PSEUDOCODE and IMPLEMENTATION DETAILS for three-layered GSM’s

Document written by Nikolaos Mavridis, May 2005

In this document:
1) Notation is introduced.
2) The basic GSM operational cycle is described.
3) The functions comprising the GSM operational cycle are analyzed in pseudocode, and various options are discussed.
Note: A full Bayesian treatment might also be attempted (not described here).

Read in conjunction with the paper:
“Grounded Situation Models for Robots: Bridging Language, Perception and Action”
Paper authored by N. Mavridis and D. Roy
And submitted to the:
AAAI-05 workshop on modular construction of humanlike intelligence

Notation
Assume a property dimension (for example, color) with M-dim measurements (M=3 for color), and N verbal categories (N=8 for Red, Green, Blue, Cyan, Magenta, Yellow, White, Black)

Stochastic Layer (L1):
Continuous-variable M-dimensional distribution
p(x), where x is an M-dim vector
Might be approximated by discrete-variable M-dimensional distribution, with N1 x N2 bins, covering the expected range of values (rectangular bins by default – other partitions possible too)

Categorical Layer (L3):
Discrete-variable 1-dimensional distribution
P(C), where C belongs to 1…N

Continuous Layer (L2):
xcont = M-dimensional vector, continuous-valued

Classifier:
Function C = F(x)
where C is integer 1…N, and x is M-dimensional vector (i.e. Verbal Category C = F(measurement value x) might be explicitly known, or realized as neural net, SVM, nearest-neighbor etc.)
Operational Cycle
(shorthand notation explained below)

InitL1L2L3
DoForever {
If (incoming sensory data)
   Senses-to-L1
   L1-to-L2
   L1-to-L3
Else If (incoming speech data)
   Words-to-L3
   L3-to-L1
   L1-to-L2
Else (no incoming data – diffuse)
   Diffuse-L1
   L1-to-L2
   L1-to-L3
Endif

If (incoming request for words)
   L3-to-words
Else If (incoming request for action parameter)
   L2-to-actionparams
Endif
}

In our current modular implementation,
These operations are distributed across modules:
VISOR and PROPRIOCEPTOR contribute sensory data
IMAGINER contributes speech data
INQUIRER and REMEMBERER create words and action parameters
GSM diffuses data that have not been updated

InitL1L2L3) Initialise Distributions

Stochastic Layer:
p(x) = ?
either:
1) Non-informative prior
   For example, uniform distribution covering the expected range of values
or:
2) Empirically derived prior
   Accumulate experience from colors of objects seen so far
   I.e. estimate distribution by all the incoming x’s of the past sensory experiences

Categorical Layer:
P(c) = ?
Fed by stochastic layer through the stochastic-to-categorical feeding process (see below)

Continuous Layer:
Xcont = ?
Fed by stochastic layer through the stochastic-to-continuous feeding process (see below)
L1-to-L3) Stochastic-to-Categorical feeding process

For C=1…N (i.e. iterate through verbal categories)

\[ P(C) = \text{Integral of } p(x) \text{ over those } x \text{-values such that } F(x) = C \]

(In the case of discrete-variable binned approximation to \( p(x) \),
this integral can be simply approximated by iterating over all bins,
and summing the probability contribution of only those bins
whose center \( x \) indeed gives \( F(x) = C \) for the particular verbal
category \( C \) under consideration, i.e.

\[
\text{sum} = 0 \\
\text{For } I1 = 1…N1 \ (\text{iterate over bins}) \\
\quad \text{For } I2 = 1…N2 \\
\quad \quad \quad \text{…} \\
\quad \quad \quad \quad \text{If } F(\text{center}(I1, I2, \ldots)) \equiv C \\
\quad \quad \quad \quad \quad \text{sum} = \text{sum} + p(I1,I2,\ldots) \\
\quad \quad \quad \quad \quad \text{endif} \\
\quad \quad \quad \quad \text{…} \\
\quad \quad \text{Next I2} \\
\quad \text{Next I1} \\
\quad P(C) = \text{sum} \\
\) \\

Next C

L3-to-L1) Categorical-to-Stochastic feeding process

For C=1…N (i.e. iterate through verbal categories)

Either:

1) Non-informative prior

Distribute \( P(C) \) over \( p(x) \) for those \( x \)-values such that \( F(x) = C \),
so that the integral of \( p(x) \) over those \( x \)-values is equal to \( P(C) \).

(One reasonable choice of how to distribute: distribute uniformly,
i.e. set \( p(x) \) for those \( x \)-values such that \( F(x) = C \) equal
to \( P(C) / \text{Area(region of } x \text{'s such that } F(x) = C \) in the case of discrete-variable binned approximation to \( p(x) \): Find number of bins such that \( F(\text{center}(I1,I2,\ldots)) = C \), and call it \( Nc \).
And then for those bins assign \( p(I1,I2,\ldots) = P(C)/Nc \), i.e. (of course the code found below can be optimized much more!)

\[
Nc = 0 \\
\text{For } I1 = 1…N1 \ (\text{iterate over bins}) \\
\quad \text{For } I2 = 1…N2 \\
\quad \quad \quad \text{…} \\
\quad \quad \quad \quad \text{If } F(\text{center}(I1, I2, \ldots)) \equiv C \\
\quad \quad \quad \quad \quad Nc=Nc+1 \\
\quad \quad \quad \quad \quad \text{endif} \\
\quad \quad \quad \quad \text{…} \\
\quad \quad \text{Next I2} \\
\quad \text{Next I1} \\
\quad \text{For } I1 = 1…N1 \\
\quad \quad \text{For } I2 = 1…N2 \\
\quad \quad \quad \text{…} \\
\quad \quad \quad \quad \text{If } F(\text{center}(I1, I2, \ldots)) \equiv C \\
\quad \quad \quad \quad \quad p(I1,I2,\ldots) = P(C) / Nc \\
\quad \quad \quad \quad \quad \text{endif} \\
\quad \quad \quad \quad \text{…} \\
\quad \text{Next I2} \\
\]
Next I
Or:
2) Empirically-derived prior
Estimate \( p(x \mid \text{verbal label} = C) \) from human training data
Then weigh this estimate by \( P(C) \) and contribute accordingly to \( p(x) \).
This might in essence admit partial overlap of the measurement regions corresponding to different verbal categories.
Recall that as the classifiers perfectly partition the measurement space, we are silently admitting that such an overlap does not exist or that we can anyway tolerate the residual classification error that would result from this “fuzziness” of the category boundaries.

Next C

Senses-to-L1) Sensory measurement-to-Stochastic feeding process

Let’s assume incoming measurement \( x_{\text{new}} \)
Before the measurement, we had stochastic layer distribution \( p(x) = p_{\text{old}}(x) \)
After the measurement, the updated distribution will be \( p(x) = p_{\text{new}}(x) \)
Thus, we have to choose an update function \( f_{\text{upd}} \) such that:
\[
p_{\text{new}}(x) = f_{\text{upd}}(x_{\text{new}}, p_{\text{old}}(x)).
\]
A simple solution:
Assume “windowing function” \( w_{\text{new}}(x) \) (like parzen windows)
\( w_{\text{new}}(x) \) should be centered at \( x_{\text{new}} \) and have integral equal to one.
Choose \( \lambda \) belonging to \([0…1]\) (confidence in new measurement)
Then, define \( f_{\text{upd}} \) such that:
\[
p_{\text{new}}(x) = (1-\lambda) \cdot p_{\text{old}}(x) + \lambda \cdot w_{\text{new}}(x)
\]
In the case of discrete-variable binned approximation to \( p(x) \):
Assume discrete-variable binned window function \( w_{\text{new}}(I_1,I_2,…) \)
\( w_{\text{new}} \) should be centered at those \( I_1, I_2, …, \)
such that center\((I_1, I_2, …,)\) approx equal to \( x_{\text{new}} \)

// Find bin coordinates \((I_{xew}(1), I_{xew}(2),…)\)
// corresponding to incoming measurement \( x_{\text{new}} \)
For \( I = 1…M \)
   \( I_{xew}(I) = \text{MeasurementToIndices}(x_{\text{new}}) \)
Next I

//Create centered window function \( w_{\text{new}} \)
//with center at \((I_{xew}(1), I_{xew}(2),…)\)
//assume prototype window function \( w \) centered at \((I_{c1}, I_{c2},…)\)
For \( I_1 = 1…N_1 \) //iterate over bins
   For \( I_2 = 1…N_2 \)
      …
      //of course, care for non-negative matrix indices
      //should be taken here etc.
      \[ w_{\text{new}}(I_1,I_2,…) = w(I_1 - I_{xew}(1) + I_{c1}, \ldots) \]
      …
Next I_2
Next I_1

//Update stochastic layer by weigh-adding new window function
For \( I_1 = 1…N_1 \)
   For \( I_2 = 1…N_2 \)
      …
\[ p(I_1, I_2, \ldots) = (1-\lambda) \cdot p(I_1, I_2, \ldots) + \lambda \cdot w_{\text{new}}(I_1, I_2, \ldots) \]

... Next \( I_2 \)
Next \( I_1 \)

Other analytical or empirical solutions for \( f_{\text{upd}}(x_{\text{new}}, p_{\text{old}}(x)) \)
can of course be used too.

**Words-to-L3) Speech derived information-to-Categorical feeding process**

*Hard update:*
Here we are assuming that new verbal information totally overrides existing information (sensory/verbal) totally.

Either:
Assume incoming verbal category \( C_{\text{new}} \).
Then
\[ P(C_{\text{new}}) = 1 \]
And set all other \( P(C) \) such that \( C \neq C_{\text{new}} \) to be equal to zero

Or:
Assume incoming verbal category \( C_{\text{new}} \) with confidence \( \lambda \) belonging to \([0\ldots1]\)
Then
\[ P(C_{\text{new}}) = \lambda \]
And set all other \( P(C) \) such that \( C \neq C_{\text{new}} \) to be equal to \((1-\lambda) / (N-1)\)

Or:
Assume incoming verbal category distribution \( P_{\text{new}}(C) \)
Then just copy \( P_{\text{new}}(C) \) into \( P(C) \) for every \( C \) belonging to \( 1\ldots N \)

Or:
(if empirical data \( p(x \mid \text{verbal label} = C) \) available)
Set this to stochastic. Feed through to categorical

*Soft update:*
If we assume that new verbal information has to be blended with existing:

Either:
Assume incoming verbal category \( C_{\text{new}} \),
and confidence to new information = \( \lambda \) belonging to \([0\ldots1]\)
Then
\[ P(C_{\text{new}}) = (1-\lambda) \cdot P(C_{\text{new}}) + \lambda \]
And set all other \( P(C) \) such that \( C \neq C_{\text{new}} \) to be equal to \( P(C) = (1-\lambda) \cdot P(C) \)

Or:
(if empirical data \( p(x \mid \text{verbal label} = C) \) available, and confidence to new information = \( \lambda \) belonging to \([0\ldots1]\))
Update stochastic as:
\[ p_{\text{new}}(x) = (1-\lambda) \cdot p_{\text{old}}(x) + \lambda \cdot p(x \mid \text{verbal label} = C) \]
Feed through to categorical.

Various combinations and variants of the above can be used.
L1-to-L2) Stochastic-to-Continuous feeding process

Some averaging function chosen, accepting an M-D distribution p(x), and returning an M-D vector xcont
For example, one can choose mean, median etc.
xcont = ExpectedValue{p(x)}

L3-to-words) Categorical-to-description feeding process

Here we seek a function from a discrete-variable distribution to a set of templates.
Let’s assume verbalization of categories W(C) so that for example W(1) = “red” etc.
A decision tree with conditionals on the probability of the two most probable categories and the entropy is used, producing language generation templates such as:

If (Pmax1>Threshold1)
   Sentence = W(C such that P(C) = Pmax1)
Else If (Pmax1>Threshold2)
   Sentence = “Most probably ” + W(C such that P(C) = Pmax1)
Else If (Pmax1>Threshold3)
   Sentence = “Most probably ” + W(C such that P(C) = Pmax1) + “but maybe not”
Else If (Pmax1>.4 and Pmax2>.4)
   Sentence = “Either “ + W(C such that P(C) = Pmax1) + “or “ + W(C such that P(C) = Pmax2)
   …. (etc.)

Such a tree can be empirically learned,
Or some other empirically-trained generation model can be derived.

Also, an “inversion” of the above mapping can be used in feeding incoming descriptions such as “There is blue object most probably at the left” to the categorical layer (this was covered in the previous section – this is just an addendum to that section). Again one must use a simplifying assumption to overcome the one-to-many aspect of the inversion.

L2-to-actionparam) Continuous-to-action parameter feeding process

Just supply xcont as the action parameter.
For example Lookat(OBJi)
will need position(OBJi)
which will be taken by xcont of position of object i

DiffuseL1) Diffuse Stochastic Layer

Assume old distribution pold(x)
And new distribution pnex(x)
Choose update function fdiff such that:
pnew(x) = fdiff( pold(x) )

For example, for variance-parametrised distributions, increase the variance.
In the case of discrete-variable binned approximations of p(x),
One solution is to “low-pass filter” the distribution (in signal processing terms). In order not to get into filter theory, some simple solutions suffice. For example, one could just arbitrarily choose a rectangular window for the filter mask, and specify the rectangle lengths in bins. Then, the filter mask slides over the distribution, centered on each element (pixel, voxel etc.) in question. Instead of non-uniform mask coefficients, one can again use a simplifying assumption: all coefficients equal to one over the mask area. Then in essence, the filter performs local averaging in a rectangular neighborhood around the center of the mask. For example, for the simple one dimensional case with window size = 3:

For I=1…N
   pnew(I) = (1/3) * (pold(I-1) + pold(I) + pold(I+1))
Next I

Or for two dimensions and a 3x3 window:

For I1=1…N1
   For I2=1…N2
      pnew(I1,I2) =  (1/9) * (pold(I1-1,I2-1) + pold(I1-1,I2) + pold(I1-1,I2+1) +
       + pold(I1-1,I2) + pold(I1,I2) + pold(I1+1,I2) +
       + pold(I1+1,I2-1) + pold(I1+1,I2) + pold(I1+1,I2+1))
   Next I2
Next I1

In order to control the filter averaging effect, one could add a lambda coefficient belonging to [0…1]. Then, the diffusion function looks like:

For I=1…N
   pnew(I) = (1-lambda) * (1/3) * (pold(I-1) + pold(I) + pold(I+1)) +
            + lambda * pold(I)
Next I

In essence, this increases the contribution of the central bin, and decreases the averaging effect. Of course, in the above example we have not considered the treatment of negative indexes. One should also always take care of boundary conditions; one simple hack solution is to change the weighing when the mask is near the boundary, for example:

pnew(1) = (1/2)*pold(1) + pold(2))
while
pnew(2) = (1/3)*pold(1) + pold(2) + pold(3))

In our implementation, a variable-size rectangular window is used, and the lambda coefficient is hand-tuned for reasonable diffusion speeds (higher for position, less for color, less for size). Of course lambda as well as the window size or shape (or even each separate coefficient) could be empirically tuned, given data on the statistics of temporal change of property values (position, color, size etc.)

Nikolaos Mavridis, Cambridge, MA, May 2005