Approximating Labelled Markov Processes

Prakash Panangaden[1]

[1] School of Computer Science
McGill University

August 2008, Workshop on Approximate Behavioural Equivalences

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Outline

1. Introduction
2. Labelled Markov processes
3. Probabilistic bisimulation
4. Approximation
5. Better approximants
6. Approximating by averaging
7. Conclusion

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathrm{L}, \forall a \in \mathrm{L}\ T_a : S \times S \longrightarrow [0, 1])$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathrm{L}, \forall a \in \mathrm{L}\ T_a : S \times S \longrightarrow [0,1])$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathrm{L}, \forall a \in \mathrm{L}\ T_a : S \times S \longrightarrow [0,1])$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

Strong Probabilistic Bisimulation: Larsen-Skou

- Let $\mathcal{S} = (S, \mathrm{L}, T_a)$ be a PTS. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-equivalence class, A, $T_a(s, A) = T_a(s', A)$.
- The notation $T_a(s, A)$ means "the probability of starting from s and jumping to a state in the set A."
- Two states are bisimilar if there is some bisimulation relation R relating them.

Strong Probabilistic Bisimulation: Larsen-Skou

- Let $\mathcal{S} = (S, \mathrm{L}, T_a)$ be a PTS. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-equivalence class, A, $T_a(s, A) = T_a(s', A)$.
- The notation $T_a(s, A)$ means “the probability of starting from s and jumping to a state in the set A.”
- Two states are bisimilar if there is some bisimulation relation R relating them.

Strong Probabilistic Bisimulation: Larsen-Skou

- Let $\mathcal{S} = (S, \mathrm{L}, T_a)$ be a PTS. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-equivalence class, A, $T_a(s, A) = T_a(s', A)$.
- The notation $T_a(s, A)$ means “the probability of starting from s and jumping to a state in the set A.”
- Two states are bisimilar if there is some bisimulation relation R relating them.

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- **In general, the state space of a labelled Markov process may be a *continuum*.**

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- **In general, the state space of a labelled Markov process may be a *continuum*.**

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- In general, the state space of a labelled Markov process may be a *continuum*.

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- **In general, the state space of a labelled Markov process may be a *continuum*.**

Stochastic Kernels

- A *stochastic kernel* (Markov kernel) is a function

  $$\tau : S \times \Sigma \longrightarrow [0, 1]$$

  with (a) $\tau(s, \cdot) : \Sigma \longrightarrow [0, 1]$ a (sub)probability measure and (b) $\tau(\cdot, A) : X \longrightarrow [0, 1]$ a measurable function.
- Though apparantly asymmetric, these are the stochastic analogues of binary relations
- and the uncountable generalization of a matrix.

Stochastic Kernels

- A *stochastic kernel* (Markov kernel) is a function

  $$\tau : S \times \Sigma \longrightarrow [0,1]$$

  with (a) $\tau(s,\cdot) : \Sigma \longrightarrow [0,1]$ a (sub)probability measure and (b) $\tau(\cdot, A) : X \longrightarrow [0,1]$ a measurable function.
- Though apparantly asymmetric, these are the stochastic analogues of binary relations
- and the uncountable generalization of a matrix.

Stochastic Kernels

- A *stochastic kernel* (Markov kernel) is a function

  $$\tau : S \times \Sigma \longrightarrow [0, 1]$$

  with (a) $\tau(s, \cdot) : \Sigma \longrightarrow [0, 1]$ a (sub)probability measure and (b) $\tau(\cdot, A) : X \longrightarrow [0, 1]$ a measurable function.
- Though apparantly asymmetric, these are the stochastic analogues of binary relations
- and the uncountable generalization of a matrix.

Formal Definition of LMPs

Fix a set of labels L. An LMP is a tuple $(S, \Sigma, \forall \alpha \in \mathrm{L}.\tau_\alpha)$ where $\tau_\alpha : S \times \Sigma \longrightarrow [0, 1]$ is a *stochastic kernel*.

Larsen-Skou Bisimulation

- Let $\mathcal{S} = (S, \Sigma, \tau)$ be a labelled Markov process. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $A \in \Sigma$, $\tau_a(s, A) = \tau_a(s', A)$. Two states are bisimilar if they are related by a bisimulation relation.
- Can be extended to bisimulation between two different LMPs.

Larsen-Skou Bisimulation

- Let $\mathcal{S} = (S, \Sigma, \tau)$ be a labelled Markov process. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $A \in \Sigma$, $\tau_a(s, A) = \tau_a(s', A)$. Two states are bisimilar if they are related by a bisimulation relation.
- Can be extended to bisimulation between two different LMPs.

Logical Characterization

- $$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$
- We say $s \models \langle a \rangle_q \phi$ iff
  $$\exists A \in \Sigma . (\forall s' \in A . s' \models \phi) \wedge (\tau_a(s, A) > q).$$
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

Logical Characterization

- $$\mathcal{L} ::== \mathrm{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$
- We say $s \models \langle a \rangle_q \phi$ iff
  $$\exists A \in \Sigma.(\forall s' \in A.s' \models \phi) \wedge (\tau_a(s, A) > q).$$
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

Logical Characterization

- $$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$
- We say $s \models \langle a \rangle_q \phi$ iff
$$\exists A \in \Sigma.(\forall s' \in A.s' \models \phi) \wedge (\tau_a(s, A) > q).$$
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

The Approximation Construction: Take 1

- Given a labelled Markov process $\mathcal{S} = (S, i, \Sigma, \tau)$, an integer n and a *rational* number $\epsilon > 0$, we define $\mathcal{S}(n, \epsilon)$ to be an n-step unfolding approximation of $\mathcal{S}$.
- Its state-space is divided into $n + 1$ levels which are numbered $0, 1, \ldots, n$.
- A state is a pair (X, l) where $X \in \Sigma$ and $l \in \{0, 1, \ldots, n\}$.
- *At each level*, the sets that define states form a partition of S. The initial state of $\mathcal{S}(n, \epsilon)$ is at level n and transitions only occur between a state of one level to a state of one lower level.

The Approximation Construction: Take 1

- Given a labelled Markov process $\mathcal{S} = (S, i, \Sigma, \tau)$, an integer n and a *rational* number $\epsilon > 0$, we define $\mathcal{S}(n, \epsilon)$ to be an n-step unfolding approximation of $\mathcal{S}$.
- Its state-space is divided into $n + 1$ levels which are numbered $0, 1, \ldots, n$.
- A state is a pair (X, l) where $X \in \Sigma$ and $l \in \{0, 1, \ldots, n\}$.
- *At each level*, the sets that define states form a partition of S. The initial state of $\mathcal{S}(n, \epsilon)$ is at level n and transitions only occur between a state of one level to a state of one lower level.

The Approximation Construction: Take 1

- Given a labelled Markov process $\mathcal{S} = (S, i, \Sigma, \tau)$, an integer n and a *rational* number $\epsilon > 0$, we define $\mathcal{S}(n, \epsilon)$ to be an n-step unfolding approximation of $\mathcal{S}$.
- Its state-space is divided into $n + 1$ levels which are numbered $0, 1, \ldots, n$.
- A state is a pair (X, l) where $X \in \Sigma$ and $l \in \{0, 1, \ldots, n\}$.
- *At each level*, the sets that define states form a partition of S. The initial state of $\mathcal{S}(n, \epsilon)$ is at level n and transitions only occur between a state of one level to a state of one lower level.

The Approximation Construction: Take 1

- Given a labelled Markov process $\mathcal{S} = (S, i, \Sigma, \tau)$, an integer n and a *rational* number $\epsilon > 0$, we define $\mathcal{S}(n, \epsilon)$ to be an n-step unfolding approximation of $\mathcal{S}$.
- Its state-space is divided into $n + 1$ levels which are numbered $0, 1, \ldots, n$.
- A state is a pair (X, l) where $X \in \Sigma$ and $l \in \{0, 1, \ldots, n\}$.
- *At each level*, the sets that define states form a partition of S. The initial state of $\mathcal{S}(n, \epsilon)$ is at level n and transitions only occur between a state of one level to a state of one lower level.

The states

- States are defined by induction on their level. Level 0 has one state $(S, 0)$.
- Given the sets from level l, we define states of level $l+1$ as follows. Suppose that there are m states at level l, we partition the interval $[0, 1]$ into intervals of size ϵ/m. Let $(B_j)_{j\in I}$ stand for this partition; i.e. for

$$\{\{0\},(0,\epsilon/m],(\epsilon/m,2\epsilon/m],\ldots\}.$$

The states

- States are defined by induction on their level. Level 0 has one state $(S, 0)$.
- Given the sets from level l, we define states of level $l + 1$ as follows. Suppose that there are m states at level l, we partition the interval $[0, 1]$ into intervals of size ϵ/m. Let $(B_j)_{j \in I}$ stand for this partition; i.e. for

$$\{\{0\}, (0, \epsilon/m], (\epsilon/m, 2\epsilon/m], \ldots\}.$$

The states 2

For each (C, l) at level l and each B_j and each label we form the sets

$$C_j^{(a)} := \tau_a(\cdot, C)^{-1}(B_j).$$

Now we form the partition generated by the $C_j^{(a)}$; these are the sets at level $l + 1$.

Transitions

Transitions can happen from a state of level $l+1$ to a state of level l, and the transition probability function is given by

$$\rho_a((X,k),(B,l)) = \begin{cases} \inf_{t\in X} \tau_a(t,B)) & \text{if } k = l+1, \\ 0 & \text{otherwise.} \end{cases}$$

The initial state p_0 of $\mathcal{S}(n,\epsilon)$ is the unique state (X,n) such that X contains i, the initial state of $\mathcal{S}$.

What is happening?

- Bisimulation is the greatest fixed point of a suitable functional on relations.
- The fixed point is attained at ω.
- The approximations are really approximating the bisimulation relation by giving the equivalence classes.

What is happening?

- Bisimulation is the greatest fixed point of a suitable functional on relations.
- The fixed point is attained at ω.
- The approximations are really approximating the bisimulation relation by giving the equivalence classes.

What is happening?

- Bisimulation is the greatest fixed point of a suitable functional on relations.
- The fixed point is attained at ω.
- The approximations are really approximating the bisimulation relation by giving the equivalence classes.

Approximants are below

Every labelled Markov process $\mathcal{S}$ simulates all its approximations of the form $\mathcal{S}(n, \epsilon)$. More precisely, every state (X, l) of $\mathcal{S}(n, \epsilon)$ $(l \leq n)$ is simulated in $\mathcal{S}$ by every $s \in X$.

Approximants eventually get there

- If a state $s \in S$ satisfies a formula $\phi \in \mathcal{L}_\vee$, then there is some approximation $\mathcal{S}(n, \epsilon)$ such that $(X_s, n) \models \phi$.
- We also have a metric convergence theorem and a domain-theoretic convergence theorem.

Approximants eventually get there

- If a state $s \in S$ satisfies a formula $\phi \in \mathcal{L}_{\vee}$, then there is some approximation $\mathcal{S}(n, \epsilon)$ such that $(X_s, n) \models \phi$.
- We also have a metric convergence theorem and a domain-theoretic convergence theorem.

How can we do better?

- We would like to “orient” the approximation process so that they are tailored to some formulas of interest.
- We do not get the original process in the limit, but a bisimulation equivalent of it. Sometimes this is “spectacularly” not what we want.
- We can fix both the problems above but then we end up with the situation that the approximants are not LMPs. [DD, LICS03]
- We can fix this too with a new approach to approximation based on conditional expectations. [DDP, CONCUR03]

How can we do better?

- We would like to “orient” the approximation process so that they are tailored to some formulas of interest.
- We do not get the original process in the limit, but a bisimulation equivalent of it. Sometimes this is “spectacularly” not what we want.
- We can fix both the problems above but then we end up with the situation that the approximants are not LMPs. [DD, LICS03]
- We can fix this too with a new approach to approximation based on conditional expectations. [DDP, CONCUR03]

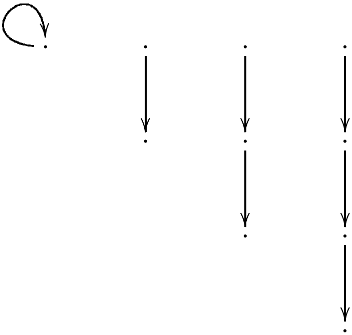
How can we do better?

- We would like to “orient” the approximation process so that they are tailored to some formulas of interest.
- We do not get the original process in the limit, but a bisimulation equivalent of it. Sometimes this is “spectacularly” not what we want.
- We can fix both the problems above but then we end up with the situation that the approximants are not LMPs. [DD, LICS03]
- We can fix this too with a new approach to approximation based on conditional expectations. [DDP, CONCUR03]

How can we do better?

- We would like to "orient" the approximation process so that they are tailored to some formulas of interest.
- We do not get the original process in the limit, but a bisimulation equivalent of it. Sometimes this is "spectacularly" not what we want.
- We can fix both the problems above but then we end up with the situation that the approximants are not LMPs. [DD, LICS03]
- We can fix this too with a new approach to approximation based on conditional expectations. [DDP, CONCUR03]

Unwinding a Loop



In the limit we get the infinite chain. This is indeed bisimilar to the loop but this seems a dumb way to approximate.

Allow Cycles

- The fix is simple: allow cycles.
- The states are organized into levels as before; but the depth defers to how much we observe the system rather than to how deep its transition graph is.
- Transitions occur between a state at a given level and states *at the same level* or states at one level below.

Allow Cycles

- The fix is simple: allow cycles.
- The states are organized into levels as before; but the depth defers to how much we observe the system rather than to how deep its transition graph is.
- Transitions occur between a state at a given level and states *at the same level* or states at one level below.

Allow Cycles

- The fix is simple: allow cycles.
- The states are organized into levels as before; but the depth defers to how much we observe the system rather than to how deep its transition graph is.
- Transitions occur between a state at a given level and states *at the same level* or states at one level below.

Transition Probabilities

Let $(X, l+1)$ and $(Y, l+1)$ be states of level $l+1$ and (Z, l) a state of level l.
Then

$$\begin{aligned} \rho_a((X, l+1), (Y, l+1)) &= \inf_{x \in X} \tau_a(x, Y) \\ \rho_a((X, l+1), (Z, l)) &= \inf_{x \in X} \tau_a(x, Z) \\ &- \sum_{i=1}^{k} \rho_a((X, l+1), (Z_i, l+1)) \end{aligned}$$

where $\cup Z_i = Z$ and $(Z_i, l+1)$ is a state for every i.
Unspecified transitions are given the value 0.

The Idea

- The transition probabilities between states at the same level are made as large as possible while still staying below what the original system would say.
- This is not enough to guarantee accuracy since the partition is constructed to be matched with the transitions to the lower level. We add in the transition probabilities to the level l states to fix this.
- If X is a state at level l (i.e. a member of the level l partition of S) then every $x \in X$ has transition probabilities within ϵ/m of each other to any set of the partition at level $l + 1$.
- This allows one to prove all the approximation lemmas.

The Idea

- The transition probabilities between states at the same level are made as large as possible while still staying below what the original system would say.
- This is not enough to guarantee accuracy since the partition is constructed to be matched with the transitions to the lower level. We add in the transition probabilities to the level l states to fix this.
- If X is a state at level l (i.e. a member of the level l partition of S) then every $x \in X$ has transition probabilities within ϵ/m of each other to any set of the partition at level $l + 1$.
- This allows one to prove all the approximation lemmas.

The Idea

- The transition probabilities between states at the same level are made as large as possible while still staying below what the original system would say.
- This is not enough to guarantee accuracy since the partition is constructed to be matched with the transitions to the lower level. We add in the transition probabilities to the level l states to fix this.
- If X is a state at level l (i.e. a member of the level l partition of S) then every $x \in X$ has transition probabilities within ϵ/m of each other to any set of the partition at level $l + 1$.
- This allows one to prove all the approximation lemmas.

The Idea

- The transition probabilities between states at the same level are made as large as possible while still staying below what the original system would say.
- This is not enough to guarantee accuracy since the partition is constructed to be matched with the transitions to the lower level. We add in the transition probabilities to the level l states to fix this.
- If X is a state at level l (i.e. a member of the level l partition of S) then every $x \in X$ has transition probabilities within ϵ/m of each other to any set of the partition at level $l+1$.
- This allows one to prove all the approximation lemmas.

Approximations converge

- The previous results about the approximations being simulated by the process are still true.
- The convergence theorems are still true.
- The new result:
  For every finite-state process there exists a bisimilar approximation.

Approximations converge

- The previous results about the approximations being simulated by the process are still true.
- The convergence theorems are still true.
- The new result:
  For every finite-state process there exists a bisimilar approximation.

Approximations converge

- The previous results about the approximations being simulated by the process are still true.
- The convergence theorems are still true.
- The new result:
  For every finite-state process there exists a bisimilar approximation.

Accuracy

- The theorem about approximants eventually satisfying a formula is painful to prove. Why?
- By taking the infima we lose accuracy.
- We have additivity because we define the state to state transitions by infima and extend to sets of states by additivity. If we try to define transition probabilities to all *sets* states by infima directly, we are more accurate
- *but we lose additivity*.

Accuracy

- The theorem about approximants eventually satisfying a formula is painful to prove. Why?
- By taking the infima we lose accuracy.
- We have additivity because we define the state to state transitions by infima and extend to sets of states by additivity. If we try to define transition probabilities to all *sets* states by infima directly, we are more accurate
- *but we lose additivity*.

Accuracy

- The theorem about approximants eventually satisfying a formula is painful to prove. Why?
- By taking the infima we lose accuracy.
- We have additivity because we define the state to state transitions by infima and extend to sets of states by additivity. If we try to define transition probabilities to all *sets* states by infima directly, we are more accurate
- *but we lose additivity*.

Accuracy

- The theorem about approximants eventually satisfying a formula is painful to prove. Why?
- By taking the infima we lose accuracy.
- We have additivity because we define the state to state transitions by infima and extend to sets of states by additivity. If we try to define transition probabilities to all *sets* states by infima directly, we are more accurate
- *but we lose additivity*.

Guiding the Approximation

- To guide the approximation we can quotient by the equivalence induced by a subset of formulas.
- This should converge because of the logical characterization of bisimulation.
- We will have to deal with loss of additivity here too.
- One can live with *capacities* instead of measures.
- Details in LICS03 paper by Danos and Desharnais.

Guiding the Approximation

- To guide the approximation we can quotient by the equivalence induced by a subset of formulas.
- This should converge because of the logical characterization of bisimulation.
- We will have to deal with loss of additivity here too.
- One can live with *capacities* instead of measures.
- Details in LICS03 paper by Danos and Desharnais.

Guiding the Approximation

- To guide the approximation we can quotient by the equivalence induced by a subset of formulas.
- This should converge because of the logical characterization of bisimulation.
- We will have to deal with loss of additivity here too.
- One can live with *capacities* instead of measures.
- Details in LICS03 paper by Danos and Desharnais.

Guiding the Approximation

- To guide the approximation we can quotient by the equivalence induced by a subset of formulas.
- This should converge because of the logical characterization of bisimulation.
- We will have to deal with loss of additivity here too.
- One can live with *capacities* instead of measures.
- Details in LICS03 paper by Danos and Desharnais.

Guiding the Approximation

- To guide the approximation we can quotient by the equivalence induced by a subset of formulas.
- This should converge because of the logical characterization of bisimulation.
- We will have to deal with loss of additivity here too.
- One can live with *capacities* instead of measures.
- Details in LICS03 paper by Danos and Desharnais.

Averages

- Approximation has been based on the idea of “approximating from below” as we are inspired by domain theory and logic.
- In probability it is natural to work with averages. We should approximate by averaging over “cells.”
- How does this interact with the logic?
- *We need a given measure on the system*.

Averages

- Approximation has been based on the idea of “approximating from below” as we are inspired by domain theory and logic.
- In probability it is natural to work with averages. We should approximate by averaging over “cells.”
- How does this interact with the logic?
- *We need a given measure on the system*.

Averages

- Approximation has been based on the idea of "approximating from below" as we are inspired by domain theory and logic.
- In probability it is natural to work with averages. We should approximate by averaging over "cells."
- How does this interact with the logic?
- *We need a given measure on the system*.

Averages

- Approximation has been based on the idea of "approximating from below" as we are inspired by domain theory and logic.
- In probability it is natural to work with averages. We should approximate by averaging over "cells."
- How does this interact with the logic?
- *We need a given measure on the system*.

Conditional Expectations

- Let $X : S \longrightarrow \mathbb{R}$ be a random variable on a probability space (S, Σ, p). The *expectation value* of X is $\mathbb{E}_p(X) = \int Xdp$; a number.
- What if we know that the outcome is in a set $Q \in \Sigma$? Then we can revise the expectation to
$$\mathbb{E}_p(X|Q) = \int_Q Xdp.$$
- What if we know for every set $B \in \Lambda$, where Λ is a σ-sub-algebra of Σ whether the outcome is in B or not? We can recompute the expectation value for every set $B \in \Lambda$. But how do we present all these revised expectation values?

Conditional Expectations

- Let $X : S \longrightarrow \mathbb{R}$ be a random variable on a probability space (S, Σ, p). The *expectation value* of X is $\mathbb{E}_p(X) = \int X dp$; a number.
- What if we know that the outcome is in a set $Q \in \Sigma$? Then we can revise the expectation to

$$\mathbb{E}_p(X|Q) = \int_Q X dp.$$

- What if we know for every set $B \in \Lambda$, where Λ is a σ-sub-algebra of Σ whether the outcome is in B or not? We can recompute the expectation value for every set $B \in \Lambda$. But how do we present all these revised expectation values?

Conditional Expectations

- Let $X : S \longrightarrow \mathbb{R}$ be a random variable on a probability space (S, Σ, p). The *expectation value* of X is $\mathbb{E}_p(X) = \int Xdp$; a number.
- What if we know that the outcome is in a set $Q \in \Sigma$? Then we can revise the expectation to

$$\mathbb{E}_p(X|Q) = \int_Q Xdp.$$

- What if we know for every set $B \in \Lambda$, where Λ is a σ-sub-algebra of Σ whether the outcome is in B or not? We can recompute the expectation value for every set $B \in \Lambda$. But how do we present all these revised expectation values?

Conditional Expectations 2

- We can define a *new random variable* $\mathbb{E}_p(X|\Lambda) : S \longrightarrow \mathbb{R}$ called the *conditional expectation* of X given Λ which is Λ-measurable and satisfies

$$\forall B \in \Lambda \int_B X dP = \int_B \mathbb{E}_P(X|\Lambda) dP.$$

- How do we know this exists? Yet another fundamental result of Kolmogorov. The functional

$$\mathbb{E}_p(\cdot|\Lambda) : \mathcal{L}^1(S, \Sigma, p) \longrightarrow \mathcal{L}^1(S, \Lambda, p)$$

is linear, continuous and unique up to a set of p-measure 0.

Conditional Expectations 2

- We can define a *new random variable* $\mathbb{E}_p(X|\Lambda) : S \longrightarrow \mathbb{R}$ called the *conditional expectation* of X given Λ which is Λ-measurable and satisfies

$$\forall B \in \Lambda \int_B X dP = \int_B \mathbb{E}_P(X|\Lambda) dP.$$

- How do we know this exists? Yet another fundamental result of Kolmogorov. The functional

$$\mathbb{E}_p(\cdot|\Lambda) : \mathcal{L}^1(S, \Sigma, p) \longrightarrow \mathcal{L}^1(S, \Lambda, p)$$

is linear, continuous and unique up to a set of p-measure 0.

Independence

- Given a probability space (S, Σ, p) a random variable X is said to be *independent* of a sub-σ-algebra Λ if for every event A in the σ-algebra $\sigma(X)$ generated by X and $B \in \Lambda$ we have $p(A \cap B) = p(A)p(B)$.
- If X is independent of Λ then $\mathbb{E}_p(X|\Lambda) = \mathbb{E}_p(X)$; i.e. a constant. If X is completely dependent on Λ (i.e. it is Λ-measurable) then $\mathbb{E}_p(X|\Lambda) = X$.

Independence

- Given a probability space (S, Σ, p) a random variable X is said to be *independent* of a sub-σ-algebra Λ if for every event A in the σ-algebra $\sigma(X)$ generated by X and $B \in \Lambda$ we have $p(A \cap B) = p(A)p(B)$.
- If X is independent of Λ then $\mathbb{E}_p(X|\Lambda) = \mathbb{E}_p(X)$; i.e. a constant. If X is completely dependent on Λ (i.e. it is Λ-measurable) then $\mathbb{E}_p(X|\Lambda) = X$.

Equivalences and Expectations

- Suppose that we have an equivalence relation R on (S, Σ, p) and that Λ is generated by the equivalence classes. Let A be an R-equivalence class and $s \in A$.
- Then $Y = \mathbb{E}_p(X|\Lambda)$ has to be constant on R-equivalence classes or else Y would not be Λ-measurable.
- $$Y(s)p(A) = \int_A Y dp = \int_A X dp = \mathbb{E}_p 1_A X.$$

  Thus we get

  $$Y(s) = \frac{1}{p(A)} \cdot \mathbb{E}_p 1_A X$$

Equivalences and Expectations

- Suppose that we have an equivalence relation R on (S, Σ, p) and that Λ is generated by the equivalence classes. Let A be an R-equivalence class and $s \in A$.
- Then $Y = \mathbb{E}_p(X|\Lambda)$ has to be constant on R-equivalence classes or else Y would not be Λ-measurable.
- $$Y(s)p(A) = \int_A Ydp = \int_A Xdp = \mathbb{E}_p 1_A X.$$

  Thus we get

  $$Y(s) = \frac{1}{p(A)} \cdot \mathbb{E}_p 1_A X$$

Equivalences and Expectations

- Suppose that we have an equivalence relation R on (S, Σ, p) and that Λ is generated by the equivalence classes. Let A be an R-equivalence class and $s \in A$.
- Then $Y = \mathbb{E}_p(X|\Lambda)$ has to be constant on R-equivalence classes or else Y would not be Λ-measurable.
- $$Y(s)p(A) = \int_A Ydp = \int_A Xdp = \mathbb{E}_p 1_A X.$$

  Thus we get

  $$Y(s) = \frac{1}{p(A)} \cdot \mathbb{E}_p 1_A X$$

Averaging

In other words: the values of Y are obtained by averaging over the R-equivalence classes.

Uniqueness Lost

- Computatonally we do not like phrases like “defined upto a set of measure 0” or “almost surely” or “a version of”...
- (S, Σ, p), $\Lambda \subseteq \Sigma$ a sub-σ-algebra and define an equivalence relation $s \sim t$ if and only if $\forall Q \in \Lambda . s \in Q \Leftrightarrow t \in Q$.
- We say p is *granular* over Λ if and only if for all $s \in S . p([s]) \neq 0$.

Uniqueness Lost

- Computatonally we do not like phrases like “defined upto a set of measure 0” or “almost surely” or “a version of”...
- (S, Σ, p), $\Lambda \subseteq \Sigma$ a sub-σ-algebra and define an equivalence relation $s \sim t$ if and only if $\forall Q \in \Lambda . s \in Q \Leftrightarrow t \in Q$.
- We say p is *granular* over Λ if and only if for all $s \in S . p([s]) \neq 0$.

Uniqueness Lost

- Computatonally we do not like phrases like “defined upto a set of measure 0” or “almost surely” or “a version of”...
- (S, Σ, p), $\Lambda \subseteq \Sigma$ a sub-σ-algebra and define an equivalence relation $s \sim t$ if and only if $\forall Q \in \Lambda . s \in Q \Leftrightarrow t \in Q$.
- We say p is *granular* over Λ if and only if for all $s \in S . p([s]) \neq 0$.

Granularity Examples

- Consider the probability triple $([0,1)^2, \mathcal{B}_2, \lambda_2)$. Take Λ to be $\mathcal{B} \times [0,1]$. Then $[s] = \{s\} \times [0,1]$ and $\lambda_2([s]) = 0$. The Λ-measurable sets are too “thin” to be granular.
- For the same space take Λ to be generated by the squares $[k/n,(k+1)/n) \times [h/n,(h+1)/n)$ for $k,h \in \{0,\ldots,n-1\}$. Then $\lambda_2([s]) = 1/n^2$; thus here λ_2 is granular over Λ.

Granularity Examples

- Consider the probability triple $([0,1)^2, \mathcal{B}_2, \lambda_2)$. Take Λ to be $\mathcal{B} \times [0,1]$. Then $[s] = \{s\} \times [0,1]$ and $\lambda_2([s]) = 0$. The Λ-measurable sets are too "thin" to be granular.
- For the same space take Λ to be generated by the squares $[k/n,(k+1)/n) \times [h/n,(h+1)/n)$ for $k,h \in \{0,\ldots,n-1\}$. Then $\lambda_2([s]) = 1/n^2$; thus here λ_2 is granular over Λ.

Uniqueness Regained

Suppose that (S, Σ, p) is a probability triple and $\Lambda \subseteq \Sigma$ with p granular over Λ. It is easy to prove that if $X, Y : S \longrightarrow \mathbb{R}$ are both Λ-measurable then

$$X = Y a.s. \Rightarrow X = Y.$$

Thus we get rid of “versions.”

Approximation

- Given an LMP (S, Σ, τ) and a *finite* set $\mathcal{F}$ of formulas of $\mathcal{L}^*$ we consider the σ-algebra $\sigma(\mathcal{F})$ generated by the sets $\{[\![\phi]\!] | \phi \in \mathcal{F}\}$.
- p is granular over $\sigma(\mathcal{F})$. Taking Λ to be $\sigma(\mathcal{F})$ we can quotient to get $\mathcal{S}_{\mathcal{F}}$ a *finite-state* approximant of size at most $2^{|\mathcal{F}|}$.

Approximation

- Given an LMP (S, Σ, τ) and a *finite* set $\mathcal{F}$ of formulas of $\mathcal{L}^*$ we consider the σ-algebra $\sigma(\mathcal{F})$ generated by the sets $\{[\![\phi]\!] | \phi \in \mathcal{F}\}$.
- p is granular over $\sigma(\mathcal{F})$. Taking Λ to be $\sigma(\mathcal{F})$ we can quotient to get $\mathcal{S}_{\mathcal{F}}$ a *finite-state* approximant of size at most $2^{|\mathcal{F}|}$.

Properties of the approximation

- If $\mathcal{F}$ is a finite *subformula-closed* set of formulas of $\mathcal{L}^*$ then $\mathcal{S}_{\mathcal{F}}$ and $\mathcal{S}$ agree on all formulas of $\mathcal{F}$.
- If $\mathcal{F}_i$ is an increasing family of subformula-closed sets of formulas of $\mathcal{L}^*$ converging to all of $\mathcal{L}^*$ then $\mathcal{S}_i$ converges in our metrics to $\mathcal{S}$.

Properties of the approximation

- If $\mathcal{F}$ is a finite *subformula-closed* set of formulas of $\mathcal{L}^*$ then $\mathcal{S}_{\mathcal{F}}$ and $\mathcal{S}$ agree on all formulas of $\mathcal{F}$.
- If $\mathcal{F}_i$ is an increasing family of subformula-closed sets of formulas of $\mathcal{L}^*$ converging to all of $\mathcal{L}^*$ then $\mathcal{S}_i$ converges in our metrics to $\mathcal{S}$.

Recent work

- In QEST 2005 Bouchard-Cote et al. showed how to “estimate” the approximants by sampling. Special techniques to avoid small blocks in the partition.
- Norm Ferns et al. have developed techniques based on infinite-dimensional LPs to approximate MDPs (with rewards).

Recent work

- In QEST 2005 Bouchard-Cote et al. showed how to “estimate” the approximants by sampling. Special techniques to avoid small blocks in the partition.
- Norm Ferns et al. have developed techniques based on infinite-dimensional LPs to approximate MDPs (with rewards).

Conclusions

- We have 3 approaches to approximation
- The first is concrete but cannot be tailored to specific formulas
- The second can be tailored to specific formulas of interest but takes one into the realm of pre-LMPs
- The third - based on averaging - can also be tailored and keeps us in the realm of LMPs.

Conclusions

- We have 3 approaches to approximation
- The first is concrete but cannot be tailored to specific formulas
- The second can be tailored to specific formulas of interest but takes one into the realm of pre-LMPs
- The third - based on averaging - can also be tailored and keeps us in the realm of LMPs.

Conclusions

- We have 3 approaches to approximation
- The first is concrete but cannot be tailored to specific formulas
- The second can be tailored to specific formulas of interest but takes one into the realm of pre-LMPs
- The third - based on averaging - can also be tailored and keeps us in the realm of LMPs.

Conclusions

- We have 3 approaches to approximation
- The first is concrete but cannot be tailored to specific formulas
- The second can be tailored to specific formulas of interest but takes one into the realm of pre-LMPs
- The third - based on averaging - can also be tailored and keeps us in the realm of LMPs.