

Modeling the fMRI Signal via Hierarchical Clustered Hidden Process Models

Radu Stefan Niculescu PhD¹, Tom M. Mitchell PhD², R. Bharat Rao PhD¹

¹Computer Aided Diagnosis & Therapy Group, Siemens Medical Solutions, Malvern, PA

²Computer Science Department, Carnegie Mellon University, Pittsburgh, PA

Keywords: fMRI, Probabilistic Models, Bayesian Networks, Clustering

ABSTRACT

Machine Learning techniques have been used quite widely for the task of predicting cognitive processes from fMRI data. However, these models do not describe well the fMRI signal when it is generated by multiple cognitive processes that are simultaneously active. In this paper we consider the problem of accurately modeling the fMRI signal of a human subject who is performing a task involving multiple concurrent cognitive processes. We present a Hierarchical Clustering extension of Hidden Process Models which, by taking advantage of automatically discovered similarities in the activation among neighboring voxels, achieves significantly better performance than standard generative models in terms of Average Log Likelihood.

1. INTRODUCTION

Functional Magnetic Resonance Imaging (fMRI) is a technique for obtaining three-dimensional images of activity in the brain, over time. A typical fMRI image contains several thousand voxels, where each voxel covers a few tens of cubic millimeters of brain tissue. The fMRI activation observable due to neural activity extends for approximately 10 seconds after the neural activity, resulting in a temporally blurred response. In [1] we present a brief overview of machine learning approaches to fMRI analysis.

In order to model the collection of cognitive processes that give rise to this brain activation, we have recently developed *Hidden Process Models* (HPMs). HPMs provide a probabilistic approach to estimating the fMRI activation generated over time by each of several individual cognitive processes that combine to form this observed fMRI brain activation. Learning accurate HPMs is often difficult in practice due to the large number of model parameters that must be estimated, and due to the noise and sparsity of fMRI data. This paper presents an approach to improving the accuracy of learned HPM models by reducing the effective number of parameters to be estimated. In particular, we present an algorithm that discovers clusters of neighboring brain locations whose HPM parameters can be shared, and a learning algorithm that takes advantage of these parameter-sharing assumptions to estimate parameters more reliably. We show experimentally that by taking advantage of the discovered parameter sharing assumptions our algorithm is able to learn more accurate HPMs.

2. THE DATASET

The experiments reported here are based on fMRI data collected in a study of sentence and picture comprehension [2]. Subjects in this study were presented with a sequence of 40 trials. In 20 of these trials, the subject was first presented with a sentence for 4 seconds, such as “*The plus sign is above the star sign.*”, then a blank screen for 4 seconds, and finally a picture such as:



for another 4 seconds. During each trial, the subject was required to press a *Yes* or *No* button to indicate whether the sentence correctly described the picture. During the remaining 20 trials the picture was presented first and the sentence presented second, using the same timing.

In this dataset, the voxels were grouped into 24 anatomically defined spatial regions of interest (ROIs), each voxel having a resolution of 3 by 3 by 5 millimeters. An image of the brain was taken every half second. For each trial, we considered only the first 32 images (16 seconds) of brain activity. Due to the time it takes to complete the experiments, the results reported in this paper are limited to data from a single human subject. For this subject, our dataset tracked the activity of 4698 different voxels.

3. APPROACH

A *Hidden Process Model* (HPM) [3] is a probabilistic framework for modeling the fMRI time series data in all brain voxels of a human subject who is performing a cognitive task. A hidden process may be thought of as a mental process that generates fMRI activity in the brain, in response to an external *stimulus*. For example, a *ComprehendPicture* process may describe the fMRI signal that happens in the brain starting when the subject is presented with a picture. A *ComprehendSentence* process may provide the same characterization for the situation when a subject is reading a sentence. HPMs assume several cognitive processes may be active at some point in time, and assume in such cases the observed fMRI signal is the sum of the corresponding processes, translated according to their starting times. Hidden Process Models can be viewed as a subclass of Dynamic Bayesian networks, as described in [2].

Formally, a *Hidden Process Model* is defined by a collection of time series (also called hidden processes) P_1, \dots, P_K . For

each process P_k with $1 \leq k \leq K$, denote by P_{kt} the value of its corresponding time series at time t after the process starts. Also, let X_t be the activation in voxel X at time t . If process P_k starts at time t_k , then a Hidden Process Model predicts the random variable X_t will follow the distribution:

$$X_t \sim N\left(\sum_k P_{k(t-t_k+1)}, \sigma^2\right)$$

where σ^2 is considered to be the variance in the measurement and is kept constant across time. For the above formula to make sense, we consider $P_{kt}=0$ if $t < 0$.

Figure 1 shows an example of a Hidden Process Model for the cognitive task introduced in Section 2. This Hidden Process Model has two processes: “Sentence” (P_1) and “Picture” (P_2). Each trial has length $T=32$ fMRI snapshots (16 seconds) and the same holds for both processes. This figure shows the fMRI activation in a voxel X during a trial where the sentence is presented at time $t_1=1$ and the picture is shown at $t_2=17$ (8 seconds after t_1). After time t_2 the two processes overlap and the fMRI signal X_t is the sum of the corresponding values of the two processes plus $N(0, \sigma^2)$ measurement variance. The blue dotted line represents the fMRI activity that would happen after time T .

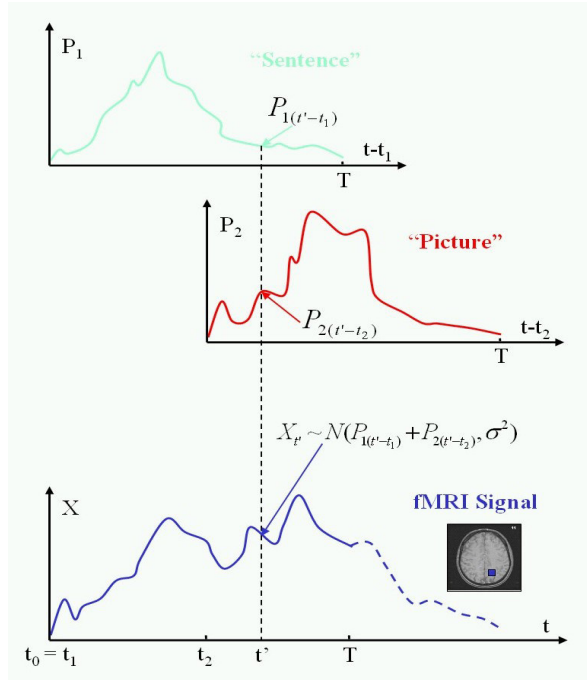


Figure 1. Example of a Hidden Process Model

In general, HPMs allow modeling uncertainty about the timing of hidden processes, allow uncertainty about the types of the processes, and allow for multiple instances of the same process to be active simultaneously [3]. However, in the treatment and experiments in this paper we make three simplifying assumptions. We assume the times at which the hidden processes occur are known, that the types of the processes are known, and that two instances of the same type of process may not be active simultaneously. These

three simplifying assumptions lead to a formulation of HPMs that is equivalent to the approach of [4] based on multivariate regression within the General Linear Model.

In a typical fMRI experiment, the subject often performs the same cognitive task during multiple *trials*, providing multiple observation sequences for each voxel X . In our framework we denote by X_{nt} the value of X_t during trial n and by t_{nk} the starting point of process P_k during trial n . Let N be the number of observations. We can now write:

$$X_{nt} \sim N\left(\sum_k P_{k(t-t_{nk}+1)}, \sigma^2\right)$$

While not entirely necessary for our method to work, we assume that X is tracked for the same length of time in each trial. Let T be the length of every such trial (observation). Since we are not modeling what happens when $t > T$, we can also assume that each process has length T .

The natural constraints of this domain lead to an opportunity to specify prior knowledge in the form of parameter constraints, as follows: an external stimulus will typically influence the activity in multiple voxels of the brain during one cognitive task. For example, looking at a picture may activate many voxels in the visual cortex (several ideas for exploiting spatial coherence are investigated in [5]). The activation in these voxels may be different at each given point in time. Intuitively, that means the same stimulus may be described by different hidden process parameters in different voxels. However, certain groups of voxels that are close together often exhibit similarly shaped time series, but with different amplitudes. In this case, we believe it is reasonable to assume that the underlying hidden processes corresponding to these voxels are proportional to one another. Our experiments will prove this assumption helps learn better models than the ones that choose to ignore it.

Formally, we say that the voxels X^1, \dots, X^V “share” their corresponding Hidden Process parameters if there exist base processes P_1, \dots, P_K and constants c_{kv} for $1 \leq v \leq V$ such that:

$$X_{nt}^v \sim N\left(\sum_k c_{kv} \cdot P_{k(t-t_{nk}+1)}, \sigma^2\right)$$

and the activations in these voxels are independent given the parameters of the model. σ^2 represents the measurement variance which is also assumed shared across these voxels.

We consider how to efficiently perform Maximum Likelihood estimation of the parameters of the voxels X^1, \dots, X^V , assuming they share their corresponding Hidden Process Model parameters as described above. The parameters to be estimated are the base process parameters $P = \{P_{kt} | 1 \leq k \leq K, 1 \leq t \leq T\}$, the scaling constants $C = \{c_{kv} | 1 \leq k \leq K, 1 \leq v \leq V\}$ (one for each voxel and process) and the common measurement variance σ^2 . Recall that N represents the number of observations. The log-likelihood of the model is:

$$\begin{aligned} l(P, C, \sigma) &= \frac{-NTV \cdot \log 2\pi}{2} - NTV \cdot \log \sigma - \frac{1}{2\sigma^2} \sum_{n,d,v} (x_{nt}^v - \sum_k c_{kv} \cdot P_{k(t-t_{nk}+1)})^2 \\ &= \frac{-NTV \cdot \log 2\pi}{2} - NTV \cdot \log \sigma - \frac{1}{2\sigma^2} \cdot l'(P, C) \end{aligned}$$

It is easy to see that the function l' does not depend on the variance σ^2 and it is a sum of squares, where the quantity inside each square is a linear function in both P and C . Based on this observation, in Algorithm 1 we describe a method to compute the Maximum Likelihood estimators for the parameters that are shared across the voxels in our set.

Let \bar{X} be the column vector $\{x_{nt}^v\}$. Start with (\hat{P}, \hat{C}) an initial random guess, then repeat Steps 1 and 2 until they converge to the minimum of the function $l'(\hat{P}, \hat{C})$.

STEP 1. Write $l'(\hat{P}, \hat{C}) = \|A \cdot \hat{P} - \bar{X}\|^2$ where A is a $N \times V$ by $K \times T$ matrix depending on the current estimate \hat{C} of the scaling constants. Minimize with respect to \hat{P} using ordinary Least Squares to get a new estimate $\hat{P} = (A^T \cdot A)^{-1} \cdot A^T \cdot \bar{X}$.

STEP 2. Minimize l' with respect to \hat{C} same as in Step 1.

STEP 3. Once convergence is reached by repeating the above two steps, let $\hat{\sigma}^2 = \frac{l'(\hat{P}, \hat{C})}{NVT}$.

Algorithm 1. Maximum Likelihood estimators for voxels which share Hidden Process parameters

It might seem that this is a very expensive algorithm because it is an iterative method. However, we found that when applied to fMRI data in our experiments, it usually converges in 3-5 repetitions of Steps 1 and 2. We believe that the main reason why this happens is because at each partial step during the iteration we compute a closed form global minimizer on either P or C instead of using a potentially expensive gradient descent algorithm. In Section 4 we experimentally show the benefits of our algorithm over methods that do not utilize parameter sharing assumptions.

One may suspect that it is easy to learn the parameters of the above model because it is a particular case of bilinear model. However, this is not the case. In the bilinear model representation [6], the style matrices will correspond to process parameters and the content vectors will correspond to scaling constants. It is easy to see that in our case the style matrices have common pieces, depending on when the processes started in each example. Therefore, the SVD method presented in [6] that assumes independence of these style matrices is not appropriate in our problem. In [7] we present a more general framework that incorporates parameter sharing constraints in learning of Bayes Nets.

In general it is very difficult to specify a priori which voxels share their Hidden Process Model parameters. Algorithm 2 introduces *Hierarchical Hidden Process Models*, which use a nested cross-validation hierarchical approach to both come up with a partition of the brain in clusters of voxels that share their Hidden Process Model parameters and estimate these parameters at the same time.

STEP 1. Split the 40 examples into a set containing 20 folds $F = \{F_1, \dots, F_{20}\}$, each fold containing one example where the sentence is presented first and one example where the picture is presented first.

STEP 2. For all $1 \leq k \leq 20$, keep fold F_k aside and learn a model from the remaining folds using Steps 3-5.

STEP 3. Start with a partition of all voxels in the brain by their ROIs and mark all subsets as *Not Final*.

STEP 4. While there are subsets in the partition that are *Not Final*, take any such subset and try to split it using equally spaced planes in all three directions (in our experiments we split each subset into $16 = 4 \times 2 \times 2$ smaller subsets). If the cross-validation Average Log Likelihood of the model learned from these new subsets using Algorithm 1 (based on folds $F \setminus F_k$) is lower than the cross-validation Average Log Likelihood of the initial subset for folds in $F \setminus F_k$, then mark the initial subset as *Final* and discard its subsets. Otherwise remove the initial subset from the partition and replace it with its subsets which then mark as *Not Final*.

STEP 5. Given the partition computed by STEPS 3 and 4, based on the 38 data points in $F \setminus F_k$, learn a Hidden Process Model that is shared for all voxels inside each subset of the partition. Use this model to compute the log score for the examples/trials in F_k .

STEP 6. In Steps 2-4 we came up with a partition for each fold F_k . To come up with one single model, compute a partition using STEPS 3 and 4 based on all 20 folds, then, based on this partition learn a model as in STEP 5 using all 40 examples. The Average Log Score of this last model can be estimated by averaging the numbers obtained in STEP 5.

Algorithm 2. Hierarchical Hidden Process Model learning

4. EXPERIMENTS

We model the activity in the brain using a Hidden Process Model with two processes, corresponding to the cognitive processes of comprehending a *Sentence* or a *Picture*. The start time of each processes is assumed to be known in advance (i.e., we assume the process begins immediately upon seeing the sentence or picture stimulus). The true underlying distribution of voxel activation is not known and also data is scarce, which prevents us from keeping a large held-out test. Therefore we estimated the performance of our three models using the Average Log Likelihood, based on a leave two out cross-validation approach, where each fold contains one example in which the sentence is presented first, and one example in which the picture is presented first.

In our experiments we compare three HPM models. The first model, *StHPM*, which we consider a baseline, consists of a standard Hidden Process Model learned independently for each voxel. The second model, *ShHPM*, is a Hidden Process Model where all voxels in a ROI share their Hidden Process parameters. *ShHPM* is learned using Algorithm 1. The third model, *HieHPM* is a Hierarchical Hidden Process Model learned using Algorithm 2.

Trials	StHPM	ShHPM	HieHPM	Cells
6	-30497	-24020	-24020	1
8	-26631	-23983	-23983	1
10	-25548	-24018	-24018	1
12	-25085	-24079	-24084	1
14	-24817	-24172	-24081	21
16	-24658	-24287	-24048	36
18	-24554	-24329	-24061	37
20	-24474	-24359	-24073	37
22	-24393	-24365	-24062	38
24	-24326	-24351	-24047	40
26	-24268	-24337	-24032	44
28	-24212	-24307	-24012	50
30	-24164	-24274	-23984	60
32	-24121	-24246	-23958	58
34	-24097	-24237	-23952	61
36	-24063	-24207	-23931	59
38	-24035	-24188	-23921	59
40	-24024	-24182	-23918	59

Table 1. The effect of training set size on the Average Log Likelihood of the three models in CALC (Visual Cortex)

Our first set of experiments, summarized in Table 1, compares the three models based on their performance in the Visual Cortex (CALC). This is one of the ROIs actively involved in this cognitive task and it contains 318 voxels. The training set size was varied from 6 examples to all 40 examples, in multiples of two. Sharing parameters for groups of voxels proved very beneficial and the impact was best observed when the training set size was the smallest. With an increase in the number of examples, the performance of *ShHPM* starts to degrade because it makes the biased assumption that all voxels in CALC share their corresponding parameters. While this assumption paid off with small training set size because of the reduction in variance, it definitely hurt in terms of bias with larger sample size. However, we will see in other experiments that in certain ROIs, this assumption holds and in those cases the gains in performance may be quite large.

As expected, the hierarchical model *HieHPM* performed better than both *StHPM* and *ShHPM* because it takes advantage of shared Hidden Process Model parameters while not making the restrictive assumption of sharing across all voxels in a ROI. The largest difference in performance between *HieHPM* and *StHPM* is observed at 6 examples, in which case *StHPM* basically fails to learn a reasonable model while the highest difference between *HieHPM* and *ShHPM* occurs at the maximum number of examples, presumably when the bias of *ShHPM* is most harmful. As the number of training examples increases, both *StHPM* and *HieHPM* tend to perform better and better and one can see that the marginal improvement in performance obtained by the addition of two new examples tends to shrink as both models approach convergence. While with an infinite amount of data, one would expect *StHPM* and *HieHPM* to converge to the true model, at 40 examples,

HieHPM still outperforms the baseline model *StHPM* by a difference of 106 in terms of Average Log Likelihood, which is an improvement of e^{106} in terms of data likelihood.

Probably the measure that shows best the improvement of *HieHPM* over the baseline *StHPM* is the number of examples needed by *StHPM* to achieve the same performance as *HieHPM*. It turns out that on average, *StHPM* needs roughly 2.9 times the number of examples needed by *HieHPM* in order to achieve the same level of performance in the Visual Cortex (CALC).

The last column of Table 1 displays the number of clusters of voxels in which *HieHPM* partitioned CALC. As can be seen, at small sample size *HieHPM* draws its performance from reductions in variance by using only one cluster of voxels. However, as the number of examples increases, *HieHPM* improves by finding more and more refined partitions. This number of shared voxel sets tends to stabilize around 60 clusters once the number of examples reaches 30, which yields an average of more than 5 voxels per cluster given that CALC is made of 318 voxels. For a training set of 40 examples, the largest cluster has 41 voxels while many clusters consist of only one voxel.

ROI	Voxels	StHPM	ShHPM	HieHPM	Cells
CALC	318	-24024	-24182	-23918	59
LDLPFC	440	-32918	-32876	-32694	11
LFEF	109	-8346	-8299	-8281	6
LIPL	134	-9889	-9820	-9820	1
LIPS	236	-17305	-17187	-17180	8
LIT	287	-21545	-21387	-21387	1
LOPER	169	-12959	12909	-12909	1
LPPREC	153	-11246	-11145	-11145	1
LSGA	6	-441	-441	-441	1
LSPL	308	-22637	-22735	-22516	4
LT	305	-22365	-22547	-22408	18
LTRIA	113	-8436	-8385	-8385	1
RDLPFC	349	-26390	-26401	-26272	40
RFEF	68	-5258	-5223	-5223	1
RIPL	92	-7311	-7315	-7296	11
RIPS	166	-12559	-12543	-12522	20
RIT	278	-21707	-21720	-21619	42
ROPER	181	-13661	-13584	-13584	1
RPPREC	144	-10623	-10558	-10560	1
RSGA	34	-2658	-2654	-2654	1
RSPL	252	-18572	-18511	-18434	35
RT	284	-21322	-21349	-21226	24
RTRIA	57	-4230	-4208	-4208	1
SMA	215	-15830	-15788	-15757	10
All Brain	4698	-352234	-351770	-350441	299

Table 2. Performance of the three models over whole brain and in several ROIs when learned using all 40 examples

The second set of experiments (see Table 2) describes the performance of the three models for each of the 24 individual ROIs of the brain, and also when trained over the entire brain. While we have seen that *ShHPM* was biased in CALC, we see here that there are several ROIs where it

makes sense to assume that all voxels share their Hidden Process Model parameters. In fact, in most of these regions, *HieHPM* finds only one cluster of voxels. Actually, *ShHPM* outperforms the baseline model *StHPM* in 18 out of 24 ROIs while *HieHPM* outperforms *StHPM* in 23 ROIs. One may ask how *StHPM* can possibly outperform *HieHPM* on a ROI, since *HieHPM* may also represent the case when there is no sharing. The explanation is that the hierarchical approach can get stuck in a local maximum of the data log-likelihood over the search space if it cannot improve by splitting at a specific step, since it is a greedy process that does not look beyond that split for a finer grained partition. Fortunately, this problem is very rare in our experiments.

Over the whole brain, *HieHPM* outperforms *StHPM* by 1792 in terms of log likelihood while *ShHPM* outperforms *StHPM* only by 464. The main drawback of the *ShHPM* is that it also makes a very restrictive sharing assumption and therefore *HieHPM* emerges as the recommended approach.

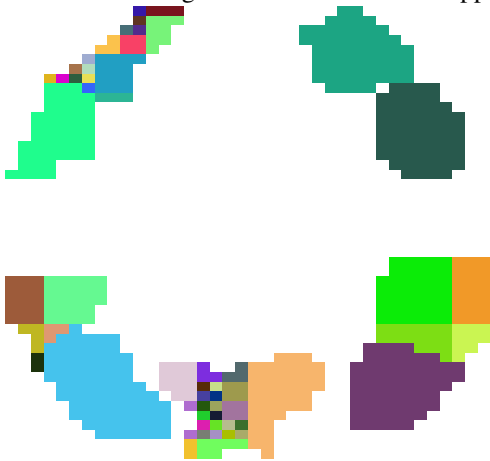


Figure 2. Clusters learned by *HieHPM* in slice 5 of the brain

As mentioned above, *HieHPM* automatically learns clusters of voxels that share their Hidden Process Model parameters. Figure 2 shows the portions of these learned clusters in slice five of the eight vertical slices that make up the 3D brain image captured by the fMRI scanner. Neighboring voxels that were assigned by *HieHPM* to the same cluster are pictured with the same color. Note that there are several very large clusters in this picture. This may be because of the fact that it makes sense to share voxel parameters at the level of an entire specific ROI if the cognitive process does not activate voxels in this ROI. However, large clusters are also found in areas like CALC, which we know is directly involved in visual processing.

In Figure 3 we present the learned *Sentence* hidden process for the voxels in slice five of the Visual Cortex (CALC). The graphs corresponding to voxels that belong to the same cluster have been painted in the same color as in Figure 2. To make these graphs readable, we only plotted the base processes, disregarding the scaling constants corresponding to each voxel within a given cluster.

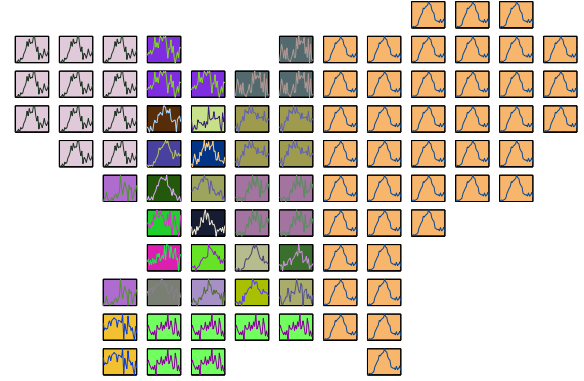


Figure 3. Learned *Sentence* process for all voxels in slice 5 of the Visual Cortex (CALC)

5. CONCLUSIONS

We have presented an approach to improving the accuracy of learned models of hidden cognitive processes from fMRI data. Building on earlier work on Hidden Process Models, we develop an algorithm to automatically discover sets of spatially contiguous voxels whose HPM parameters can be shared. We also present an algorithm for learning subject to these parameter-sharing constraints. Experiments show that this algorithm learns models with shared parameters that outperform our original HPM algorithms in terms of cross-validated data log likelihood of the resulting model.

Directions for future work include accommodating uncertainty in the onset times of the hidden processes, exploring clusters of different shapes, as well as using data from multiple human subjects.

Acknowledgements: This research was supported by funding from Siemens, NSF grants CCR-0085982, CCR-0122581, Darpa PAL program NBCD030010, and a grant from W.M. Keck Foundation.

REFERENCES

- [1] Mitchell TM, Hutchinson R, Just M, Newman S, Niculescu RS, Pereira F, et al. Learning to decode cognitive states from brain images. *Machine Learning*. 2004;57(1-2):145-175.
- [2] Carpenter PA, Just MA, Keller TA, Eddy WF, Thulborn KR. Time course of fMRI-activation in language and spatial networks during sentence comprehension. *NeuroImage*. 1999;10:216-224.
- [3] Hutchinson R, Mitchell TM, Rustandi I. Hidden process models. Carnegie Mellon University, February 2006. Report No: CS-CALD-05-116.
- [4] Dale AM. Optimal experimental design for event-related fMRI. *Human Brain Mapping*. 1999;8:109-114.
- [5] Mitra PS. Automated Knowledge Discovery from Functional Magnetic Resonance Images using Spatial Coherence. PhD Thesis, University of Pittsburgh, July 2006.
- [6] Tenenbaum JB, Freeman WT. Separating style and content with bilinear models. *Neural Computation*. 2000;12(6):1247-1283.
- [7] Niculescu RS. Exploiting parameter domain knowledge for learning in bayesian networks. Carnegie Mellon University, 2005. Report No: CMU-TR-05-147.