X

This routine provides solutions of active inference (minimisation of variational free energy) using a generative model based upon a Markov decision process (or hidden Markov model, in the absence of action). The model and inference scheme is formulated in discrete space and time. This means that the generative model (and process) are finite state machines or hidden Markov models whose dynamics are given by transition probabilities among states and the likelihood corresponds to a particular outcome conditioned upon hidden states.

When supplied with outcomes, in terms of their likelihood (O) in the absence of any policy specification, this scheme will use variational message passing to optimise expectations about latent or hidden states (and likelihood (A) and prior (B) probabilities). In other words, it will invert a hidden Markov model. When called with policies, it will generate outcomes that are used to infer optimal policies for active inference.

XX

This routine provides solutions of active inference (minimisation of variational free energy) using a generative model based upon a Markov decision process. The model and inference scheme is formulated in discrete space and time. This means that the generative model (and process) are hidden Markov models whose dynamics are given by transition probabilities among states and the likelihood corresponds to a particular outcome conditioned upon hidden states.

XXX

This routine provides solutions of active inference (minimisation of variational free energy) using a generative model based upon a Markov decision process. The model and inference scheme is formulated in discrete space and time. This means that the generative model (and process) are hidden Markov models whose dynamics are given by transition probabilities among states and the likelihood corresponds to a particular outcome conditioned upon hidden states.

X

This implementation equips agents with the prior beliefs that they wil maximise expected free energy: expected free energy is the free energy of future outcomes under the posterior predictive distribution. This can be interpreted in several ways - most intuitively as minimising the KL divergence between predicted and preferred outcomes (specified as prior beliefs) - while simultaneously minimising ambiguity.

XX

This implementation equips agents with the prior beliefs that they will maximise expected free energy. Variational free energy can be interpreted in several ways - most intuitively as minimising the KL divergence between predicted and preferred outcomes (specified as prior beliefs) - while simultaneously minimising ambiguity.

XXX

This implementation equips agents with the prior beliefs that they will maximise expected free energy. Expected free energy can be interpreted in several ways - most intuitively as minimising the KL divergence between predicted and preferred outcomes (specified as prior beliefs) -i.e., risk while simultaneously minimising ambiguity. Alternatively, this can be rearranged into expected information gain and expected value, where value is the log of prior preferences (overstates or outcomes).

X

This particular scheme is designed for any allowable policies or control sequences specified in MDP.V. Constraints on allowable policies can limit the numerics or combinatorics considerably. Further, the outcome space and hidden states can be defined in terms of factors; corresponding to sensory modalities and (functionally) segregated representations, respectively. This means, for each factor or subset of hidden states there are corresponding control states that determine the transition probabilities.

XX

This particular scheme is designed for any allowable policies or control variables specified in MDP.U. Constraints on allowable policies can limit the numerics or combinatorics considerably. Further, the outcome space and hidden states can be defined in terms of factors; corresponding to sensory modalities and (functionally) segregated representations, respectively. This means, for each factor or subset of hidden states there are corresponding control states that determine the transition probabilities. in this implementation, hidden factors are combined using a Kronecker intensive product to enable exact Bayesian inference using belief propagation (the Kronecker tensor form ensures that conditional dependencies among hidden factors are evaluated).

XXX

This implementation generalises previous MDP based formulations of active inference by equipping each factor of latent states with a number of paths; some of which may be controllable and others not. Controllable factors are now specified with indicator variables in the vector MDP.U. Furthermore, because the scheme uses sophisticated inference (i.e., a recursive tree search accumulating path integral is of expected free energy) a policy reduces to a particular combination of controllable paths or dynamics over factors. In consequence, posterior beliefs cover latent states and paths; with their associated variational free energies. Furthermore, it is now necessary to specify the initial states and the initial paths using D and E respectively. In other words, he now plays the role of a prior over the path of each factor that can only be changed if it is controllable (it no longer corresponds to a prior over policies).

X

This specification simplifies the generative model, allowing a fairly exhaustive model of potential outcomes.

In brief, the agent encodes beliefs about hidden states in the past (and in the future) conditioned on each policy. The conditional expectations determine the (path integral) of free energy that then determines the prior over policies. This prior is used to create a predictive distribution over outcomes, which specifies the next action.

XX

In this belief propagation scheme, the next action is evaluated in terms of the free energy expected under all subsequent actions until some time horizon (specified by MDP.T). This expected free energy is accumulated along all allowable paths or policies (see the subroutine spm_forward); effectively, performing a deep tree search over future sequences of actions. Because actions are conditionally independent of previous actions, it is only necessary to update posterior beliefs over hidden states at the current time point (using a Bayesian belief updating) and then use the prior over actions (based upon expected free energy) to select the next action. Previous actions are realised variables and are used when evaluating the posterior beliefs over current states.

In brief, the agent encodes beliefs about hidden states in the past conditioned on realised outcomes and actions. The resulting conditional expectations determine the (path integral) of free energy that then determines an empirical prior over the next action, from which the next realised action sampled

XXX

In this belief propagation scheme, the next action is evaluated in terms of the free energy expected under all subsequent actions until some time horizon (specified by MDP.T). This expected free energy is accumulated along all allowable paths or policies (see the subroutine spm_forward); effectively, performing a deep tree search over future sequences of actions. Because actions are conditionally independent of previous actions, it is only necessary to update posterior beliefs over hidden states at the current time point (using a Bayesian belief updating) and then use the prior over actions (based upon expected free energy) to select the next action. Previous actions are inferred under the posterior beliefs over current states; i.e., inferred state transitions.

In brief, the agent encodes beliefs about hidden states in the past conditioned on realised outcomes and actions. The resulting conditional expectations determine the (path integral) of free energy that then determines an empirical prior over the next action, from which the next realised action sampled

X

In addition to state estimation and policy selection, the scheme also updates model parameters; including the state transition matrices, mapping to outcomes and the initial state. This is useful for learning the context. Likelihood and prior probabilities can be specified in terms of concentration parameters (of a Dirichlet distribution (a,b,c,..). If the corresponding (A,B,C,..) are supplied, they will be used to generate outcomes; unless called without policies (in hidden Markov model mode). In this case, the (A,B,C,..) are treated as posterior estimates.

If supplied with a structure array, this routine will automatically step through the implicit sequence of epochs (implicit in the number of columns of the array). If the array has multiple rows, each row will be treated as a separate model or agent. This enables agents to communicate through acting upon a common set of hidden factors, or indeed sharing the same outcomes.

XX

In addition to state estimation and policy selection, the scheme also updates model parameters; including the state transition matrices, mapping to outcomes and the initial state. This is useful for learning the context. Likelihood and prior probabilities can be specified in terms of concentration parameters (of a Dirichlet distribution (a,b,c,..). If the corresponding (A,B,C,..) are supplied, they will be used to generate outcomes.

If supplied with a structure array, this routine will automatically step through the implicit sequence of epochs (implicit in the number of columns of the array). If the array has multiple rows, each row will be treated as a separate model or agent. This enables agents to communicate through acting upon a common set of hidden factors, or indeed sharing the same outcomes.

XXX

In addition to state and path estimation (and policy selection), the scheme also updates model parameters; including the state transition matrices, mapping to outcomes and the initial state. This is useful for learning the context. Likelihood and prior probabilities can be specified in terms of concentration parameters (of a Dirichlet distribution (a,b,c,...). If the corresponding (A,B,C,...) are supplied, they will be used to generate outcomes.

If supplied with a structure array, this routine will automatically step through the implicit sequence of epochs (implicit in the number of columns of the array). If the array has multiple rows, each row will be treated as a separate model or agent. This enables agents to communicate through acting upon a common set of hidden factors, or indeed sharing the same outcomes.

XXX

This scheme allows for differences in the functional form of priors – specified in terms of probability transition tensors – between the generating process and generative model. The generative model is, by default, specified in terms of Dirichlet parameters, while the generative process is specified in terms of expected (likelihood and prior transition) probabilities: b and B, respectively. If the number or dimensionality of b and B do not correspond, then select OPTIONS.A = 1. This will automatically evaluate the most likely policy (combination of controllable paths) to reproduce the predicted outcomes (i.e. that which minimises variational free energy or maximises accuracy); as opposed to using the path selected by the model.

scheme is designed for any allowable policies or control variables specified in MDP.U. Constraints on allowable policies can limit the numerics or combinatorics considerably. Further, the outcome space and hidden states can be defined in terms of factors; corresponding to sensory modalities and (functionally) segregated representations, respectively. This means, for each factor or subset of hidden states there are corresponding control states that determine the transition probabilities. in this implementation, hidden factors are combined using a Kronecker intensive product to enable exact Bayesian inference using belief propagation (the Kronecker tensor form ensures that conditional dependencies among hidden factors are evaluated).

X

See also: spm_MDP, which uses multiple future states and a mean field approximation for control states - but allows for different actions at all times (as in control problems).

See also: spm_MDP_game_KL, which uses a very similar formulation but just maximises the KL divergence between the posterior predictive distribution over hidden states and those specified by preferences or prior beliefs.

Karl Friston

XX

See also: spm_MDP, which uses multiple future states and a mean field approximation for control states - but allows for different actions at all times (as in control problems).

See also: spm_MDP_VB_X, which is the corresponding variational message passing scheme for fixed policies; i.e., ordered sequences of actions that are specified a priori.

Karl Friston

XXX

See also: spm_MDP, which uses multiple future states and a mean field approximation for control states - but allows for different actions at all times (as in control problems).

See also: spm_MDP_VB_X, which is the corresponding variational message passing scheme for fixed policies; i.e., ordered sequences of actions that are specified a priori.

See also: spm_MDP_VB_XX, which is the corresponding variational message passing scheme for sophisticated policy searches under the assumption that the generative process and model have the same structure