

Multi-sensor Data Fusion Using the Influence Model

Wen Dong and Alex Pentland

E15-383, 20 Ames Street
The MIT Media Laboratory
Cambridge, MA, 02139-4307
{wdong, sandy}@media.mit.edu

Abstract

System robustness against individual sensor failures is an important concern in multi-sensor networks. Unfortunately, the complexity of using the remaining sensors to interpolate missing sensor data grows exponentially due to the “curse of dimensionality”. In this paper we demonstrate that the influence model, our novel formulation for combining evidence from multiple interactive dynamic processes, can efficiently interpolate missing data and can achieve greater accuracy by modeling the structure of multi-sensor interaction.

1. Introduction

Multi-sensor networks with tens or even hundreds of sensors are common nowadays. One example of a wearable sensor network is the Groupware systems developed for the DARPA ASSIST program [1]. In this system, data is collected in real-time from several accelerometers, microphones, cameras, and a GPS, all attached to different parts of the soldiers’ clothing. Inference of soldier state is made in real-time, and data automatically shared among different soldiers wearing the Groupware systems based on the pattern of activity shown among the group of soldiers.

A couple of observations from using the Groupware system are worth mentioning. One observation is that sensor failures are unavoidable due to insufficient power supply, sensor faults, connection errors, or other unpredictable causes. This means that an inference algorithm for soldier state must be robust against sensor failures. Another observation is that the data collected from different sensors are often correlated. For example, when the GPS data indicate that a particular person is indoors, this person is also often in a sitting position. This correlation happens in a large combinatorial space. In an early Groupware system [2] with 8 locations, 6 speaking/non-speaking status, 7 postures, and 8

activities, there were $8 \times 6 \times 7 \times 8 = 2688$ number of different combinatorial states. As a result, a good algorithm must be able to determine the structural relations between different sensors while avoiding consideration of the exponential number of combinatorial states.



Figure 1. An early Groupware System.

Another example of a body-worn sensor network is the cellphone network of the participants of the reality mining project [3]. In this project the 81 participants’ locations, their proximity information, their cellphone usages, and their cellphone states were continuously collected over a period of nine months. As in the Groupware example, the cellphone data collection for the individuals experienced several different types of abnormalities, and the cellphone data for different individuals are correlated. Thus a good data mining algorithm should be robust to abnormalities, and be able to discover the structural relationships among different participants.

We believe that our latent structure influence model is an efficient, robust method for analysing these sort of multi-sensor dynamics problems. It is in the tradition of N-heads dynamic programming on coupled hidden Markov models [4], the observable structure influence model [5], and the partially observable influence model [6], but extends these

previous models by providing greater generality, accuracy, and efficiency.

2. Influence modeling of multi-sensor dynamics

The influence model is a tractable approximation of the intractable hidden Markov modeling of multiple interacting dynamic processes. While the number of states of the hidden Markov model is the multiplication of the number of states for individual processes, the number of states of the corresponding influence model is the summation of the number of states for individual processes. The influence model attains this tractability by linearly combining the contributions of latent state distributions of individual processes at time $t - 1$ to get the latent state distributions of individual processes at time t .

Let us assume that we have C interacting stochastic processes and $m_c (1 \leq c \leq C)$ number of latent states corresponding to process c in the system's behavior space. Following Asavathiratham [5] we use $D_{C \times C}$ as the network (influence) matrix, whose columns each add up to 1, and $A^{(c_1, c_2)} (1 \leq c_1, c_2 \leq C)$ as the inter-process state transition matrix, whose rows adds up to 1. The influence matrix is defined as the Kronecker product $H = D \otimes A = (d_{c_1, c_2} A^{(c_1, c_2)})$ where H is a block matrix whose sub-matrix at row c_1 and column c_2 is $d_{c_1, c_2} A^{(c_1, c_2)}$. We express the marginal probabilistic distributions of states $s_t^{(c)}$ for process c at time t as a row vector consisting of row vectors $p(\vec{s}_t) = (p(\vec{s}_t^{(1)}), p(\vec{s}_t^{(2)}), \dots, p(\vec{s}_t^{(C)}))$, where $p(\vec{s}_t^{(c)}) = (p(s_{t,1}^{(c)}), p(s_{t,2}^{(c)}), \dots, p(s_{t,m_c}^{(c)}))$, $1 \leq c \leq C$, $\sum_j p(s_{t,j}^{(c)}) = 1$ is the row vector of the probability distribution on each latent state. Using this notation, the influence process is then $p(\vec{s}_{t+1}) = p(\vec{s}_t) \cdot H$, $p(\vec{s}_1) = \vec{\pi}$, so that the marginal distributions at the next time step is a linear combination of the marginal distributions in the current time step. There are many ways to map from an influence matrix H to a master Markov matrix $G = (g_{i,j}), g_{s_t^{(1)} s_t^{(2)} \dots s_t^{(C)} \rightarrow s_{t+1}^{(1)} s_{t+1}^{(2)} \dots s_{t+1}^{(C)}} = \prod_c \sum_{c_1} h_{s_{t+1}^{(c_1)}, s_t^{(c)}}$. We follow Asavathiratham's idea (see [5] sec. 5.11) in our paper. The observations $o_t^{(c)}$ for process c at sample times $1 \leq t \leq +\infty$ can be in multinomial or Gaussian distribution and is statistically determined by the corresponding latent state $s_t^{(c)}$. When the observations for process c are multinomial, we use n_c to represent the number of observation symbols $o_t^{(c)} \in [1 \dots n_c]$ and use $B^{(c)} = (b_{i,j}^{(c)})$ to represent the observation matrix $p(\vec{o}_t^{(c)}) = p(\vec{s}_t^{(c)}) \cdot B^{(c)}$. When the observations for process c are Gaussian, we use n_c to represent the dimensionality of the m_c number of Gaussian distributions corresponding to each latent state $s_t^{(c)} \in [1 \dots m_c]$. The EM algorithm for latent state esti-

mation is given as follows:

$$\begin{aligned} \vec{\alpha}_t^* &= \begin{cases} \vec{\pi}_{1 \times \sum m_c} & t = 1 \\ \vec{\alpha}_{t-1} \cdot H \cdot \text{diag}[\vec{b}_t] & t > 1 \end{cases} \\ \mathcal{N}_t &= \begin{pmatrix} \frac{1}{\sum_{i=1}^{m_1} \vec{\alpha}_{t,i}^{*(1)}} & & \\ & \dots & \\ & & \frac{1}{\sum_{i=1}^{m_C} \vec{\alpha}_{t,i}^{*(C)}} \end{pmatrix} \\ \vec{\alpha}_t &= \vec{\alpha}_t^* \cdot \mathcal{N}_t \\ \vec{\beta}_t &= \begin{cases} \vec{1}_{\sum m_c \times 1} & t = T \\ H \cdot \text{diag}[\vec{b}_t] \mathcal{N}_{t+1} \vec{\beta}_{t+1} & t < T \end{cases} \\ \vec{\gamma}_t &= \vec{\alpha}_t \cdot \text{diag}[\vec{\beta}_t] \\ \xi_{t-1 \rightarrow t} &= \text{diag}[\vec{\alpha}_{t-1}] \cdot H \cdot \text{diag}[\vec{b}_t] \cdot \mathcal{N}_t \cdot \text{diag}[\vec{\beta}_t] \\ p(\vec{o}) &= \prod_{t,c} \left(\sum_{i=1}^{m_c} \vec{\alpha}_{t,i}^{*(c)} \right) \end{aligned}$$

The parameters are re-estimated as follows:

$$\begin{aligned} A_{ij} &= \text{normalize} \left[\sum_{t=2}^T \xi_{t-1 \rightarrow t}^{(i,j)} \right] \\ S &= \begin{pmatrix} \vec{1}_{1 \times m_1} & & \\ & \dots & \\ & & \vec{1}_{1 \times m_C} \end{pmatrix} \\ d_{ij} &= \text{normalize} \left[S \sum_{t=2}^T \xi_{t-1 \rightarrow t} S^T \right] \\ \vec{\pi}^{(c)} &= \text{normalize}[\vec{\gamma}_1^{(c)}] \\ p^{(c)}(\vec{o}^{(c)} | \vec{s}^{(c)}) &= \text{normalize} \left[\sum_t \vec{\gamma}_t^{(c)T} \cdot \vec{\delta}_{y_t, \vec{o}^{(c)}} \right] \\ \mu^{(c)} &= \frac{\sum_t \vec{\gamma}_t^{(c)T} \cdot \vec{o}_t^{(c)}}{\sum_t \vec{\gamma}_t^{(c)} \cdot \vec{1}_{m_c \times 1}} \\ \sigma_i^{(c)2} &= \frac{\sum_t \gamma_{t,i}^{(c)} \vec{o}_t^{(c)T} \vec{o}_t^{(c)}}{\sum_t \vec{\gamma}_t^{(c)} \cdot \vec{1}_{m_c \times 1}} - \vec{\mu}_i^{(c)T} \cdot \vec{\mu}_i^{(c)} \end{aligned}$$

3. Experimental Results

In this section, we illustrate how an influence model can capture the correlations among different dynamic processes and thus improve the overall classification performance. The noisy body sensor net example illustrates the structure that the influence model tries to capture, and how an influence model can be used to improve classification precision. We then extend the noisy body sensor net example and compare the training errors and the testing errors of different

dynamic models. Finally, we show a wearable sensor net example in which the goal is real-time context recognition, and the influence model is used to discover hidden structure among speech, location, activity, and posture signals in order to allow for more accurate and robust classification of the wearers state.

3.1. Noisy body sensor net example

In the noisy body sensor net example, we have six stochastic processes, and we sample these six processes with six body sensors. Each process can be either signaled (one) or non-signaled (zero) at any time, and the corresponding body sensor has approximately 10% of its samples flipped. The interaction of the six stochastic processes behind the scene looks like this: processes one through three tend to have the same states; processes four through six tend to have the same states; the processes are more likely to be non-signaled than to be signaled; and the processes tend to stick to their states for a stretch of time. The parameters of the model are given as the following and are going to be estimated: $A_{ij} = \begin{pmatrix} .99 & .01 \\ .08 & .92 \end{pmatrix}, 1 \leq i, j \leq 6,$

$B_i = \begin{pmatrix} .9 & .1 \\ .1 & .9 \end{pmatrix}, 1 \leq i \leq 6, d_{ij} = .33, 1 \leq i, j \leq 3,$ and $d_{ij} = .33, 4 \leq i, j \leq 6.$

In Figure 2, (a) shows the sampled latent state sequences, (b) shows the corresponding observation sequences, (c) shows the influence matrix reconstructed from sampled observation sequences, and (d) shows the reconstructed latent state sequences after 300 observations. The $(i, j)^{th}$ entry of the $(c_1, c_2)^{th}$ sub-matrix of an influence matrix determines how likely that process c_1 is in state i at time t and process c_2 is in state j at time $t + 1$. It can be seen from Figure 2 (c) that the influence model computation recovers the structure of the interaction.

The influence model can normally attain around 95% accuracy in predicting the latent states for each process. The reconstructed influence matrix has only 9% relative differences with the original one. Using only observations of other chains we can predict a missing chain’s state with 87% accuracy.

3.2. Comparison of dynamic models

The training errors and the testing errors of the coupled hidden Markov model, the hidden Markov model, and the influence model are compared in this example. The setup of the comparison is described as the following. We have a Markov process with 2^C , where $C = 10$, number of states and a randomly generated state transition matrix. Each system state \vec{s}_t is encoded into a binary $s_t^{(1)} \dots s_t^{(C)}$. Each of

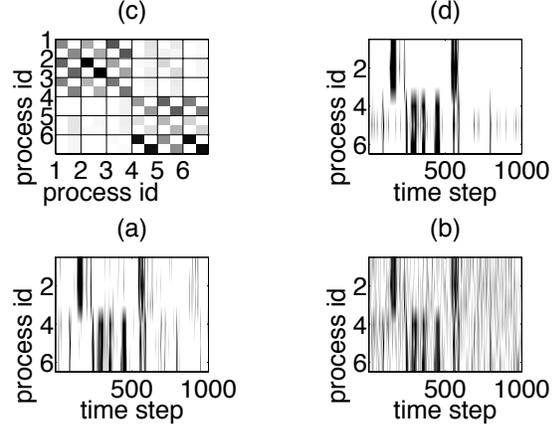


Figure 2. Inference from observations of interacting dynamic processes.

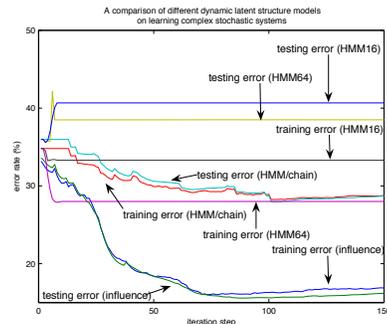


Figure 3. Comparison of dynamic models.

the $m_c = 2$ evaluations of “digit” $s_t^{(c)}$ corresponds a different 1-d Gaussian observation $o_t^{(c)}$: Digit $s_t^{(c)} = 1$ corresponds to $o_t^{(c)} \sim \mathcal{N}[\mu_1 = 0, \sigma_1^2 = 1]$; Digit $s_t^{(c)} = 2$ corresponds to $o_t^{(c)} \sim \mathcal{N}[\mu_2 = 1, \sigma_2^2 = 1]$.

In most real sensor nets we normally have redundant measures and an insufficient observations to accurately characterize sensor redundancy using standard methods. Figure 3 compares the performances of several dynamic latent structure models applicable to multi-sensor systems. Of the 1000 samples $(\vec{o}_t)_{1 \leq t \leq 1000}$, we use the first 250 for training and all 1000 for validation.

There are two interesting points. First, the logarithmically scaled number of parameters of the influence model allows us to attain high accuracy based on a relatively small number of observations. This is because the eigenvectors of the master Markov model we want to approximate are either mapped to the eigenvectors of the corresponding influence model, or mapped to the null space of the corresponding event matrix thus is not observable from the influence model, and that in addition the eigenvector with the

largest eigenvalue (i.e., 1) is mapped to the eigenvector with the largest eigenvalue of the influence matrix [5]. Secondly, both the influence model and the hidden Markov model applied to individual processes are relatively immune to overfitting, at the cost of low convergence rates. This situation is intuitively the same as the numerical analysis wisdom that a faster algorithm is more likely to converge to a local extremum or to diverge.

3.3. Real-time Context Recognition

An early version of the Groupwear real-time context recognition system was developed by Blum [2], and is comprised of a Sharp Zaurus PDA, an ambient audio recorder, and two accelerometers, worn on hip and wrist (see Figure 1). This system is designed to classify in real time eight locations, six speaking/non-speaking status, six postures, and eight activities. The classification is carried out in two steps: A pre-classifier (single Gaussian, mixture of Gaussians, or C4.5) is first invoked on the audio and accelerometer features to get a moderate pre-classification result of the above four categories.

The pre-classification result of different categories is then fed into an influence model to learn inter-sensor structure, and then this learned structure is used to generate an improved post-classification result. In this example the influence model learns the conditional probabilities that relate the four categories (location, audio, posture, and activity) and then uses this learned influence matrix to improve the overall performance. For example, given that the Groupwear user is typing, we can inspect the row of the influence matrix corresponding to “typing” and see that this person is very likely to be either in the office or at home, to not be speaking, and to be sitting. As a result, the action of typing can play a critical role to disambiguating confusions between sitting and standing, or between speaking vs not-speaking, but not between office and home.

By combining evidence across different categories using the influence model, the classification errors for locations, speaking/non-speaking, postures, and activities decreased by an average of 23%, from 38%, 22%, 8% and 27% to 28%, 19%, 8%, and 17% respectively. The post-classification for postures does not show significant improvement because of two reasons: (1) it is already precise enough considering that we have labeling imprecision in our training data and testing data, and (2) it is the driving force for improving the other categories, and no other categories are more certain than the posture category.

4. Conclusion

In this paper, we have presented the formulation of a latent structure influence model, given the parameter learning

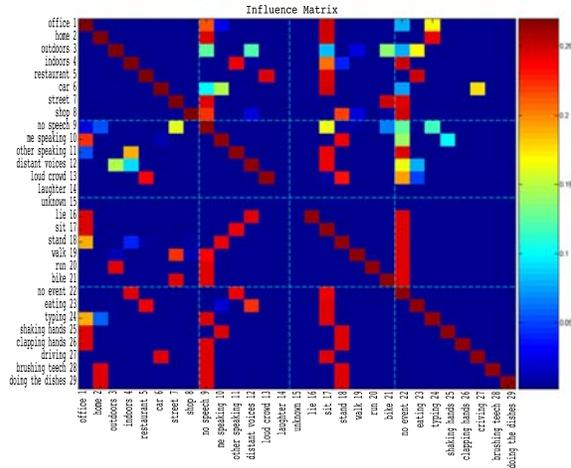


Figure 4. Influence matrix learned by the EM algorithm.

and latent state estimation algorithms, and demonstrated the latent structure influence model’s superior performance in multi-sensor network analysis.

References

- [1] 2004. URL http://www.darpa.mil/ipto/solicitations/open/04-38_PIP.htm.
- [2] Mark Blum. Real-time context recognition. Master’s thesis, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich (ETH), 2005.
- [3] Nathan Eagle and Alex Pentland. Reality mining: Sensing complex social systems. *Journal of Personal and Ubiquitous Computing*, 2005.
- [4] Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8):831–843, 2000.
- [5] Chalee Asavathiratham. *The Influence Model: A Tractable Representation for the Dynamics of Networked Markov Chains*. PhD thesis, MIT, 1996.
- [6] Sumit Basu, Tanzeem Choudhury, Brian Clarkson, and Alex Pentland. Learning human interactions with the influence model. Technical report, MIT Media Laboratory Vision & Modeling Technical Report #539, 2001. URL <http://vismod.media.mit.edu/tech-reports/TR-539.pdf>.