Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Leveraging Domain Knowledge in Multitask Bayesian Network Structure Learning

Diane Oyen and **Terran Lane**
Department of Computer Science
University of New Mexico

Abstract

Network structure learning algorithms have aided network discovery in fields such as bioinformatics, neuroscience, ecology and social science. However, challenges remain in learning informative networks for related sets of tasks because the search space of Bayesian network structures is characterized by large basins of approximately equivalent solutions. Multitask algorithms select a set of networks that are near each other in the search space, rather than a score-equivalent set of networks chosen from independent regions of the space. This selection preference allows a domain expert to see only differences supported by the data. However, the usefulness of these algorithms for scientific datasets is limited because existing algorithms naively assume that all pairs of tasks are equally related. We introduce a framework that relaxes this assumption by incorporating domain knowledge about task-relatedness into the learning objective. Using our framework, we introduce the first multitask Bayesian network algorithm that leverages domain knowledge about the relatedness of tasks. We use our algorithm to explore the effect of task-relatedness on network discovery and show that our algorithm learns networks that are closer to ground truth than naive algorithms and that our algorithm discovers patterns that are interesting.

Introduction

Scientists in domains such as neuroscience use network structure learning algorithms to discover patterns of interaction in multivariate data. For these datasets, multitask learning algorithms learn robust models even when the number of data samples collected in a specific task are limited, but there are several tasks that are believed to be similar (Caruana 1997; Baxter 2000). For example, in group neuroimaging studies, we learn functional brain networks for several populations of subjects and treat the data from each population as a task. We expect the population-specific networks to have a lot in common but not to be identical. Furthermore, we may be able to describe relationships between the populations, such as a study of functional brain networks for populations of different ages of subjects. The groups that are close in age should have the most similar networks.

Our goal is to learn the best set of networks that are interesting to the domain expert. The concept of incorporating human preferences into unsupervised learning objectives is an active research trajectory in clustering (Dasgupta and Ng 2009) and topic modeling (Chang et al. 2009). In these unsupervised learning problems data-driven measures of goodness of the learned model are insufficient and can only be addressed by incorporating human objectives. We bring this concept to network structure learning. The search space of Bayesian network structures is characterized by large basins of approximately equivalent solutions (Friedman and Koller 2003). Multitask algorithms provide a first step toward incorporating a human bias by selecting a set of networks that are near each other in the search space, rather than a score-equivalent set of networks chosen from independent regions of the space (Niculescu-Mizil and Caruana 2007).

Existing multitask methods for unsupervised problems typically assume that all pairs of tasks are equally related. This assumption makes these algorithms too rigid to handle datasets where different pairs of tasks have widely varying degrees of task-relatedness. Furthermore, they provide no mechanism for incorporating human objectives for task-relatedness. A few specialized multitask network discovery applications have recently incorporated specific domain knowledge about task-relatedness (Husmeier, Dondelinger, and Lèbre 2010; Dondelinger, Lèbre, and Husmeier 2010; Liu et al. 2010).

We introduce a framework for multitask structure learning that relaxes the assumption that tasks are equally related. In many applications we have prior beliefs about the relatedness of tasks based on metadata or domain expert knowledge. Using our framework, we develop the first multitask Bayesian network structure learning algorithm to incorporate task-relatedness as a parameter. With our algorithm we explore various factors in the problem space: the number of tasks, the true similarity between tasks, and the topology of task-relatedness. We compare the performance of our algorithm with naive algorithms (those without task-relatedness knowledge). We find that our algorithm generalizes to validation data and fits ground truth better than naive algorithms.

Finally, we learn functional brain networks from neuroimaging data with our Bayesian network algorithm. For a given pool of subjects, there are a number of “natural” task divisions (e.g, pooling by age or by medication). We explore different divisions of subjects into tasks, with corresponding task relatedness metrics and discuss the interesting patterns found by our algorithm. Domain knowledge about task-relatedness improves both the robustness of learned networks and addresses human objectives.

Related Work

Multitask learning algorithms generally represent the similarity among tasks in one of three ways: all tasks are assumed to be equally similar (Caruana 1997; Niculescu-Mizil and Caruana 2007); the similarity among tasks is estimated from the same data that is used to train the model (Thrun and O'Sullivan 1996; Eaton, desJardins, and Lane 2008); or the similarity between tasks is provided by another source such as task-specific domain information or an expert (Bakker and Heskes 2003). This third option, which we apply to network discovery, has been used successfully for zero-shot classification problems when no training data are available for certain tasks (Larochelle, Erhan, and Bengio 2008; Palatucci et al. 2009).

Multitask learning has been applied to learn Gaussian graphical models (Honorio and Samaras 2010) and Bayesian networks (Niculescu-Mizil and Caruana 2007; Luis, Sucar, and Morales 2009). Unlike our framework, these models assume that all tasks are equally related. There is recent work in specialized application-specific algorithms that share information only between tasks that are believed to be most similar (Liu et al. 2010; Husmeier, Dondelinger, and Lèbre 2010; Dondelinger, Lèbre, and Husmeier 2010). These applications demonstrate the benefit of domain knowledge.

Preliminaries: Multitask Structure Learning

Probabilistic graphical models compactly describe joint probability distributions by encoding independencies in multivariate data. Multitask learning enforces a bias toward learning similar independency patterns among tasks.

A Bayesian network $B = \{G, \theta\}$ describes the joint probability distribution over n random variables $\mathbf{X} = [X_1, X_2, \ldots, X_n]$, where G is a directed acyclic graph and the conditional probability distributions are parameterized by θ (Heckerman, Geiger, and Chickering 1995). An edge (X_i, X_j) in G means that the child X_j is conditionally independent of all non-descendants given its parent X_i. A Markov random field (MRF) encodes similar conditional independencies with an undirected graphical model (Kindermann and Snell 1980). The *structure* of the network, G, is of particular interest in many domains as it is easy to interpret and gives valuable information about the interaction of variables.

A set of tasks with data sets D_k and networks G_k for $k \in \{1, \ldots, K\}$ can be learned by optimizing:

$$P(G_{1:K}|D_{1:K}) \propto P(D_{1:K}|G_{1:K})P(G_{1:K}) \qquad (1)$$

Breaking this into independent single-task learning problems (STL) assumes all tasks are independent of each other, which simplifies Equation 1 to:

$$P_{STL}(G_{1:K}|D_{1:K}) \propto \textstyle\prod_{k=1}^{K} P(D_k|G_k)P(G_k)$$

Multitask learning (MTL) does not assume that tasks are independent, however it does generally assume that $P(D_k|G_k)$ is independent of all other G_i so Equation 1 simplifies to:

$$P_{MTL}(G_{1:K}|D_{1:K}) \propto P(G_{1:K}) \textstyle\prod_{k=1}^{K} P(D_k|G_k)$$

In multitask learning, the joint structure prior, $P(G_{1:K})$, is used to encode a bias toward similar structures. We can break down this joint distribution into a product of conditionals, so that $P(G_{1:K}) = P(G_1) \prod_{i=2}^{K} P(G_i|G_{1:i-1})$. Many multitask algorithms (including those outlined in this paper), make a further simplifying assumption that $P(G_k|G_{1:k-1}) = \prod_{i=1}^{k-1} P(G_k|G_i)$. That is, the joint prior over structures can be described by pairwise sharing of information among tasks:

$$P(G_{1:K}) \triangleq \prod_{i=2}^{K} P(G_i) \prod_{j=1}^{i-1} P(G_i|G_j) \qquad (2)$$

Task-Relatedness Aware Multitask Learning

We introduce our general framework for incorporating prior knowledge about task-relatedness in multitask structure learning. The goal is to include a weighting scheme for the amount of information sharing between different pairs of tasks. First, we define a symmetric matrix, $\boldsymbol{\mu}$, of size $K \times K$ where each element, $\mu_{ij} \geq 0$, describes prior information about the degree of relatedness between tasks i and j. These values come from a task-relatedness metric that describes the degree of relatedness of pairs of tasks. We find it more convenient to work with the inverse of the metric so that $\mu_{ij} = 0$ means that the tasks are independent, and large values of μ_{ij} mean a high degree of relatedness. Then, using the general description of the joint prior over network structure, in Equation 2, we use $\boldsymbol{\mu}$ to weight the transfer bias among pairs of tasks. With this additional input about the relatedness of tasks, the new **T**ask-**R**elatedness **A**ware **M**ultitask objective (TRAM) to maximize is:

$$\begin{aligned} P_{TRAM}(G_{1:K}|D_{1:K}, \boldsymbol{\mu}) &\propto P(G_{1:K}|\boldsymbol{\mu}) \prod_{i=1}^{K} P(D_i|G_i) \\ P(G_{1:K}|\boldsymbol{\mu}) &\triangleq \frac{1}{Z_{\boldsymbol{\mu}}} \textstyle\prod_{i=2}^{K} P(G_i) \prod_{j=1}^{i-1} P(G_i|G_j)^{\mu_{ij}} \end{aligned} \qquad (3)$$

Key benefits of the TRAM framework are:

- Introduces a task relatedness metric which allows explicit control of information sharing between tasks.
- Subsumes MTL (all elements of $\boldsymbol{\mu}$ are 1) and STL (all elements of $\boldsymbol{\mu}$ are 0).
- Includes existing application-specific models as discussed in the Special Cases section.
- Provides convenient mechanism for incorporating task-relatedness in multitask network structure learner, such as our Bayesian network learner discussed in the next section.

This framework is general enough to cover any network discovery algorithm that enforces bias between pairs of tasks. Extensions would be required to cover higher-order relationships among tasks, such as describing task-relatedness as a Markov random field with appropriate higher-order potential functions to penalize differences among related tasks.

Multitask Learning of Bayesian Networks

Our novel task-relatedness aware multitask Bayesian network structure learning algorithm illustrates the use of the framework. To apply the objective function in Equation 3 to multitask Bayesian networks we define $P(D_i|G_i)$ as the Bayesian likelihood score $P(D|G) = \int P(D|G,\theta)P(\theta|G)d\theta$. The prior over structures encodes the bias toward similar structures by penalizing differences in edges among tasks:

$$P(G_i|G_j) \triangleq \frac{1}{Z_{ij}}(1-\alpha)^{\Delta(G_i,G_j)}$$

where Δ is a graph distance metric. The parameter $\alpha \in [0,1]$ controls the relative strength of fit to data versus bias toward similar models. When $\alpha = 0$, the objective function is equivalent to learning the tasks independently. When $\alpha = 1$ the only solutions that produce a non-zero probability are those in which $\Delta(G_i, G_j) = 0$, in other words all structures must be identical. The parameters are always inferred independently for each task.

Any graph distance metric can be used for Δ depending on the desired definition of structure similarity. If all D_i come from analogous random variables, the distance metric can be a simple graph edit distance. In our experiments, we use edit distance (the number of edge additions, deletions or reversals necessary to change G_i into G_j).

Optimization of the multitask Bayesian network structure learning objective proceeds by searching over the space of DAG for a high-scoring set of DAGs. We follow a commonly used search heuristic, greedy search, which starts from an initial structure and then iteratively makes the best change (edge addition, deletion or reversal) to the network structure until no further improvements can be made. The best change is the one that gives the greatest improvement in score. We are optimizing several network structures simultaneously, therefore one edge in each task can be changed at each iteration. Incorporating the task-relatedness metric does not incur any computational cost above standard multitask learning.

Special Cases

Now we show how the framework subsumes two application-specific examples from the literature. The dynamic Bayesian network structure learning algorithms with inter-time segment sharing in Husmeier, Dondelinger, and Lèbre (Husmeier, Dondelinger, and Lèbre 2010) and Dondelinger, Lèbre, and Husmeier (Dondelinger, Lèbre, and Husmeier 2010) can be written using the TRAM framework as follows. Each time segment is a task i for which they learn a graph G_i that is biased toward having a similar structure to the previous time segment, G_{i-1}. The structure prior is $P(G_{1:K}) = P(G_1)\prod_{i=2}^{K}(1/Z_i)\exp(-\beta\Delta(G_i, G_{i-1}))$, where β is a hyper-parameter and $\Delta(G_i, G_j)$ is the Hamming distance between edge sets. To fit into our framework, we write the prior according to Equation 3 with the task-relatedness metric defined as $\mu_{ij} = 1$ for $i = \{2 \dots K\}$ and $j = i - 1$, and $\mu_{ij} = 0$ otherwise.

Experiments on Synthetic and fMRI Data

We empirically evaluate our TRAM Bayesian network learning algorithm on synthetic and real-world data. For comparison, we also learn each network independently with single-task learning (STL) and learn a single network structure for all contexts (AVG), so named because this assumes that there is some "average" network that is representative of all tasks. Note, AVG learns the same structure for all tasks, but that the parameters are independent of the other tasks. We also compare against a standard multitask learning algorithm (MTL) that assumes all tasks are equally related (all $\mu_{ij} = 1$) (Niculescu-Mizil and Caruana 2007). For these experiments, we use greedy structure search, starting from an empty network and use a Bayesian score. For TRAM and MTL, we tune the strength parameter, α, with a small hold-out set (10% of the training data). All reported results are averaged over 10-fold cross validation.

Netsim Data

We use benchmark data from Smith et al. (2011) which is generated from known networks to simulate rich realistic functional magnetic resonance imaging (fMRI) data. They generated the data using a hemodynamic response model (Friston, Harrison, and Penny 2003). We quantize the given continuous data into binary and fit multinomial functions when learning the networks. We use the benchmark data for 50 synthetic subjects with 200 training samples per subject from 50-node networks. The given network structures for all subjects are identical (the functional relationships between nodes are subject-specific) but we are interested in models where the structure differs. Therefore, we modify the structures by re-labeling various numbers of nodes for some tasks and then combining those re-labelings for other tasks. For example, at the top of Figure 1 the adjacency matrix for a given network is used as the first task. To create a related task, we swap the node labels between a few pairs of nodes thus changing some edges of the adjacency matrix as seen in the next task. Re-labeling produces isomorphic networks, so that we can use the data provided and maintain the general properties of each network while giving different-looking network structures to the structure learning algorithms. TRAM is not given the true measure of similarity between tasks, instead we set $\mu_{ij} = 1$ for each pair of tasks i, j with an edge in the task-relatedness topology, and 0 otherwise.

Results for NetSim Data

We vary the number of differences between tasks, the number of tasks, and the topology of task-relatedness (see Figure 1). The second row of Figure 1 shows the average percent improvement in likelihood of holdout data for TRAM

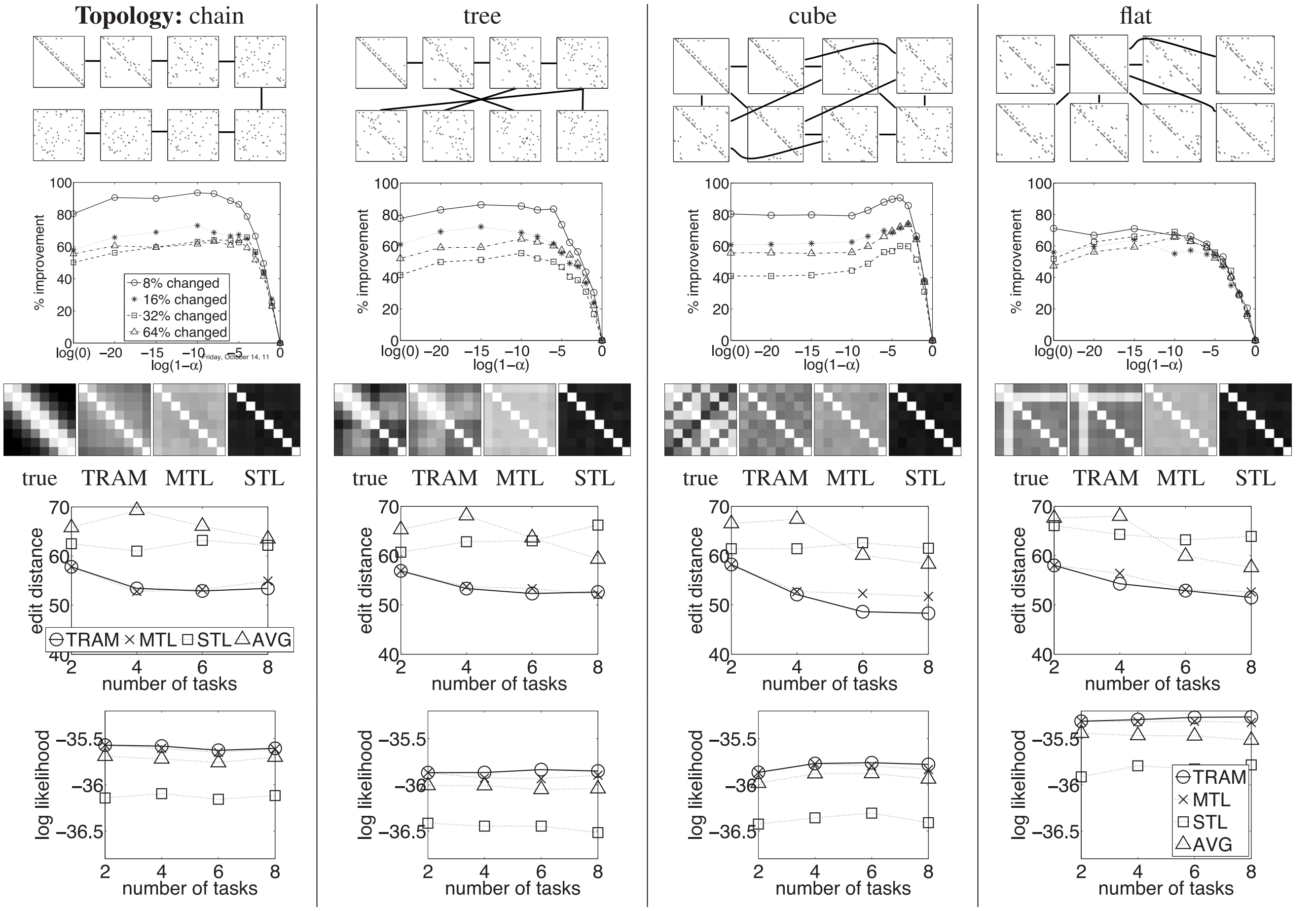


Figure 1: NetSim data results. **Top row:** Our generated task-relatedness topologies. Each square node in the topology graph represents a task. The square itself is an image of the adjacency matrix of the ground truth network where dots in the images represent directed edges in the ground truth network. The lines between nodes in the task-relatedness topology indicate that the two tasks are similar with $\mu_{ij} = 1$. **Second row:** TRAM's percent improvement in likelihood on holdout data over STL, across values of α for various levels of true task similarity (% changed). **Third row:** Similarity between tasks for the *true* networks and learned networks from TRAM, MTL and STL for 8 tasks as measured by graph edit distance. White squares mean < 30, black squares mean > 100. **Fourth row:** Edit distance (down is good) of learned networks to ground truth for the four algorithms for 2, 4, 6, and 8 tasks. **Bottom row:** Score of learned networks (up is good) for the four algorithms for 2, 4, 6, and 8 tasks.

over STL. On the x-axis, we vary the strength parameter α on a log scale. When $\alpha = 0$, the tasks are learned independently (the right end of the plot). As we move across the plot to the left, the bias toward learning similar models increases until $\alpha = 1$ at the left end of the plot, where all structures learned are identical to each other. Each line in the plot corresponds to a generative process of node re-labeling that changes the node label for the given percentage of nodes in the network, where high percentages mean there are more differences in the true networks between tasks. As expected, the plots show that when the true networks are most similar (8% changed), the performance gained by TRAM over STL is greatest. As the number of true differences between networks increases, biasing the models toward each other is still a large improvement over STL, but if the strength parameter gets too high then performance degrades.

The bottom three rows of plots in Figure 1 compare the performance of all algorithms on the datasets with 32% of nodes relabeled (other results show similar trends and are omitted for space). The row of grayscale images show the similarity among task-specific networks as measured by graph edit distance. For example, in the true networks for the *chain* topology, we see that the first task is increasingly dissimilar to the other tasks as we look across the the top row. STL learns networks that are highly dissimilar to each other while MTL and TRAM learn networks that are more similar, reflecting the bias in these algorithms. TRAM is the only algorithm that reflects the patterns of task similarity given by the true networks.

Perhaps more importantly, the bottom two rows of Figure 1 indicate that TRAM learns models that are as close to ground truth as MTL and always better than STL and AVG. We did not perform experiments on training set size because existing literature has well documented that

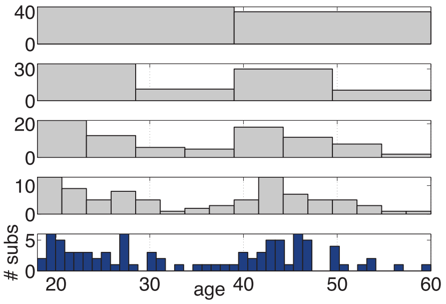


Figure 2: **Age groups.** Bins of subjects grouped by age for 2, 4, 6, and 16 tasks. Each box is a task; the width shows the age range and the height shows the number of subjects in the task. The bottom row is a histogram of all subjects.

multitask algorithms are most beneficial on small datasets (Niculescu-Mizil and Caruana 2007). The NetSim data provides 200 samples per task which we find is insufficient for good single-task learning, making the data a good candidate for multitask learning. We ran experiments with smaller amounts of training data and found those results to be consistent with existing literature.

Network Discovery in fMRI

Functional MRI measures the activation level of voxels in the brain. Typically, hundreds of such images are collected over the period of time that a subject is in the scanner. We use data from a large schizophrenia study, where 384 volumes are sampled per subject. Voxels are mapped into 150 regions of interest (ROIs) based on the Talaraich atlas (Lancaster et al. 2000). The fMRI data for each ROI in each subject is independently detrended and discretized into four levels of activity. Thus, our data is 150 variables by 384 samples per subject. We use the same algorithms as with NetSim.

Age-Groups as Tasks

A fundamental question for multitask learning in practice is — how do we define a task? While we cannot fully answer the question in this paper, we do explore how the number of tasks for a fixed dataset affect the performance of the learned models. We experiment with dividing our dataset into various numbers of tasks by grouping subjects into tasks by age. We take 86 subjects from our dataset (the control subjects in the schizophrenia study) and group them based on the age of the subject. Figure 2 shows how we create 4 different learning problems by dividing the dataset into 2, 4, 8, and 16 tasks. The training data is the same across these problems, but the number of tasks is different. We define the task-relatedness values $\mu_{ij} = e^{-(\bar{a}_i - \bar{a}_j)^2/(2\sigma^2)}$ where $\bar{a}_i$ is the average age of subjects in task i and σ^2 is the variance of ages of all subjects. As an example, $\mu_{1j} = [1, .89, .67, .37, .18, .09, .03, .005]$ for the youngest group (task 1) versus tasks j in order of increasing age for 8 tasks. For comparison, we also try binary-valued μ with $\mu_{ij} = 1$ for pairs of tasks i, j that are adjacent to each other in age-order and 0 otherwise.

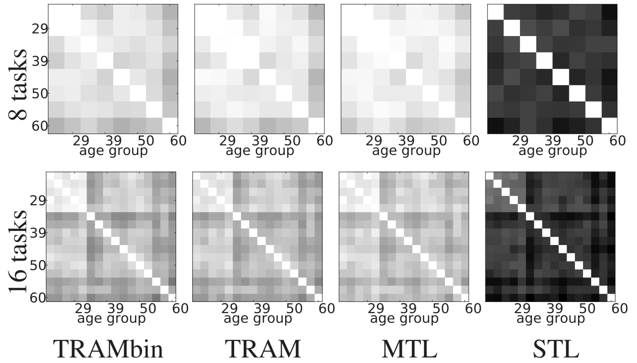


Figure 3: **Age data task similarity.** Edit distance between learned task-specific networks. White is <100, black is >300.

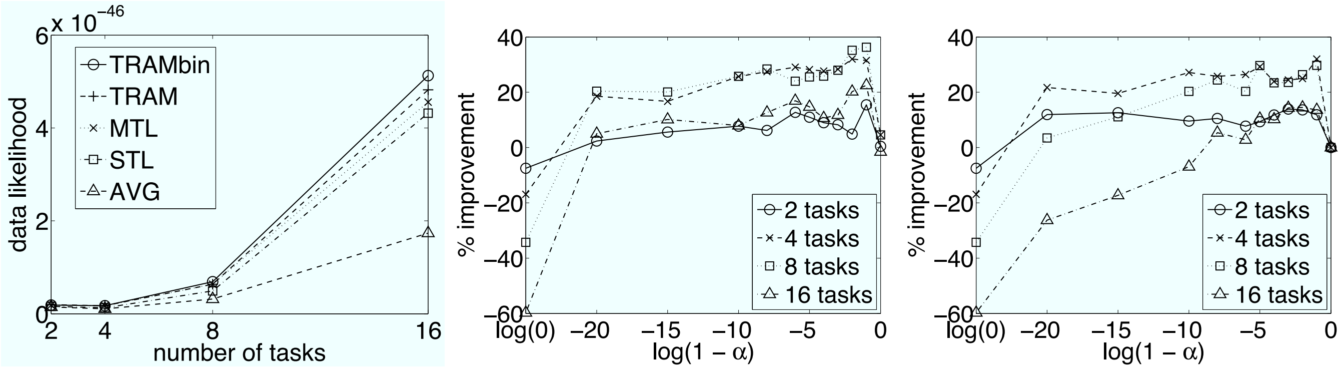
Results from the age data are shown in Figures 4 and 3. Plots 4(b) and 4(c) show TRAMbin and TRAM's sensitivity to the strength parameter α as measured by the improvement in likelihood on test data versus STL. We see that both are an improvement over STL but TRAMbin appears less sensitive to α. AVG (the left edge of the plot at $\log(0)$) actually causes negative transfer that is increasingly bad as the number of tasks grows. Therefore, some biasing of models helps, but too much degrades performance.

The Figure 4(a) shows the overall comparison between algorithms of data likelihood on test data. Here, the strength parameters of TRAMbin, TRAM and MTL have been tuned on 10% of the training data. We see that splitting the dataset into more tasks improves the performance of all algorithms, even though the underlying dataset is the same. TRAMbin is always the highest performing algorithm, with TRAM and MTL close behind. The lines appear quite close, but the differences are significant everywhere except between TRAMbin, TRAM and MTL for 2 and 4 tasks according to paired t-tests at p=0.05. The improvement in performance of AVG is somewhat surprising because tasks share the same structure. However, recall that the parameters of different tasks are independent, therefore splitting data into tasks also allows AVG to fit parameters to specific tasks.

Interestingly, TRAMbin and TRAM find quite a few network edges that are different for the oldest age group than for the others (see Figure 3). All algorithms find more differences from the oldest group to the others, but for TRAM and TRAMbin many of these edges exist with high robustness in the oldest group, but none of the others. Other edges exist with high robustness in the younger groups but never in the oldest group.

Tasks Defined by Medication Type

Often we want to look at populations of subjects with a certain type of mental illness, but we expect that different drug treatments will have an effect on brain activity. To address this, we divide the subjects from the schizophrenia study into 7 tasks. One of the tasks is the group of control subjects. The other tasks are schizophrenic patients divided into 6 groups representing the medication they are taking (Figure 5).



(a) All algorithms (b) TRAMbin (c) TRAM

Figure 4: **Age data.** (a) Likelihood of holdout data. All differences between algorithms are significant at p=.05 except for between TRAM, TRAMbin and MTL at 2 and 4 tasks. (b) TRAMbin's increase in performance over STL. (c) TRAM's increase in performance over STL.

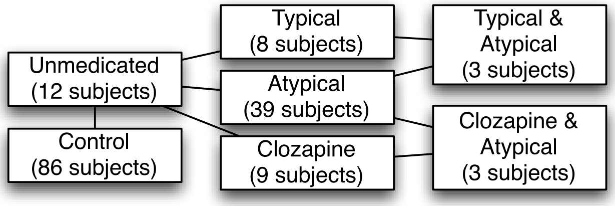

Typical (8 subjects)
Typical & Atypical (3 subjects)
Unmedicated (12 subjects)
Atypical (39 subjects)
Clozapine & Atypical (3 subjects)
Control (86 subjects)
Clozapine (9 subjects)

Figure 5: Task relatedness for Drug data. Each square is a task, edges between tasks represent $\mu = 1$.

Figure 6 shows improvement in data likelihood versus STL. The improvement is greater for TRAM than MTL. AVG always performs worst of all of the algorithms. As expected, the improvement of TRAM over STL is greatest when there is the least data (Figure 6(b)). All algorithms learn networks that show high variation between the control group and all other tasks. The networks learned by TRAM in particular show that networks for subjects on Clozapine type drugs are most similar to those on drug combinations including Clozapine than they are to any other group. On the other hand, for subjects on Typical type drugs TRAM learns brain networks that are highly similar to all other groups.

Discussion and Future Work

Task-relatedness knowledge can improve both the robustness of learned networks and address human objectives. A natural question is how to define the task-relatedness metric $\boldsymbol{\mu}$. Previous application specific algorithms employed binary task-relatedness weights. Our experiments support the intuition that binary weights that give the topology of tasks that are directly related is preferable to fine-tuning real-valued weights. These findings warrant further investigation. A similar question is how fragile algorithms become when domain knowledge is poor. In practice, we found that using a misleading $\boldsymbol{\mu}$ causes TRAM to produce results equivalent to MTL, which is not surprising because MTL is a case of TRAM with a fixed $\boldsymbol{\mu}$. Theoretical definitions of good task-relatedness knowledge would be interesting future work.

Another important direction of research is to estimate task-relatedness from data. However, the data-driven approach answers a different question than addressed in this paper. Instead, TRAM is incorporating a human-specified objective. This concept of incorporating human preferences into learning objectives is an active research trajectory in unsupervised learning (Dasgupta and Ng 2009; Chang et al. 2009). In these problems data-driven measures of goodness of the learned model are insufficient and can only be addressed by incorporating human objectives. We have introduced this concept to network structure learning.

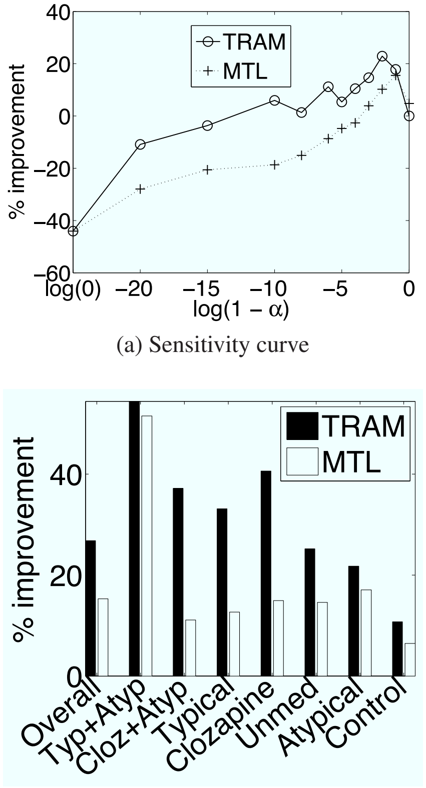
Learning a large number of Bayesian network tasks efficiently is another direction for future work. Currently, no multitask Bayesian network learning algorithm adequately addresses this. Task-relatedness knowledge may be useful to break up the problem into manageable-sized chunks. It would also be interesting to investigate transferring bias among only the parts of the Bayesian network model that we are most interested in and allow a more efficient independent search for other parts of the model.

Conclusion

We have shown that naive assumptions about task-relatedness limit the effectiveness of multitask learning algorithms. Relaxing these assumptions in the objective function through a task-relatedness metric is a necessary step in improving the performance of multitask network structure learning algorithms. We introduced our general framework, TRAM, for incorporating domain knowledge into multitask network structure learning objectives. Our framework allows a natural and flexible way to represent domain knowledge. Also, we presented a novel multitask Bayesian network structure learning algorithm with TRAM. Empirical evaluation shows that leveraging domain knowledge produces models that are both robust and reflect a domain expert's objective.

Acknowledgements

We would like to thank Vince Clark and the Mind Research Network for providing data and introducing us to this in-



(a) Sensitivity curve

(b) Per-task improvement

Figure 6: Drug dataset results. (a) Increase in performance over STL across values of the strength parameter for TRAM and MTL. (b) Improvement over STL for tuned TRAM and MTL. Note tasks are ordered by increasing number of subjects.

teresting problem. We are grateful to Eric Eaton and Vamsi Potluru for discussions and feedback on early drafts. Thanks also to Alex Niculescu-Mizil for providing his multitask Bayesian network code. This work was supported by the Office of Naval Research under grant N000141110139 and the National Science Foundation under grant IIS-0705681.

References

Bakker, B., and Heskes, T. 2003. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research* 4:83–99.

Baxter, J. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research* 12:149–198.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Chang, J.; Boyd-Graber, J.; Gerrish, S.; Wang, C.; and Blei, D. 2009. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 1–9.

Dasgupta, S., and Ng, V. 2009. Single data, multiple clusterings. In *NIPS Workshop on Clustering: Science or Art? Towards Principled Approaches*.

Dondelinger, F.; Lèbre, S.; and Husmeier, D. 2010. Heterogeneous continuous dynamic Bayesian networks with flexible structure and inter-time segment information sharing. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*.

Eaton, E.; desJardins, M.; and Lane, T. 2008. Modeling transfer relationships between learning tasks for improved inductive transfer. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I (ECML PKDD -08)*, 317–332.

Friedman, N., and Koller, D. 2003. Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* 50(1):95–125.

Friston, K. J.; Harrison, L.; and Penny, W. 2003. Dynamic causal modelling. *NeuroImage* 19(4):1273–1302.

Heckerman, D.; Geiger, D.; and Chickering, D. M. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20(3):197–243.

Honorio, J., and Samaras, D. 2010. Multi-task learning of Gaussian graphical models. In *Proceedings of the 27th International Conference on Machine Learning (ICML -10)*.

Husmeier, D.; Dondelinger, F.; and Lèbre, S. 2010. Inter-time segment information sharing for non-homogeneous dynamic Bayesian networks. In *Advances in Neural Information Processing Systems 23*, 901–909.

Kindermann, R., and Snell, J. L. 1980. *Markov Random Fields and Their Applications*, volume 1 of *Contemporary Mathematics*. Providence, Rhode Island: American Mathematical Society.

Lancaster, J. L.; Woldorff, M. G.; Parsons, L. M.; Liotti, M.; Freitas, C. S.; Rainey, L.; Kochunov, P. V.; Nickerson, D.; Mikiten, S. A.; and Fox, P. T. 2000. Automated Talairach atlas labels for functional brain mapping. *Human Brain Mapping* 10(3):120–131.

Larochelle, H.; Erhan, D.; and Bengio, Y. 2008. Zero-data learning of new tasks. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, 646–651.

Liu, Y.; Niculescu-Mizil, A.; Lozano, A.; and Lu, Y. 2010. Temporal graphical models for cross-species gene regulatory network discovery. In *Proc LSS Comput Syst Bioinform Conf*, volume 9, 70–81.

Luis, R.; Sucar, L. E.; and Morales, E. F. 2009. Inductive transfer for learning Bayesian networks. *Machine Learning* 79(1-2):227–255.

Niculescu-Mizil, A., and Caruana, R. 2007. Inductive transfer for Bayesian network structure learning. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*.

Palatucci, M.; Pomerleau, D.; Hinton, G.; and Mitchell, T. 2009. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems (NIPS)*, 1410–1418.

Thrun, S., and O'Sullivan, J. 1996. Discovering structure in multiple learning tasks: The TC algorithm. In *Proceedings of International Conference on Machine Learning*, 489–497.