Overview of Decompositional, Model-based Learning

Brian C. Williams

Computational Sciences Division MS 269-2

NASA Ames Research Center,

Moffett Field, CA 94305 USA

Abstract

A new generation of sensor rich, massively distributed autonomous systems are being developed that have the potential for unprecedented performance, such as smart buildings, reconfigurable factories, adaptive traffic systems and remote earth ecosystem monitoring. To achieve high performance these massive systems will need to accurately model themselves and their environment from sensor information. Accomplishing this on a grand scale requires automating the art of large-scale modeling. This paper presents a formalization of *decompositional, model-based learning (DML)*, a method developed by observing a modeler's expertise at decomposing large scale model estimation tasks. The method exploits a striking analogy between learning and consistency-based diagnosis. Moriarty, an implementation of DML, has been applied to thermal modeling of a smart building, demonstrating a significant improvement in learning rate.

1 Introduction

Through artful application, adaptive methods, such as nonlinear regression and neural nets, have been demonstrated as powerful modeling and learning techniques, for a broad range of tasks including environmental modeling, diagnosis, control and vision. These technologies are crucial to tackling grand challenge problems, such as earth ecosystem modeling, which require an army of modeling experts. In addition, hardware advances in cheap sensing, actuation, computation and networking have enabled a new category of autonomous system that is sensor rich, massively distributed, and largely immobile. These "immobile robots" are rapidly being deployed in the form of networked building energy systems, chemical plant control networks, reconfigurable factories and earth observing satellite networks. To achieve high performance these massive systems will need to accurately model themselves and their environment from sensor information. However, the labor and skill involved makes these adaptive methods economically infeasible for most large scale modeling, learning and control problems. Our goal is to automate the expertise embodied by a skilled community of modelers at decomposing and coordinating large scale model estimation or learning tasks, and to develop these methods both in the context of data analysis and hybrid control problems. The approach we call *decompositional, model-based learning (DML)*, and is embodied in a system called *Moriarty*. DML is a key element of a larger program to develop *model-based autonomous systems (MBAs)*. MBAs achieve unprecedented performance through capabilities for self-modeling (e.g., DML) and self-configuration. A complement to Moriarty, called *Livingstone*, performs discrete modeling and self-configuration[11], and will fly a deep space probe in 1988.

Our work on DML was developed in the context of synthesizing an optimal heating and cooling control system for one zone of a self-modeling building. To study this synthesis process we built a testbed for fine grained sensing and control of a building, called the *responsive environment* [4, 3], and used this testbed to study the manual art of our control engineers at decomposing the model estimation part of the control problem [13].

A key insight offered by our study is that the process of decomposing a large model estimation problem is analogous to that used in model-based diagnosis to solve large scale multiple fault diagnosis problems. The decomposition of a diagnostic problem is based on the concept of a *conflict* – a minimal subset of a first order model that is inconsistent with the set of observations [2, 7]. Decompositional learning is based on the analogous concept of a *dissent* – a minimal subset of an algebraic model that is *overdetermined* given a set of sensed variables.

The model decomposition task begins with a system of equations, including a set of sensed variables, and a set of parameters to be estimated from sensor data. DML generates the dissents of the equations and uses these dissents to generate a set of estimators that together cover all parameters. It then coordinates the individual estimations, and combines the shared results. This abstract focuses solely on the task of generating a set of dissents and a corresponding estimator for each dissent.

2 Model Estimation

Statistical modeling involves estimating the parameters $\mathbf{p}$ of a system $SD = \langle \mathbf{e}, \mathbf{v}, \mathbf{p}, \mathbf{c}, \mathbf{s} \rangle$, where $\mathbf{e}$ is a vector of system equations, $\mathbf{v}$ is the set of state variables, $\mathbf{c}$ are known constants, $\mathbf{s} \subset \mathbf{v}$ are sensed variables, and $D = \langle \mathbf{d}_i \rangle$ is sensed data.[1] For example, an office's energy and mass flow (heat, air and water) is modeled by a vector of equations $\mathbf{e}$ involving seventeen state variables $\mathbf{v}$, nine of which are sensed $\mathbf{s}$. There are four known contants $\mathbf{c}$ and there are seven unknown parameters $\mathbf{p}$:[2]

$$\begin{aligned}
\mathbf{s} &= \langle T_{ext}, T_{sply}, T_{rm}, \frac{dT_{sply}}{dt}, \frac{dT_{rm}}{dt}, X_{rht}, X_{dmpr}, F_{sply}, P_{dct} \rangle^T \\
\mathbf{c} &= \langle \rho_{lkg}, \rho_{dmpr}(X_{dmpr}), C_0, X_{rhtmax} \rangle^T
\end{aligned}$$

[1]variables in bold, such as $\mathbf{e}$ denote vectors.$\mathbf{v}^T$ transposes a row vector to a column vector.

[2]X, F, T, q and P denote position, air flow, temperature, heat flow and pressure, respectively.

$$\mathbf{p} \;=\; \langle R_{dct}, C_{rht}, Q_{rhtmax}, C_{rm}, \sigma_{wall}, Q_{eqp}, Q_{slr}(t)\rangle^T.$$

$\mathbf{e}$ consists 14 equations:

$$\begin{aligned}
F_{ext} &= F_{sply} && (1)\\
F_{sply} &= F_{rtrn} && (2)\\
Q_{ext} &= C_0 F_{ext} T_{ext} && (3)\\
Q_{sply} &= C_0 F_{sply} T_{sply} && (4)\\
Q_{rtrn} &= C_0 F_{rtrn} T_{rm} && (5)\\
Q_{rhtcap} &= Q_{ext} - Q_{sply} + Q_{rht} && (6)\\
Q_{rhtcap} &= C_{rht}\frac{dT_{sply}}{dt} && (7)\\
Q_{rht} &= \left(\frac{Q_{rhtmax}}{X_{rhtmax}}\right) X_{rht} && (8)
\end{aligned}$$

$$\begin{aligned}
F_{ext} &= F_{lkg} + F_{dmpr} && (9)\\
F_{lkg} &= \left(\frac{\rho_{lkg}}{R_{dct}}\right)\sqrt{P_{dct}} && (10)\\
F_{dmpr} &= \left(\frac{\rho_{dmpr}(X_{dmpr})}{R_{dct}}\right)\sqrt{P_{dct}} && (11)\\
Q_{rm} &= Q_{sply} + Q_{eqp} + Q_{slr}(t) &&\\
&\quad - Q_{wall} - Q_{rtrn} && (12)\\
Q_{rm} &= C_{rm}\frac{dT_{rm}}{dt} && (13)\\
Q_{wall} &= \sigma_{wall}(T_{rm} - T_{ext}) && (14)
\end{aligned}$$

Estimation involves adjusting the set of model parameters to maximize the agreement between a specified model and the data using, for example, a Bayesian or least squares criteria. Using least squares, for example, involves selecting one of the sensed variables y from $\mathbf{s}$, and using equations $\mathbf{e}$ to construct an estimator $y = f(\mathbf{x};\mathbf{p}';\mathbf{c}')$ that predicts y from parameters $\mathbf{p}' \subset \mathbf{p}$, other sensed variables $\mathbf{x} \subset \mathbf{s}$ and constants $\mathbf{c}' \subset \mathbf{c}$. The maximum likelihood estimate is the set of parameter values $\mathbf{p}{*}'$ that minimizes the mean-square error between the measured and predicted y:

$$\mathbf{p}{*}' = arg\,\min_{p} \sum_{\langle y_i,\mathbf{x}_i\rangle \in D} (y_i - f(\mathbf{x}_i;\mathbf{p}';\mathbf{c}'))^2$$

where y_i and the $\mathbf{x}_i$ are in the ith sampling of sensor values D for $\mathbf{s}$.

The modelers first attempted to estimate all parameters of the thermal problem at once, which required solving a 7-dimensional, nonlinear optimization problem involving a multimodal objective space. A Levenberg-Marquardt algorithm was applied repeatedly to this problem, but consistently became lost in local minima, and did not converge after several hours.

3 The Art of Model Decomposition

It is typically infeasible to estimate the parameters of a large model using a single estimator that covers all parameters. However, there is often a large set of possible estimators to choose from, and the number of parameters contained in each estimator varies widely. The art of data analysis (and DML) involves decomposing a task into a set of "simplest" estimators that minimize the dimensionality of the search space and the number of local minima, thus improving learning rate and accuracy. Each estimator together with the appropriate subset of sensor data forms an estimation subproblem that can be solved separately, either sequentially or in parallel.

To construct a decomposition we note that the central idea behind estimation is to select those parameters $\mathbf{p}$ that minimize the error between a model $\mathbf{M}[\mathbf{p}](\mathbf{v}) = \mathbf{z}$ and a set of data points $D = \langle \mathbf{s}_i \rangle$ for sensed variables $\mathbf{s}'$. What is important is that the existence of this error results from the model being *overdetermined by the sensed variables*.

$\mathbf{M}$ and $\mathbf{s}'$ need not be the complete set of equations $\mathbf{e}$ and sensed variables $\mathbf{s}$. Any subsystem of equations and sensed variables $\langle \mathbf{e}_d, \mathbf{s}_d \rangle$ that is overdetermined may be used to perform an estimation. Of course, not all subsystems are equal. Roughly speaking, the convergence rate is best reduced by minimizing the number of parameters mentioned in $\mathbf{e}_d$, and the accuracy of the estimation is improved by minimizing the number of sensed variables. The key consequence is that any overdetermined subsystem $\langle \mathbf{e}_d, \mathbf{s}_d \rangle$ that is minimal under subset, is better than a subsystem that is not minimal. Thus we need only generate the minimal overdetermined subsystems $\langle \mathbf{e}_d, \mathbf{s}_d \rangle$. We call these *dissents*.

To generate the set of dissents we determine for each variable v a minimal set of sensed variables and equations $\langle \mathbf{e}_c, \mathbf{s}_c \rangle$ that determine the value of v. We call $\langle \mathbf{e}_c, \mathbf{s}_c \rangle$ a support of v. Two support for v overdetermine v, and if minimal constitute a dissent $\langle \mathbf{e}_{c1} \cup \mathbf{e}_{c2}, \mathbf{s}_{c1} \cup \mathbf{s}_{c2} \rangle$. Finally, to generate the support of every v we use a local propagation algorithm, which starts by labeling each of the sensed variables s_i with a support $\langle \{\}, \{s_i\} \rangle$, and propagates the support through equations to determine the support of other variables. All pairs of support for each variable is used to generate a dissent.

Note that this propagation is analogous to environment propagation during the conflict detection phase of model-based diagnosis (MBD)[2][5]. To develop the propagation algorithm we exploit a close analogy between dissents and the MBD concept of conflicts. A conflict summarizes a logical inconsistency between a model and a set of observations, while a dissent describes a disagreement between a model and a set of sensor data. Both are a measure of disagreement between a model and observations. For MBD this is a logical disagreement, and a conflict is a minimal inconsistent theory. For DML this disagreement is a continuous error (on a euclidean metric), and a dissent is a minimal over-determined subsystem.

The concept of an "environment" in MBD parallels that of support in DML. An environment is a minimal set of component modes that entail a value for some variable, for example, $v = 6$. If two predictions are inconsistent (e.g., $v = 6$ and $v = 5$), then the union of their two environments form a conflict. Thus while an environment entails a prediction for x, a support determines the value of x, given sensor data. Exploiting this parallel DML propagates support and detects dissents similar to how MBD propagates environments and detects conflicts[2]. Note that while the algorithms are roughly similar, the mapping between logic and equational system results in key differences that are beyond the scope of an extended abstract.

Briefly, the decomposition algorithm DG1 (figure 1) propagates out from the sensed (exogenous) variables through the equations, similar to how environments are propagated outwards from distinguished literals through a network of clauses to generate a set of "environments". The function *CreateDecomposition* kicks off the propagation, while *AddSupport* recursively propagates through successive local equations, and turns the support of sensed variables $\mathbf{s}$ into dissents. The core of the algorithm is the function *Propagate*, which passes a

```
function CreateDecomposition(s_T)
  /*system s_T */
  Initialize dissents of s_T to empty
  for s_i ∈ s do
    AddSupport(s_i,⟨{},{s_i}⟩,s_T)
  endfor
  return dissents of s_T
end CreateDecomposition

function Propagate(v,⟨e,s⟩,⟨y,e,x⟩, s_T)
  /*equation e, y its effect, x its causes,
    v ∈ x, w support ⟨e,s⟩ & system s_T */
  if e ∉ e then
    S_w = WeaveSupport(v, {⟨e,s⟩}, e, x)
    for ⟨e_w, s_w⟩ ∈ S_w
      if y ∉ variables of e_w then
        AddSupport(y,⟨e_w ∪ {e}, s_w⟩,s_T)
end Propagate

function WeaveSupport(v,S,e,x)
  /*equation e, its causes x, v ∈ x,
    & its supporters S */
  if x is empty, then
    return S
  else
    h = a variable in x
    R = x − {h}
    if h = v, then
      return WeaveSupport(φ,S,e,R)
    else
      S_2 = {⟨e,s⟩|⟨e,s⟩ ∈ Support(h), e ∉ e}
      S_c = {⟨e ∪ e_2, s ∪ s_2⟩|
               ⟨e,s⟩ ∈ S, ⟨e_2,s_2⟩ ∈ S_2}
      S'_c = {s|s ∈ S_c, ¬Overdetermined?(s)}
      Return WeaveSupport(v,S'_c,e,R)
  end WeaveSupport

function AddSupport(v,⟨e,s⟩,s_T)
  /*variable v, support ⟨e,s⟩ & system s_T */
  if v ∈ s of s_T then
    Add ⟨e, s ∪ {v}⟩ to dissents of s_T
    Add v to dissenting vars. of ⟨e, s ∪ {v}⟩
  endif
  Add ⟨e,s⟩ to the support of v
  for e ∈ equations of v do
    if e ∉ e then
      C_v = CausalOrientations(e,v)
      for c ∈ C_v
      do Propagate(v,⟨e,s⟩,c,s_T)
end AddSupport

function CausalOrientations(e,v)
  /*equation e, with v selected as a cause */
  V = variables of e
  return {⟨y,e,X⟩|y ∈ V, X = V − {y}, v ∈ X}
end CausalOrientations

function ConstructEstimator(y, ⟨e,s⟩^T)
  /*dissenting variable y & its dissent ⟨e,s⟩ */
  x = s − {y}
  d = variables of e not in x
  estimator f = 'y'
  repeat until all d are eliminated from f do
    find d ∈ d and e ∈ e such that
      d occurs in f & e and
      d doesn't occur in e − {e}
    endfind
    e = e − {e}
    Solve for d in e, producing 'd = g'
    Substitute g for all occurrences of d in f
    Simplify f
  endrepeat
  return ⟨y, f, x, parameters of f⟩
end ConstructEstimator
```

Figure 1: Decomposition generation algorithm DG1 for DML

new support through an equation e. It uses the function *WeaveSupport* to combine a newly added support of a variable with a support for each of the other variables of e, save one (v), producing a composite subsystem c (we call v an *effect* and $\mathbf{x} - \{v\}$ the *causes*). It then adds e to c to produce a candidate support s for v. The function *CausalOrientations* is used by *Propagate* to select all possible pairs of causes and effect for each local equation. Finally *ConstructEstimator* takes as input a dissent, and a *dissenting variable* y (a sensed variable $s_i \in \mathbf{s}$ where the dissent was identified by *Propagate*) and produces an estimator that computes y.

Using the propagation algorithm DML generates 8 dissents for the thermal problem, with the number of parameters contained in each dissent varying from 1 to 7. Three dissents that individually contain the smallest number of parameters, and together cover the original seven parameters are:

$$\langle\langle E9-11\rangle^T \langle F_{ext}, P_{dct}, X_{dmpr}\rangle^T\rangle$$
$$\langle\langle E1, E3-4, E6-8\rangle^T, \langle \tfrac{dT_{sply}}{dt}, T_{sply}, F_{sply}, T_{ext}X_{rht}\rangle^T\rangle$$
$$\langle\langle E2, E4-5, E12-14\rangle^T, \langle \tfrac{dT_{rm}}{dt}, T_{rm}, F_{sply}, T_{sply}, T_{ext}\rangle^T\rangle$$

These dissents are converted into estimators ($y_i = f_i(\mathbf{x}_i; \mathbf{p}_i; \mathbf{c}_i)$) by selecting one of the sensed variables as y, and solving for y in terms of the other sensed variables. Selecting F_{ext}, $\frac{dT_{sply}}{dt}$ and $\frac{dT_{rm}}{dt}$ as y for the three dissents, respectively, results in the following three estimators. The first, f_1 is:

$$F_{ext} = (\rho_{lkg} + \rho_{dmpr}(X_{dmpr}))\,\frac{\sqrt{P_{dct}}}{R_{dct}}$$

where $y_1 = F_{ext}$, $\mathbf{x}_1 = \langle P_{dct}, X_{dmpr}\rangle^T$,$\mathbf{c}_1 = \langle \rho_{lkg}, \rho_{dmpr}(X_{dmpr})\rangle^T$ and $\mathbf{p}_1 = \langle R_{dct}\rangle^T$. Estimating f_1 involves just searching along one dimension. The second estimator, f_2 is:

$$\frac{dT_{sply}}{dt} = \frac{C_0 F_{sply}\,(T_{ext} - T_{sply}) + \left(\frac{Q_{rhtmax}}{X_{rhtmax}}\right) X_{rht}}{C_{rht}}$$

where $y_2 = \frac{dT_{sply}}{dt}$, $\mathbf{x}_2 = \langle T_{sply}, F_{sply}, T_{ext}X_{rht}\rangle^T$, $\mathbf{c}_2 = \langle X_{rhtmax}, C_0\rangle^T$ and $\mathbf{p}_2 = \langle C_{rht}, Q_{rhtmax}\rangle^T$. This search is two dimensional, also a simple space to explore. Finally, f_3 is:

$$\frac{dT_{rm}}{dt} = \frac{C_0 F_{sply}\,(T_{sply} - T_{rm}) + Q_{eqp} + Q_{slr}(t) + \sigma_{wall}\,(T_{ext} - T_{rm})}{C_{rm}}$$

where $y_3 = \frac{dT_{rm}}{dt}$, $\mathbf{x}_3 = \langle T_{rm}, F_{sply}, T_{sply}, T_{ext}\rangle^T$, $\mathbf{c}_3 = \langle C_0\rangle^T$ and $\mathbf{p}_3 = \langle C_{rm}, Q_{eqp}, \sigma_{wall}, Q_{slr}(t)\rangle^T$. This involves exploring a four dimensional space, a task that is not always trivial, but far simpler than the original 7D problem.[3]

Moriarty is a lisp implementation of DML, which generates dissents using the DG1 algorithm, uses Mathematica to perform symbolic manipulations required to construct estimators, and performs data analysis by generating S-PLUS target code containing the estimators.

[3] An additional stage, not presented here (see [13]), these three estimators were in turn simplified using dominance arguments (in the spirit of [12]) to a set of seven estimators, one requiring two unknown parameters to be estimated, and the remaining six involving only one unknown.

For the thermal problem Moriarty generates S-PLUS code corresponding to the above three estimators in about 10 seconds on a Sparc 2. Given ball park initial estimates for parameters and 200 data points, a single estimator containing all seven parameters required 166 seconds to converge. In contrast the total time to estimate all parameters using the above three estimators is under 9 seconds – a factor of 20 improvement.

The strong analogy here between model-based diagnosis and learning highlights an opportunity for a rich interaction between the subdisciplines and the potential for a more unified theory. In addition, DML is similar in spirit to recent work on Bayesian learning, which uses a graphical model to strongly bias the estimation process, substantially improving convergence. This includes the work of [1, 6, 10, 8, 9] and Russell et al. Work on infering independence of decision variables provides a means of decomposing the Bayesian learning task. The primary focus there is on graphical models representing probabilistic influence, where the graphical information implicitly used by DML represents deterministic influences. DML is currently being considered in a variety of exciting model-based autonomous system applications, including NASA's next generation of fully autonomous space probes, and a "biosphere-like" habitat, called a closed loop ecological life support system.

Acknowledgements

The modeling team, of the Xerox Responsive Environment project – Zhang Ying, Joseph O' Sullivan and Scott Elrod – provided invaluable insights into the process and payoffs of applying nonlinear estimation and control methods. The responsive environment software/hardware system was developed by Rick Costanza, Scott Clearwater, Jim des Rivieres, Mike Dixon and Scott Elrod. Bill Millar and Pandu Nayak provided invaluable feedback and support.

References

[1] Wray L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

[2] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987. Reprinted in [5].

[3] Elrod et al. The responsive environment: Using ubiquitous computing for office comfort control and energy management. Technical Report Technical Report CSL-93-5, Xerox PARC, 1993.

[4] Elrod et al. Responsive office environments. *Communications of the ACM*, 36(7):84–85, 1993.

[5] Walter Hamscher, Luca Console, and Johan de Kleer. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, San Mateo, CA, 1992.

[6] David Heckerman. A tutorial on learning bayesian networks. Technical Report MSR–TR–95–06, Microsoft Research, 1995.

[7] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987. Reprinted in [5].

[8] R. Shachter, S. Anderson, and K. Poh. Directed reduction algorithms and decomposable graphs. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 237–244, 1990.

[9] R. Shachter, S. Anderson, and P. Szolovits. Global conditioning for probabilistic inference in belief networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 514–522, 1994.

[10] D. Spiegelhalter and S. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.

[11] Brian C. Williams and Pandurang Nayak. Self-configuring, reactive systems: A kernel for model-based autonomy. Technical Report Technical Report, Computational Sciences Division, 1996.

[12] Brian C. Williams and Olivier Raiman. Decompositional modeling through caricatural reasoning. In *Proceedings of AAAI-94*, pages 1199–1204, 1994.

[13] Y. Zhang, B. C. Williams, and S. Elrod. Model estimation and energy-efficient control for building management systems. Technical report, Xerox PARC, 1993.