Dialogue management: Non-parametric approaches to policy optimisation

Milica Gašić

Dialogue Systems Group, Cambridge University Engineering Department

February 11, 2016

Problems in applying RL to dialogue

Gaussian process model for Q-function

GP-Sarsa algorithm

Applying reinforcement learning to dialogue

Problems in solving dialogue as an RL task

1. Size of the optimisation problem
   - Belief state is large and continuous
   - Set of system actions also large
2. Knowledge of the environment, in this case the user
   - We do not have transition probabilities
   - Where do rewards come from?
3. RL algorithms take a long time to converge

Solutions

- Learn in reduced summary space (1)
- Learn in interaction with a simulated user (2&3)

Are these good solutions?

Theory: Reinforcement learning

Policy: deterministic $\pi : \mathcal{B} \rightarrow \mathcal{A}$ or stochastic $\pi : \mathcal{B} \times \mathcal{A} \rightarrow [0, 1]$

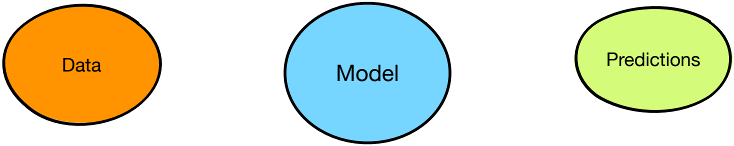
Return: $R_t^{\pi} = \sum_{k=0}^{T-l} \gamma^k r_{t+k}$

Q-function: What is the value of taking action a in belief state $\mathbf{b}$ under a policy π?

$$Q^{\pi}(\mathbf{b}, a) = E_{\pi}\left\{\sum_{k=0}^{T-t} \gamma^k r_{t+k} | b_t = \mathbf{b}, a_t = a\right\}$$

Can we find optimal Q-function with fewer data points so that we can learn from real users?

Non-parametric model for Q-function



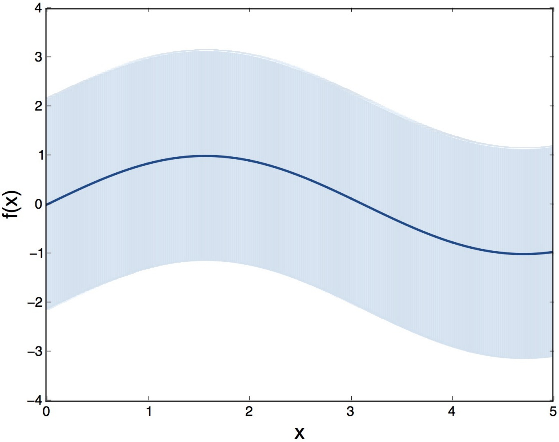
- Belief states (from belief tracker)
- Reward – a measure of dialogue quality

- Gaussian process model of the Q-function

- Optimal Q-function

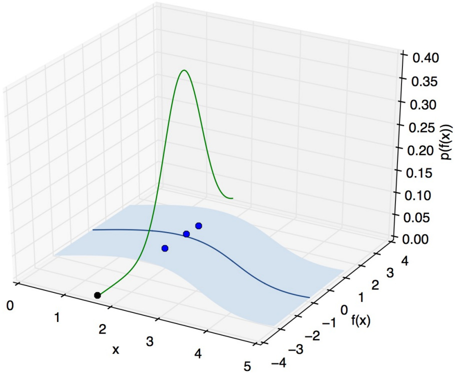
Theory: Gaussian processes prior

$$f(x) \sim \mathcal{GP}(m(x), k(x, x))$$



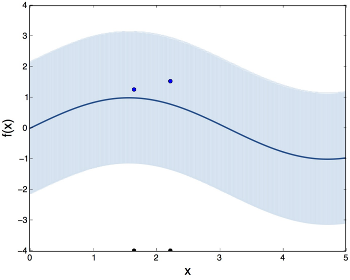
Theory: Gaussian processes kernel

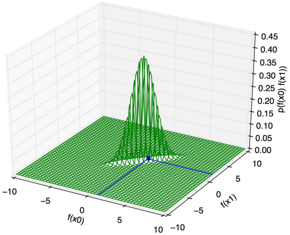
$$f(x_0) \sim \mathcal{N}(m(x_0), k(x_0, x_0))$$



Theory: Gaussian processes kernel

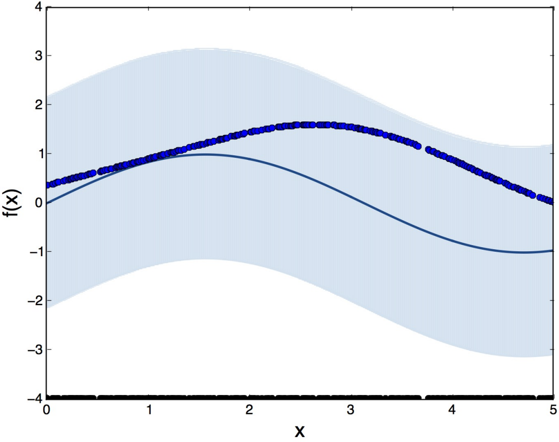
$$\left[\begin{array}{c} f(x_0) \\ f(x_1) \end{array}\right] \sim \mathcal{N}\left(\left[\begin{array}{c} m(x_0) \\ m(x_1) \end{array}\right], \left[\begin{array}{c} k(x_0, x_0), k(x_0, x_1) \\ k(x_1, x_0), k(x_1, x_1) \end{array}\right]\right)$$





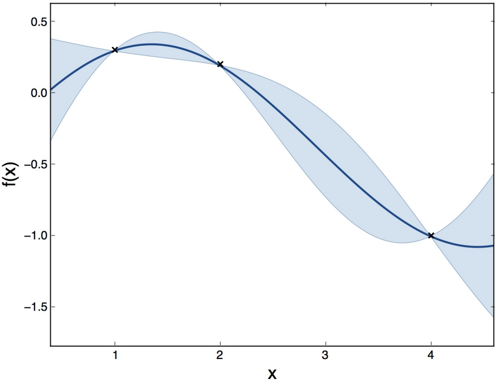
Theory: Gaussian processes kernel

Any number of function values is Gaussian distributed.



Theory: Gaussian processes posterior

- Observations **y** in **x** and $f(x)$ are jointly Gaussian distributed
- Conditional is then also a Gaussian process
  $f(x)|\mathbf{x}, \mathbf{y} \sim \mathcal{GP}(\bar{f}(x), cov(x, x))$

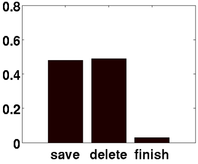


Toy dialogue problem

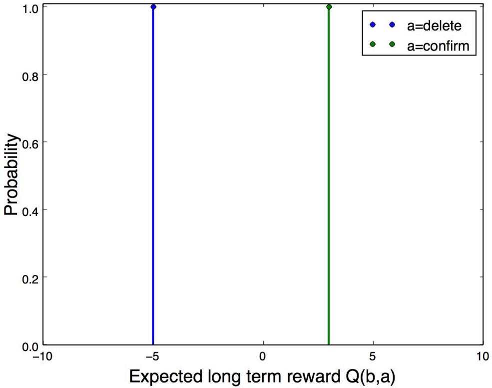
Voicemail

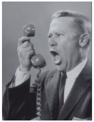
- States: The user wants the message saved, deleted or the dialogue is finished
- System actions: save the message, delete the message or confirm what the user wants

Q-function estimate without uncertainty

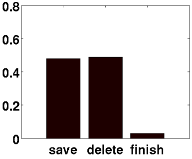


Belief state b

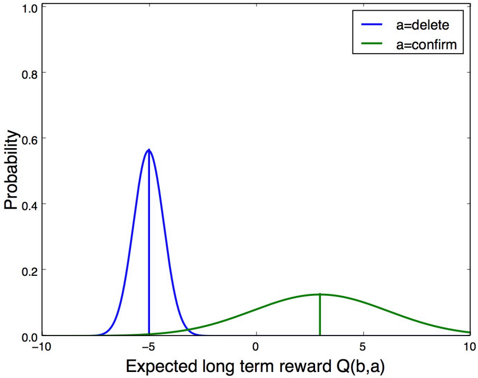


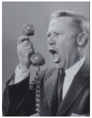


Q-function estimate with uncertainty

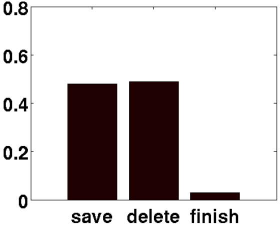
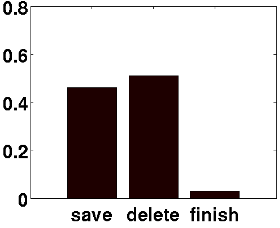


Belief state b





Role of the kernel function

Belief state	Action	Q-value
	Confirm	3
	Confirm	

Gaussian process model for Q-function [Engel et al., 2005]

- Expected return can be expressed iteratively

$$R_t^\pi = \sum_{i=0}^{\mathrm{T}} \gamma^i r_{t+i+1} = r_{t+1} + \gamma R_{t+1}^\pi$$

- Q-function is the expectation of the return

$$Q^\pi(\mathbf{b}, a) = E_\pi\left(R_t | b(s_t) = \mathbf{b}, a_t = a\right)$$

- Return can be modelled as the Q-value and residual ΔQ^π

$$R_t^\pi(\mathbf{b}, a) = Q^\pi(\mathbf{b}, a) + \Delta Q^\pi(\mathbf{b}, a).$$

- Relationship between immediate reward and Q-value is then:

$$r_{t+1}(\mathbf{b}, a) = Q^\pi(\mathbf{b}, a) - \gamma Q^\pi(\mathbf{b}', a') + \Delta Q^\pi(\mathbf{b}, a) - \gamma \Delta Q^\pi(\mathbf{b}', a')$$

Relationship between immediate rewards and Q-values

$$
\begin{aligned}
r^1 &= Q^\pi(\mathbf{b}^0, a^0) - \gamma Q^\pi(\mathbf{b}^1, a^1) \\
&\quad + \Delta Q^\pi(\mathbf{b}^0, a^0) - \gamma \Delta Q^\pi(\mathbf{b}^1, a^1) \\
r^2 &= Q^\pi(\mathbf{b}^1, a^1) - \gamma Q^\pi(\mathbf{b}^2, a^2) \\
&\quad + \Delta Q^\pi(\mathbf{b}^1, a^1) - \gamma \Delta Q^\pi(\mathbf{b}^2, a^2) \\
&\vdots \\
r^t &= Q^\pi(\mathbf{b}^{t-1}, a^{t-1}) - \gamma Q^\pi(\mathbf{b}^t, a^t) \\
&\quad + \Delta Q^\pi(\mathbf{b}^{t-1}, a^{t-1}) - \gamma \Delta Q^\pi(\mathbf{b}^t, a^t),
\end{aligned}
$$

Relationship between immediate rewards and Q-values

$$\mathbf{r}_t = \mathbf{H}_t \mathbf{q}_t^{\pi} + \mathbf{H}_t \boldsymbol{\Delta} \mathbf{q}_t^{\pi},$$

where

$$\begin{aligned}
\mathbf{r}_t &= [r^1, \ldots, r^t]^{\mathsf{T}} \\
\mathbf{q}_t^{\pi} &= [Q^{\pi}(\mathbf{b}^0, a^0), \ldots, Q^{\pi}(\mathbf{b}^t, a^t)]^{\mathsf{T}}, \\
\boldsymbol{\Delta} \mathbf{q}_t^{\pi} &= [\Delta Q^{\pi}(\mathbf{b}^0, a^0), \ldots, \Delta Q^{\pi}(\mathbf{b}^t, a^t)]^{\mathsf{T}}, \\
\mathbf{H}_t &= \begin{bmatrix} 1 & -\gamma & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & -\gamma \end{bmatrix}.
\end{aligned}$$

Gaussian process model for Q-function

Prior $Q^{\pi}(\mathbf{b}, a) \sim \mathcal{GP}\left(0, k((\mathbf{b}, a), (\mathbf{b}, a))\right)$,
$\Delta Q^{\pi}(\mathbf{b}, a) \sim \mathcal{N}(0, \sigma^2)$

Observations Belief-action pairs $\mathbf{B}_t = [(\mathbf{b}^0, a^0), \ldots, (\mathbf{b}^t, a^t)]^{\mathsf{T}}$
immediate rewards $\mathbf{r}_t = [r^1, \ldots, r^t]$

Posterior $Q^{\pi}(\mathbf{b}, a) | \mathbf{r}_t, \mathbf{B}_t$

Posterior of the Q-function

$$
\begin{aligned}
&Q^{\pi}(\mathbf{b}, a)|\mathbf{r}_t, \mathbf{B}_t \sim \mathcal{GP}(\overline{Q}(\mathbf{b}, a), \mathrm{cov}((\mathbf{b}, a), (\mathbf{b}, a))),\\
&\overline{Q}(\mathbf{b}, a) = \mathbf{k}_t(\mathbf{b}, a)^{\mathsf{T}} \mathbf{H}_t^{\mathsf{T}} (\mathbf{H}_t \mathbf{K}_t \mathbf{H}_t^{\mathsf{T}} + \sigma^2 \mathbf{H}_t \mathbf{H}_t^{\mathsf{T}})^{-1} \mathbf{r}_t,\\
&\mathrm{cov}((\mathbf{b}, a), (\mathbf{b}, a)) = k((\mathbf{b}, a), (\mathbf{b}, a))\\
&\qquad - \mathbf{k}_t(\mathbf{b}, a)^{\mathsf{T}} \mathbf{H}_t^{\mathsf{T}} (\mathbf{H}_t \mathbf{K}_t \mathbf{H}_t^{\mathsf{T}} + \sigma^2 \mathbf{H}_t \mathbf{H}_t^{\mathsf{T}})^{-1} \mathbf{H}_t \mathbf{k}_t(\mathbf{b}, a)
\end{aligned}
$$

$$
\mathbf{k}_t(\mathbf{b}, a) = [k((\mathbf{b}^0, a^0), (\mathbf{b}, a)), \dots, k((\mathbf{b}^t, a^t), (\mathbf{b}, a))]^{\mathsf{T}}
$$

$$
\mathbf{K}_t = \begin{bmatrix} k((\mathbf{b}^0, a^0), (\mathbf{b}^0, a^0)) & \cdots & k((\mathbf{b}^0, a^0), (\mathbf{b}^t, a^t)) \\ \vdots & \ddots & \vdots \\ k((\mathbf{b}^0, a^0), (\mathbf{b}^t, a^t)) & \cdots & k((\mathbf{b}^t, a^t), (\mathbf{b}^t, a^t)) \end{bmatrix}
$$

Applying this to an on-line setting

Computational complexity – need to invert Gram matrix $\mathbf{K}_t$

Sequential nature of data – need to perform updates sequentially

Kernel function – need to define correlations

GP-Sarsa algorithm

- Gram matrix is approximated with a dictionary of representative points
- Updates take place every time a reward is observed
- Kernel function is decomposed into separate kernels over belief states and actions

$$k((\mathbf{b}, a), (\mathbf{b}, a)) = k_{\mathcal{B}}(\mathbf{b}, \mathbf{b}) k_{\mathcal{A}}(a, a)$$

Sparcification

- Kernel function is a dot product of potentially infinite set of feature functions $\phi(\mathbf{b}, a) = [\phi_1(\mathbf{b}, a), \phi_2(\mathbf{b}, a), \ldots]^\mathsf{T}$

$$k((\mathbf{b}, a), (\mathbf{b}, a)) = \langle \phi(\mathbf{b}, a), \phi(\mathbf{b}, a) \rangle$$

- Gram matrix $\mathbf{K}_t$ is approximated with Gram matrix over dictionary points $\tilde{\mathbf{K}}_t$ and coefficients $\mathbf{G}_t = [\mathbf{g}_1, \ldots, \mathbf{g}_t]$

$$\mathbf{K}_t = \mathbf{\Phi}_t^\mathsf{T} \mathbf{\Phi}_t \approx \mathbf{G}_t \tilde{\mathbf{K}}_t \mathbf{G}_t^\mathsf{T}$$

- Dimensionality of $\tilde{\mathbf{K}}_t$ is $m << t$

Policy

- For given $\mathbf{b}$, for each action a, there is a Gaussian distribution $\hat{Q}(\mathbf{b}, a) \sim \mathcal{N}(\overline{Q}(\mathbf{b}, a), \mathrm{cov}((\mathbf{b}, a), (\mathbf{b}, a))))$
- Sampling from these Gaussian distributions gives Q-values $\left\{\hat{Q}(\mathbf{b}, a) : a \in \mathcal{A}\right\}$
- The highest sampled Q-value can then be selected:

$$\pi(\mathbf{b}) = \arg\max_{a} \left\{\hat{Q}(\mathbf{b}, a) : a \in \mathcal{A}\right\}$$

- This balances exploration and exploitation during learning

Kernel function

Action kernel Action space is reduced to summary space and then kernel is simple δ function: $k(a, a') = \delta_a(a')$

Belief state kernel Options:

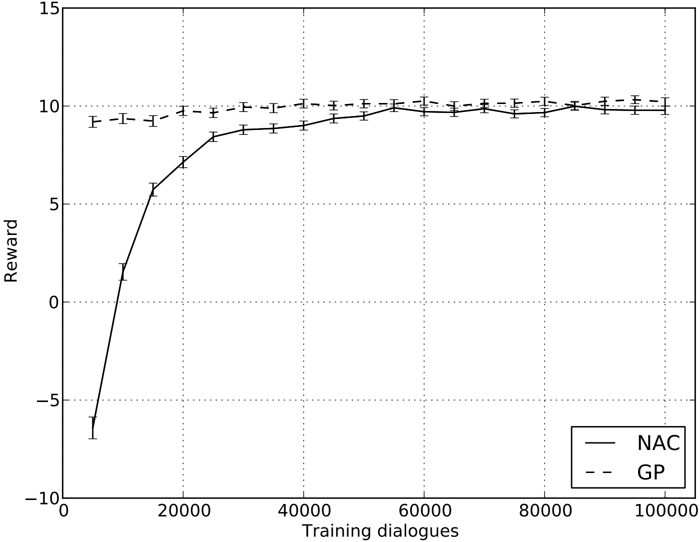
- Reduce to summary space and then calculate kernel on summary space
- Calculate the kernel directly on the full belief space
- For continuous variables use linear or Gaussian kernel

GP-Sarsa algorithm

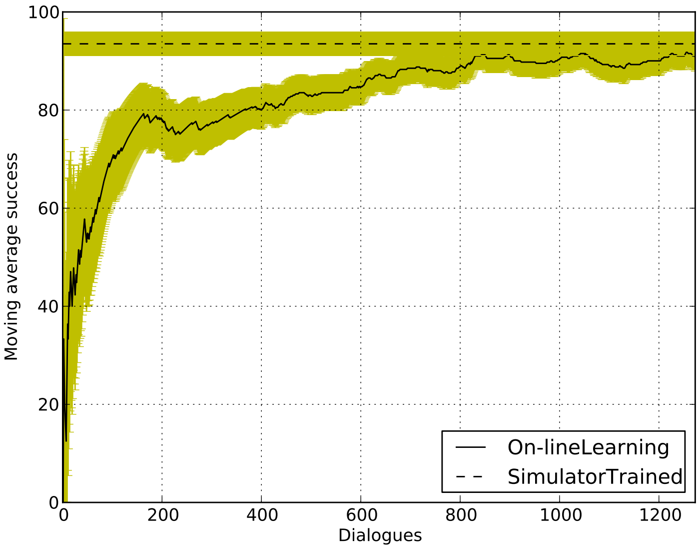
Algorithm 1 GP-Sarsa algorithm

Define prior for Q-function
for each dialogue **do**
  Initialise $\mathbf{b}$ and choose a according to current Q estimate
  if $(\mathbf{b}, a)$ is representative add to dictionary
  for each turn **do**
    Take action a observe r and next belief state $\mathbf{b}'$
    Choose a' according to current Q estimate
    if $(\mathbf{b}', a')$ is representative add to dictionary
    Update posterior mean and variance of Q
    $\mathbf{b}' \rightarrow \mathbf{b}$, $a \rightarrow a'$
  end for
end for

Comparison with NAC in a dialogue system [Gasic and Young, 2014]



Learning from real users [Gasic and Young, 2014]



Summary

- Q-function is modelled as a Gaussian process allowing posterior mean and variance to be calculated every time a reward is observed
- GP-Sarsa is a model-free, on-line algorithm which allows tractable approximation to the Gaussian process model for Q-function
- With adequate choice of the kernel function learning speed can be significantly improved
- Kernel function can be defined directly on belief state space

References I

Engel, Y., Mannor, S., and Meir, R. (2005).
Reinforcement learning with Gaussian processes.
In *Proceedings of ICML*.

Gasic, M. and Young, S. (2014).
Gaussian processes for pomdp-based dialogue manager optimization.
Audio, Speech, and Language Processing, IEEE/ACM Transactions on, 22(1):28–40.