be calculated as

$$\mu_i = \frac{\sum_{t=1}^{T} v_{it}}{T}$$

(6a)

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} (v_{it} - \mu_i)^2}{T-1}$$

(6b)
Where $T$ is the sample size, $v_{it}$ is the value of variable $V_i$ in the $t$-th instance. While for a node with $K$ parents, the MML estimations of $\{\alpha_0, \ldots, \alpha_K\}$ are the same as the estimates by least squares estimation, and the estimation for $\sigma_i^2$ is

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} (v_{it} - \sum_k \alpha_k P_a^{i}_{kt})^2}{T - K} \quad (7)$$

**Incomplete Data** However, when the data set is incomplete, parameter learning can be more complex. Under the assumption that missing data items are Missing At Random (MAR), one of the best known techniques to deal with incomplete data is the EM algorithm [9]. Lauritzen uses this algorithm to find local optimal ML or MAP estimates for Bayesian networks [10], where an Expectation step tries to complete the data set by inferring the missing data item from available information, and then a Maximization step is called to learn the optimal parameters from the completed data set. These two steps iterated until convergence. The Maximization step can only take a step towards the direction of correct parameter space, rather than finding the optimal, it is referred as the generalized EM algorithm [11].

Another popular parameter learning algorithm is the use of a stochastic approximation of the posterior distribution using Markov Chain Monte Carlo (MCMC) methods, such as Gibbs Sampling. This method treats the missing data item as unknown parameters, and for each missing data item, a value is sampled from the conditional distribution of the corresponding variables, given all the parameters and available data. The sampling iterated until convergence, and then each missing value could be replaced by some imputed value, and parameter estimation can be carried out on the imputed complete data set.

### 3 Ensemble Parameter Learning Algorithm

Ensemble learning is a learning paradigm where multiple base learners are trained for the same task, and the outcomes of these base learners are combined for dealing with future instances. Since an ensemble is often more accurate than its base learners\[12, 13, 14\], such a paradigm has become a hot topic of supervised learning and has already been successfully applied to optical character recognition\[15, 16\], scientific image analysis\[17, 18\], medical diagnosis \[19, 20\], seismic signals classification\[21\], etc. In the area of causal discovery, Dai et al. introduced ensemble method into structure learning of linear causal models to improve the learning accuracy \[22\], and Friedman et al. used a Bootstrapping approach to discover some causal features from a sparse data set \[23\], both are related to structure learning. Up to the knowledge of the authors, the research of ensemble learning are mainly focused on supervised learning, and only a few work mentioned above are involved in the causal structure discovery, yet no work has addressed the issue of ensemble parameter estimation although this may not only generate accurate parameters but also extend the usability of ensemble learning methods.
In general, an ensemble is built in two steps, that is, obtaining multiple base learners and then combining what they learn, and three issues involved are:

**Base Learner** Base Learner is an individual learning process and it is the building block of ensemble learning algorithm. Normally, it needs to be computationally efficient while learning accuracy may be not so perfect, but with good learning performance potentials while they are ensembled.

**Ensemble Strategy** Ensemble strategy is the most important issue in ensemble learning. It decides how to generate data sets from the original data set, and how to carry out individual learning using base learner.

**Integration** The result from individual learning process is usually a set of individual models, together with weights of each model. Integration will decide how to use these individual models to produce a final result.

In this section, we considered ensemble method for parameter estimation of graphical models, given complete or incomplete data set.

### 3.1 The Base Learner

The **Base Learner** is the building block in ensemble learning methods. As for parameter estimation of graphical models, all those algorithms mentioned in section 2.2 can be used as base learner.

### 3.2 The Ensemble Strategy

The ensemble strategy deals with how to generate training data set for each individual base learners. According to the nature of training data, we use two different ensemble strategy.

**Bagging for Complete Data Set** When the original training data set contains no missing values, we adopt the Bagging to generate an ensemble. Bagging (Bootstrap Aggregating) is proposed by Breiman [12], and its primary idea is to generate an ensemble of individual models with each from a bootstrap sample [24] of the original training data set.

Given an ensemble size $M$ and a training data set $D$ consisting of $T$ instances, our implementation of Bagging, generates $M - 1$ bootstraps samples with each being created by uniformly sampling $T$ instances from $D$ with replacement, then it learns parameters from each bootstrap sample, another parameter learning results is estimated from the original training data set. Therefore, if given the graphical model structure and the original data set $D$, we can finally get $M$ different sets of parameter estimation.

**Multiple Imputation for Incomplete Data Set** When the original training data set is incomplete, we adopt the Multiple Imputation to generate an ensemble. Multiple Imputation is proposed by Rubin [25], and as its name suggests,
multiple imputation replaces each missing value by a vector composed of $M \geq 2$ possible values. The $M$ values are ordered in the sense that the first components of the vectors for the missing values are used to create one completed data set, the second components of the vectors are used to create the second complete data set and so on. It, then, estimates parameters from each of those $M$ imputed data set. Therefore, we can also get $M$ sets of parameter estimation, given the incomplete data set, and the model structure.

Special computational techniques are needed to create multiple imputation for incomplete data set. In this paper we adopt MCMC method to generate multiple imputation. MCMC method is closely related to the EM algorithm, while it does this in a stochastic fashion. It first performs a random imputation of missing data under assumed values of model parameters, and then revise the parameter estimation based on the observed and imputed data. This procedure creates a Markov chain that eventually converges to a predictive distribution, that one needs to draw from to create proper multiple imputations. Once we have determined that it has converged by $k$ cycles, we can perform $M$ runs of $k$ cycles and save the completed data set from the end of each run as our $M$ imputations. Refer to [26] for a detailed discussion of this method.

3.3 Integration

With $M$ bootstrap or imputed data set, all parameter estimation algorithms for complete data set, including ML&MAP estimation, and MML estimation, can all be used to learn the parameters for the given model structure.

For a Bayesian network, let $\hat{\theta}_{ijk}^t$ be the estimates of one parameter from $t$-th base learner, $t = 1, 2, \ldots, M$, then the estimate for $\theta_{ijk}$ is the average of the $M$ estimates:

$$\bar{\theta}_{ijk} = \frac{1}{M} \sum_{t=1}^{M} \hat{\theta}_{ijk}^t$$

(8)

For a linear causal model, let $\hat{\mu}_i^t, \hat{\alpha}_k^t, \hat{\sigma}_i^t$ be some estimation of involved coefficients and variation from the $t$-th base learner, the final estimation of these parameters are the average of the $M$ estimates:

$$\mu_i = \frac{1}{M} \sum_{t=1}^{M} \hat{\mu}_i^t$$

(9a)

$$\alpha_k = \frac{1}{M} \sum_{t=1}^{M} \hat{\alpha}_k^t$$

(9b)

$$\sigma_i = \frac{1}{M} \sum_{t=1}^{M} \hat{\sigma}_i^t$$

(9c)

4 Experimental Result

In the previous section we have described an ensemble algorithm for parameter estimation of graphical models from training data set. In this section we report on experiments designed to answer this question: Can the ensemble algorithm yield better parameter estimation than the base algorithm?
4.1 Examined Graphical Models

Two experiments were conducted with synthetic data sampled from some known graphical models, including 4 linear causal models, and 4 Bayesian network. The Information on these models used in our experiments is tabulated in Table 2.

<table>
<thead>
<tr>
<th>Graphical Model</th>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiji</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Evans</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Blau</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Case9</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>Asia</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Cancer</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Burglary</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Sprinkle</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

In experiment 1, 1000 instances were sampled from each of the above graphical model. Firstly, ML estimation or MML estimation was used to learn the parameters from each data set given corresponding model structure. Then, our ensemble algorithm was used to estimate the parameters from each of the training data set.

In experiment 2, 1000 instances were sampled from each of the above models, then 20% of data values are removed in a purely random fashion, so for each model, we can get an incomplete data set. Firstly, EM algorithm was called to estimate the parameters for each of the model, then our ensemble algorithm was used to learn the parameters.

We compare the learned parameters with the original generative models in terms of the following two kinds of metrics:

**Average Error** For linear causal model, the average error of path coefficients is used to evaluate the learned result: 

\[ AE = \frac{1}{|E|} \sum_{e \in E} |\alpha_e - \hat{\alpha}_e|, \]

where \(|E|\) is the number of edges, while \(\alpha_e\) and \(\hat{\alpha}_e\) are the known and estimated path coefficient of the \(e\)-th edge, respectively.

**KL Divergence** For Bayesian network, the Kullback-Leibler (KL) divergences between the joint distributions of the original model and the learned model are used to evaluate the parameter estimation performance.

4.2 Result and Analysis

Our experiments are implemented in MATLAB with Bayes-Net ToolBox (BNT) [27]. In all these experiments, the ensemble size was set at 5, and the iteration step for EM algorithms was set at 5. For each experiment, we perform 10 runs on each data set, and then record the average results by each algorithm.
### Table 3. Experiment 1: Average Errors

<table>
<thead>
<tr>
<th>Model</th>
<th>by MML</th>
<th>by Ensemble MML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiji</td>
<td>0.025</td>
<td>0.021</td>
</tr>
<tr>
<td>Evans</td>
<td>0.026</td>
<td>0.024</td>
</tr>
<tr>
<td>Blau</td>
<td>0.022</td>
<td>0.016</td>
</tr>
<tr>
<td>Case9</td>
<td>0.016</td>
<td>0.015</td>
</tr>
</tbody>
</table>

### Table 4. Experiment 1: KL Divergence

<table>
<thead>
<tr>
<th>Model</th>
<th>by ML</th>
<th>by Ensemble ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>1.630</td>
<td>1.625</td>
</tr>
<tr>
<td>Cancer</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Burglary</td>
<td>0.043</td>
<td>0.044</td>
</tr>
<tr>
<td>Sprinkle</td>
<td>0.002</td>
<td>0.002</td>
</tr>
</tbody>
</table>

### Table 5. Experiment 2: Average Errors

<table>
<thead>
<tr>
<th>Model</th>
<th>by EM</th>
<th>by Ensemble MML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiji</td>
<td>0.020</td>
<td>0.020</td>
</tr>
<tr>
<td>Evans</td>
<td>0.024</td>
<td>0.021</td>
</tr>
<tr>
<td>Blau</td>
<td>0.022</td>
<td>0.017</td>
</tr>
<tr>
<td>Case9</td>
<td>0.016</td>
<td>0.014</td>
</tr>
</tbody>
</table>

### Table 6. Experiment 2: KL Divergence

<table>
<thead>
<tr>
<th>Model</th>
<th>by EM</th>
<th>by Ensemble ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>1.113</td>
<td>1.106</td>
</tr>
<tr>
<td>Cancer</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Burglary</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>Sprinkle</td>
<td>0.008</td>
<td>0.007</td>
</tr>
</tbody>
</table>
For experiment 1, in which all training data sets are complete, the results are given in Table 3 and Table 4. From Table 3 we can see that for parameter estimation of linear causal models, ensemble method can yield more accurate parameters than the original MML estimation. For Bayesian networks, from Table 4 we can see that ensemble method outperforms the ML estimation in 3 out of 4 data sets.

For experiment 2, in which all training data sets are incomplete, the results are given in Table 5 and Table 6. From these two tables, we can see that ensemble method can yield more accurate parameters than the EM algorithm, for linear causal models and for Bayesian networks. It is interesting to note that for parameter estimation, some parameter estimation results from incomplete data set can be better than the results from complete data set (Refer to the model Asia, Burglary).

From these experiments results, we can safely conclude that ensemble learning method can improve the accuracy of parameter estimation algorithms.

5 Conclusions
Achieving highest accuracy is always one of the essential goal of almost all the research done in the area of machine learning. In this paper, we proposed an ensemble method for parameter estimation of graphical models. It generates an ensemble through incorporating Bagging or Multiple Imputation into parameter estimation. The ensemble component reduced the bias of the single parameter estimation algorithms.

Our experimental results indicated that ensemble learning can also help to further improve the parameter estimation results, while demonstrating greater amenability to parallel and distributed processing, which is important for data mining in large data sets.

References