Knowledge transfer between speakers for personalised dialogue management

Iñigo Casanueva, Thomas Hain, Heidi Christensen, Ricard Marxer and Phil Green
Department of Computer Science, University of Sheffield, United Kingdom
{i.casanueva, t.hain, h.christensen, r.marxer, p.green}@sheffield.ac.uk

Abstract

Model-free reinforcement learning has been shown to be a promising data driven approach for automatic dialogue policy optimization, but a relatively large amount of dialogue interactions is needed before the system reaches reasonable performance. Recently, Gaussian process based reinforcement learning methods have been shown to reduce the number of dialogues needed to reach optimal performance, and pre-training the policy with data gathered from different dialogue systems has further reduced this amount. Following this idea, a dialogue system designed for a single speaker can be initialised with data from other speakers, but if the dynamics of the speakers are very different the model will have a poor performance. When data gathered from different speakers is available, selecting the data from the most similar ones might improve the performance. We propose a method which automatically selects the data to transfer by defining a similarity measure between speakers, and uses this measure to weight the influence of the data from each speaker in the policy model. The methods are tested by simulating users with different severities of dysarthria interacting with a voice enabled environmental control system.

1 Introduction

Partially observable Markov decision processes (POMDP) (Young et al., 2013) are a popular framework to model dialogue management as a reinforcement learning (RL) problem. In a POMDP, a state tracker (Thomson and Young, 2010)(Williams, 2014) maintains a distribution over possible user goals (states), called the belief state, and RL methods (Sutton and Barto, 1998) are used to optimize a metric called cumulative reward, a score that combines dialogue success rate and dialogue length. However, existing model-based RL approaches become intractable for real world sized dialogue systems (Williams and Young, 2007), and model-free approaches often need a large number of dialogues to converge to the optimal policy (Jurčíček et al., 2012).

Recently, Gaussian process (GP) based RL (Engel et al., 2005) has been proposed for dialogue policy optimization, reducing the number of interactions needed to converge to the optimal policy by an order of magnitude with respect to other POMDP models, allowing the policy to be learned directly from real users interactions (Gašić et al., 2013 a). In addition, using transfer learning methods (Taylor and Stone, 2009) to initialise the policy with data gathered from dialogue systems in different domains has increased the learning speed of the policy further (Gašić et al., 2013 b), and provided an acceptable system performance when there is no domain specific data available. In the case of dialogue managers personalised for a single speaker, data gathered from other "source" speakers can be used to pre-train the policy, but if the dynamics of the other speakers are very different, this data will have a different distribution than the data of the current "target" speaker, and therefore, using this data to train the policy model does not have any benefit. In the context of speaker specific acoustic models for users with dysarthria (a speech impairment), Christensen et al. (2014) demonstrated that using a speaker similarity metric to select the data to train the acoustic models improves ASR performance. Taking this idea into dialogue management, if a similarity metric is defined between different speakers, this metric can be used to select which data from the source speakers is used to train the model, and even to weight the influence of the data from each speaker in the model. As GP-RL is a non-parametric

Proceedings of the SIGDIAL 2015 Conference, pages 12–21,
Prague, Czech Republic, 2-4 September 2015.

method, a straightforward way to transfer knowledge is to directly initialise the GP model for the target speaker using data from source speakers, and update the GP with the data from the target speaker as this is gathered through interaction. But GP-RL soon becomes intractable as the data amount increases, limiting the amount of data that can be transferred. Gašić et al. (2013 a) proposes to transfer knowledge between domains by using the source data to train a prior GP, whose posterior is used as prior mean in the new GP. Another option is to use a GP approximation method (Quiñonero and Rasmussen, 2005) which permits data selection, use the speaker similarity metric to select the source data to initialise the policy, and then discard source data points as data points from the target speaker become available, keeping the number of data points up to a maximum.

This paper investigates knowledge transfer between speakers in the context of a spoken environmental control system personalised for speakers with dysarthria (Christensen et al., 2013), where the ASR is adapted as speaker specific data is gathered (Christensen et al., 2012), thus improving the ASR performance with usage. The paper is organised as follows: Section 2 gives the background of GP-RL and defines the methods to select and weight the transferred data. Section 3 presents the experimental setup of the environmental control system and the different dysarthric simulated users, as well as the different features used to define the speaker similarities. In Section 4 the results of the experiments are presented and explained and Section 5 concludes the paper.

2 GPs for reinforcement learning

The objective of a POMDP based dialogue manager is to find the policy $\pi(\mathbf{b}) = a$ that maximizes the expected cumulative reward c_i defined as the sum of immediate rewards from time step i until the dialogue is finished, where $a \in \mathcal{A}$ is the action taken by the manager, and the *belief state* $\mathbf{b}$ is a probability distribution over a discrete set of states $\mathcal{S}$. The *Q-function* defines the expected cumulative reward when the dialogue is in belief state $\mathbf{b}_i$ and action a_i is taken, following policy π:

$$Q(\mathbf{b}_i, a_i) = E_\pi[c_i] \text{ ; where } c_i = \sum_{n=i}^{N} \gamma^{n-i} r_n \quad (1)$$

where N is the time step at which the terminal action is taken (end of the dialogue), r_i is the immediate reward given by the reward function, and $0 \leq \gamma \leq 1$ is the discount factor, which weights future rewards. If c_i is considered to be a random variable, it can be modelled as a mean plus a residual, $c_i = Q(\mathbf{b}_i, a_i) + \Delta Q(\mathbf{b}_i, a_i)$. Then the immediate reward r_i can be written recursively as the temporal difference (TD) between Q at time i and $i + 1$:

$$\begin{aligned} r_i = {} & Q(\mathbf{b}_i, a_i) + \Delta Q(\mathbf{b}_i, a_i) \\ & -\gamma_i Q(\mathbf{b}_{i+1}, a_{i+1}) - \gamma_i \Delta Q(\mathbf{b}_{i+1}, a_{i+1}) \end{aligned} \quad (2)$$

where $\gamma_i = 0$ if a_i is a terminal action[1], and the discount factor γ otherwise. Given a set of observed *belief-action* points $(\mathbf{b}_i, a_i)$, with their respective r_i values, the set of linear equations can be represented in matrix form as:

$$\mathbf{r}_{t-1} = \mathbf{H}_t \mathbf{q}_t + \mathbf{H}_t \Delta \mathbf{q}_t \quad (3)$$

where $\mathbf{q}_t$=$[Q(\mathbf{b}_1, a_1), Q(\mathbf{b}_2, a_2), ..., Q(\mathbf{b}_t, a_t)]^\top$, $\Delta\mathbf{q}_t$=$[\Delta Q(\mathbf{b}_1, a_1), \Delta Q(\mathbf{b}_2, a_2), ..., \Delta Q(\mathbf{b}_t, a_t)]^\top$, $\mathbf{r}_{t-1} = [r_1, r_2, ..., r_{t-1}]^\top$ and

$$\mathbf{H}_t = \begin{bmatrix} 1 & -\gamma_1 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\gamma_{t-1} \end{bmatrix}$$

If the random variables $\mathbf{q}_t$ are assumed to have a joint Gaussian distribution with zero mean and $\Delta Q(\mathbf{b}_i, a_i) \sim \mathcal{N}(0, \sigma^2)$, the system can be modelled as a GP (Rasmussen and Williams, 2005), with the covariance matrix determined by a *kernel function* defined independently over the belief and the action space (Engel et al., 2005):

$$k_{i,j} = k((\mathbf{b}_i, a_i), (\mathbf{b}_j, a_j)) = k^b(\mathbf{b}_i, \mathbf{b}_j) k^a(a_i, a_j) \quad (4)$$

To simplify the notation, from now on $\mathbf{x}_i = (\mathbf{b}_i, a_i)$ will be defined as each belief-action point, and $\mathbf{K}_{Y,Y'}$ as the matrix of size $|\mathbf{Y}| \times |\mathbf{Y}'|$ whose elements are computed by the kernel function (eq. 4) between any set of points $\mathbf{Y}$ and $\mathbf{Y}'$. For a new belief-action point $\mathbf{x}_* = (\mathbf{b}_*, a_*)$, the posterior of the expected cumulative reward can be computed:

$$\begin{aligned} & Q(\mathbf{x}_*) | \mathbf{X}_t, \mathbf{r}_{t-1} \sim \mathcal{N}(\bar{Q}(\mathbf{x}_*), \hat{Q}(\mathbf{x}_*)) \\ & \bar{Q}(\mathbf{x}_*) = \mathbf{K}_{*,X} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{K}_{X,X} \mathbf{H}_t^\top + \mathbf{\Sigma}_t)^{-1} \mathbf{r}_{t-1} \\ & \hat{Q}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) \\ & \quad - \mathbf{K}_{*,X} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{K}_{X,X} \mathbf{H}_t^\top + \mathbf{\Sigma}_t)^{-1} \mathbf{H}_t \mathbf{K}_{X,*} \end{aligned} \quad (5)$$

[1]As dialogue management is an episodic RL problem, the temporal difference relationship between 2 consecutive belief-action points only happens if the points belong to the same dialogue.

where $\mathbf{X}_t$ is the set of size t of all the previously visited $(\mathbf{b}_i, a_i)$ points, $*$ denotes the set of size 1 composed by the new belief-action point to be evaluated and $\mathbf{\Sigma}_t = \sigma^2 \mathbf{H}_t \mathbf{H}_t^\top$. $\bar{Q}$ and $\hat{Q}$ represent the mean and the variance of Q respectively.

To further simplify the notation it is possible to redefine eq. 5 by defining a kernel in the temporal difference space instead of in the belief-action space. If the set of belief-action points $\mathbf{X}_t$ is redefined[2] as $\mathbf{Z}_t$ where $\mathbf{z}_i = (\mathbf{b}_i, a_i, \mathbf{b}_{i+1}, a_{i+1})$, with $\mathbf{b}_{i+1}$ and a_{i+1} set to any default values if a_i is a terminal action, a kernel function between 2 temporal difference points can be defined as:

$$\begin{aligned} k^{td}_{i,j} &= k^{td}(\mathbf{z}_i, \mathbf{z}_j) \\ &= k^{td}((\mathbf{b}_i, a_i, \mathbf{b}_{i+1}, a_{i+1}), (\mathbf{b}_j, a_j, \mathbf{b}_{j+1}, a_{j+1})) \\ &= (k_{i,j} + \gamma_i \gamma_j k_{i+1,j+1} - \gamma_i k_{i+1,j} - \gamma_j k_{i,j+1}) \end{aligned} \tag{6}$$

where $k_{i,j}$ is the kernel function in the belief-action space (eq. 4) and $\gamma_i = 0$ and $\gamma_j = 0$ if a_i and a_j are terminal actions respectively, or the discount factor γ otherwise (as in eq. 2). When a_i is a terminal action, the value of a_{i+1} and $\mathbf{b}_{i+1}$ in $\mathbf{z}_i$ is irrelevant, as it will be multiplied by $\gamma_i = 0$. In the same way, when this kernel is used to compute the covariance vector between a new test point and the set $\mathbf{Z}_t$, as the new point $\mathbf{x}_* = (\mathbf{b}_*, a_*)$ lies in the belief-action space, it is redefined as $\mathbf{z}_* = (\mathbf{b}_*, a_*, \mathbf{b}_{*+1}, a_{*+1})$ with $\mathbf{b}_{*+1}$ and a_{*+1} set to default values. Then, a_* is considered a terminal action, so $\mathbf{b}_{*+1}$ and a_{*+1} won't affect the value of k^{td}_{i*} due to $\gamma_* = 0$. A more detailed derivation of the temporal difference kernel is given in appendix A. Using the temporal difference kernel defined in eq. 6, eq. 5 can be rewritten as:

$$\begin{aligned} &Q(\mathbf{z}_*)|\mathbf{Z}_t, \mathbf{r}_{t-1} \sim \mathcal{N}(\bar{Q}(\mathbf{z}_*), \hat{Q}(\mathbf{z}_*)) \\ &\bar{Q}(\mathbf{z}_*) = \mathbf{K}^{td}_{*,Z}(\mathbf{K}^{td}_{Z,Z} + \mathbf{\Sigma}_t)^{-1}\mathbf{r}_{t-1} \\ &\hat{Q}(\mathbf{z}_*) = k^{td}(\mathbf{z}_*, \mathbf{z}_*) - \mathbf{K}^{td}_{*,Z}(\mathbf{K}^{td}_{Z,Z} + \mathbf{\Sigma}_t)^{-1}\mathbf{K}^{td}_{Z,*} \end{aligned} \tag{7}$$

where $\mathbf{K}^{td}_{Y,Y'}$ is the covariance matrix computed with the temporal difference kernel between any set of TD points $\mathbf{Y}$ and $\mathbf{Y}'$. With this notation, the shape of the equation for the posterior of Q is equivalent to classic GP regression models. Thus, it is straightforward to apply a wide range of well studied GP techniques, such as sparse methods. Redefining the belief-action set of points $\mathbf{X}_t$ as the set of temporal difference points $\mathbf{Z}_t$ also simplifies the selection of data points (e.g. to select inducing points in sparse models), because the dependency between consecutive points is well defined.

The GP literature proposes various *sparse* methods which select a subset of *inducing points* $\mathbf{U}$ of size $m < t$ from the set of training points $\mathbf{Z}$ (Quiñonero and Rasmussen, 2005). In this paper the deterministic training conditional (DTC) method is used. Once the subset of points has been selected and assuming $\Delta Q(\mathbf{b}_i, a_i) - \gamma_i \Delta Q(\mathbf{b}_{i+1}, a_{i+1}) \sim \mathcal{N}(0, \sigma^2)$ as in (Engel et al., 2003), the GP posterior can be approximated in $\mathcal{O}(t \cdot m^2)$ with the DTC method as:

$$\begin{aligned} &Q^{dtc}(\mathbf{z}_*)|\mathbf{Z}_t, \mathbf{r}_{t-1} \sim \mathcal{N}(\bar{Q}^{dtc}(\mathbf{z}_*), \hat{Q}^{dtc}(\mathbf{z}_*)) \\ &\bar{Q}^{dtc}(\mathbf{z}_*) = \sigma^{-2}\mathbf{K}^{td}_{*,U}\mathbf{\Lambda}\mathbf{K}^{td}_{U,Z}\mathbf{r}_{t-1} \\ &\hat{Q}^{dtc}(\mathbf{z}_*) = k^{td}(\mathbf{z}_*, \mathbf{z}_*) - \mathbf{\Phi} + \mathbf{K}^{td}_{*,U}\mathbf{\Lambda}\mathbf{K}^{td}_{U,*} \end{aligned} \tag{8}$$

where $\mathbf{\Lambda} = (\sigma^{-2}\mathbf{K}^{td}_{U,Z}\mathbf{K}^{td}_{Z,U} + \mathbf{K}^{td}_{U,U})^{-1}$ and $\mathbf{\Phi} = \mathbf{K}^{td}_{*,U}(\mathbf{K}^{td}_{U,U})^{-1}\mathbf{K}^{td}_{U,*}$.

Once the posterior for any new belief-action point can be computed with eq. 7 or eq. 8, the policy $\pi(\mathbf{b}) = a$ can be computed as the action a that maximizes the Q-function from the current belief state $\mathbf{b}_*$, but in order to avoid getting stuck in a local optimum, an exploration-exploitation approach should be taken. One of the advantages of GPs is that they compute the uncertainty of the expected cumulative reward in form of a variance, which can be used as a metric for *active exploration* (Geist and Pietquin, 2011) to speed up the learning of the policy with an ϵ-greedy approach:

$$\pi(\mathbf{b}_*) = \begin{cases} \arg\max_{a \in \mathcal{A}} \bar{Q}(\mathbf{b}_*, a) \text{ with prob. } (1 - \epsilon) \\ \arg\max_{a \in \mathcal{A}} \hat{Q}(\mathbf{b}_*, a) \text{ with prob. } \epsilon \end{cases} \tag{9}$$

where ϵ controls the exploration rate. The policy optimization loop is performed following the *Episodic GP-Sarsa* algorithm defined by (Gašić and Young, 2014).

2.1 Transfer learning with GP-RL

The scenario where a statistical model for a specific "target" task must be trained, but only data from different but related "source" tasks is available, is known as transfer learning (Pan and Yang, 2010). In the context of this paper the different tasks will be dialogues with different speakers, and three points of transfer learning will be addressed:

- *How to transfer the knowledge*
- In the case of multiple source speakers, *which data to transfer*, and

[2] Take into account that $|\mathbf{Z}_t| = |\mathbf{X}_t| - 1$

- *How to weight data from different sources.*

In the context of reinforcement learning (Taylor and Stone, 2009) and dialogue policy optimization (Gašić et al., 2013 a), transfer learning has been shown to increase the performance of the system in the initial stages of use and to speed up the policy learning, requiring a smaller amount of target data to reach the optimal policy.

2.1.1 Knowledge transfer

The most straightforward way to transfer the data in GP-RL is to initialise the set of temporal difference points $\mathbf{Z_t}$ of the GP with the source points and then continue updating it with target data points as they are gathered through interaction. However, this approach has a few shortcomings. First, as GP-RLs complexity increases with the number of data points, the model might quickly become intractable if it is initialised with too many source points. Also, when data points from the target speaker are gathered through interaction, the source points may not improve the performance of the system, while increasing the model complexity. Second, as the computation of the variance for a new point depends on the number of close points already visited, the variance of the new belief-action points will be reduced by the effect of the source points close in the belief-action space. If the distribution of the source data points is unbalanced, the effectiveness of the policy of eq. 9 will be affected. Gašić et al. (2013 a) proposes to use the source points to train a prior GP, and use its posterior as mean function for the GP trained with the target points. With this approach, the mean of the posterior in eq. 7 will be modified as:

$$\bar{Q}(\mathbf{z}_*) = m(\mathbf{z}_*) + \mathbf{K}^{td}_{*,Z}(\mathbf{K}^{td}_{Z,Z} + \Sigma)^{-1}(\mathbf{r}_{t-1} - \mathbf{m}_t) \tag{10}$$

where $m(\mathbf{z}_*)$ is the mean of the posterior of the Q-function given by the prior GP and $\mathbf{m}_t = [m(\mathbf{z}_0), ..., m(\mathbf{z}_t)]^\top$. If the DTC approach (eq. 8) is taken, the posterior Q-function mean becomes:

$$\bar{Q}^{dtc}(\mathbf{z}_*) = m(\mathbf{z}_*) + \sigma^{-2}\mathbf{K}^{td}_{*,U}\mathbf{\Lambda}\mathbf{K}^{td}_{U,Z}(\mathbf{r}_{t-1} - \mathbf{m}_t) \tag{11}$$

This approach has the advantage of being computationally cheaper than the former method while modelling the uncertainty for new target points more accurately, but at the cost of not taking into account the correlation between source and target points, which might reduce the performance when there is a small amount of target data.

A third approach combines the two previous methods, using a portion of the transfer points to train a GP for the prior mean function, while the rest is used to initialise the set $\mathbf{Z}_t$ of the GP that will be updated with target points. This method will be computationally cheaper than the first one while increasing the performance of the second method with a small amount of target data.

2.1.2 Transfer data selection

As non-parametric models, the complexity of GPs will increase with the number of data points, limiting the amount of source data that can be transferred. Additionally, if the points come from multiple sources, it is possible that the data distribution from some sources is more similar to the target speaker than others, hence transferring data from these sources will increase performance. We propose to extract a speaker feature vector $\mathbf{s}$ from each speaker and define a similarity function $f(\mathbf{s}, \mathbf{s}')$ between speakers (see sec. 3.4). The data can be selected by choosing the points from the source speakers more similar to the target.

With the DTC approach (eq. 8), a subset of inducing points $\mathbf{U}_m$ must be selected. The most straightforward way is to select the most similar points to the speaker from the transferred points. As the user interacts with the system and target data points are gathered, these points may be used as inducing points. This approach acts like another layer of data selection; the reduced complexity will allow for the transfer of more source points, while using the target points as inducing points will mean that only the source points that lie in the same part of the belief-action space as the target points have influence on the model.

2.1.3 Transfer data weighting

When transferring data from multiple sources, the similarity between each source and the target speaker might be different. Thus the data from a source more similar to the target should have more influence in the model than less similar ones. As a GP is defined by computing covariances between data points through a kernel function, one way to weight the data from different sources is to extend the belief-action vector used to compute the covariance with the speaker feature vector $\mathbf{s}$ explained in the previous section as $\mathbf{x}_i = (\mathbf{b}_i, a_i, \mathbf{s}_i)$, and then extend the kernel (eq. 4) by multiplying it by a new kernel in the speaker space k^s as:

$$k_{i,j}^{ext} = k((\mathbf{b}_i, a_i, \mathbf{s}_i), (\mathbf{b}_j, a_j, \mathbf{s}_j)) \\ = k^b(\mathbf{b}_i, \mathbf{b}_j) k^a(a_i, a_j) k^s(\mathbf{s}_i, \mathbf{s}_j) \tag{12}$$

By adding this extra space to the data points, the covariance between points will not only depend on the similarity between points in the belief-action space, but also in the speaker space, reducing the covariance between two points that lie in different parts of the speaker space. This approach will also help to partially deal with the variance computing problem of the first model in sec. 2.1.1, as the source points will lie on a different part of the speaker space than the new target points, thus having less influence in the variance computation.

3 Experimental setup

To test the system in a scenario with high variability between the dynamics of the speakers, the experiments are performed within the context of a voice-enabled control system designed to help speakers with dysarthria to interact with their home devices (TV, radio, lamps...), where the speakers have different severities of dysarthria (this is an instance of the homeService application (Christensen et al., 2013)). The system has a vocabulary of 36 commands and is organised in a tree setup where each node in the tree represents either a device (e.g. "TV"), a property of that device (e.g. "channel"), or actions that trigger some change in one of the devices (e.g. "one", child of "channel", will change the TV to channel one). When the system transitions to one of the terminal nodes that trigger an action, the action associated with this node is performed, and subsequently the system returns to the root node. In the following experiments a dialogue will be considered finished when one of the *terminal node actions* is carried out. In the non-terminal nodes, the user may either speak one of the commands available in that node (defined by its children nodes) to transition to them, or say the meta-command "back" to return to its parent node. The ASR is configured to recognise single words, so there is no need for a language understanding system, as the concepts are just a direct mapping from the ASR output. A more detailed explanation of the system is given in (Casanueva et al., 2014) and two example dialogues are presented in Appendix B.

3.1 Simulated dysarthric users

In the homeService application, each system is personalised for a single speaker by adapting the ASR system's acoustic model as more data is gathered through interaction, thus increasing the accuracy of the ASR over time. In the following experiments, the system is tested by interacting with a set of *simulated users* with dysarthria, where each user interacts with a set of different ASR simulators, arising from the different amounts of data used to adapt the ASR. To train the ASR simulator for these users, data from a dysarthric speech database (UASpeech database (Kim et al., 2008)) has been used. Table 1 shows the characteristics of the 15 speakers of the database, and the ASR accuracy for each speaker in the 36 word vocabulary of the system without adaptation and adapted with 500 words from that speaker. Additionally, an intelligibility measure assessment is presented for each speaker as the percentage of words spoken by each speaker which are understood by unfamiliar speakers; these are shown in the second column in table 1.

The system is tested with 6 different simulated users trained with data from low and medium intelligibility[3] speakers. Each user interacts with 4 different ASRs, adapted with 0, 150, 300 and 500 words respectively. For a more detailed explanation of the simulated users configuration, the reader may refer to (Casanueva et al., 2014).

3.2 POMDP setup

Each non-terminal node in the tree is modelled as an independent POMDP where the state set $\mathcal{S}$ is the set of possible goals of the node and the action set $\mathcal{A}$ is the set of actions associated with each goal plus an "ask" action, which requests the user to repeat his last command. The reward function for all the POMDPs is -1 for the "ask" action, and +10 for each other action if it corresponds to the user goal, or -10 otherwise, and $\gamma = 0.95$. The state tracker is a logistic regression classifier (Pedregosa et al., 2011), where classes are the set of states $\mathcal{S}$. The belief state $\mathbf{b}$ is computed as the posterior over the states given the last 5 observations (N-best lists with normalised confidence scores). For each speaker, the state tracker has been trained with data from the other 14 speakers.

[3] In (Casanueva et al., 2014) it was shown that, with a 36 command setup, statistical DM is most useful for low and medium intelligibility speakers. For high intelligibility speakers, the ASR accuracy is close to 100% so the improvement obtained from DM is small, and for very low intelligibility speakers, the absolute performance is not high enough to make the system useful.

Speaker intelligibility	Range of int. measures	Number of speakers	Speaker independent ASR accuracy range	Adapted ASR accuracy range
Very low	2% - 15%	4	12.04% - 46.80%	23.06% - 74.37%
Low	28% - 43%	3	27.04% - 55.99%	80.52% - 95.28%
Medium	58% - 62%	3	55.34% - 68.34%	85.93% - 89.61%
High	86% - 95%	5	68.14% - 97.76%	95.38% - 100.00%

Table 1: *Stats for the UASpeech database*

3.3 Policy models

The DTC approach (eq. 8) is used to compute the Q-function for the policy (eq. 9) with Gaussian noise variance $\sigma^2 = 5$. The kernel over the belief space is a radial basis function kernel (RBF):

$$k^b(\mathbf{b}_i, \mathbf{b}_j) = \sigma_k^2 \exp\left(- \frac{||\mathbf{b}_i - \mathbf{b}_j||^2}{2l_k^2} \right) \quad (13)$$

with variance $\sigma_k^2 = 25$ and lengthscale $l_k^2 = 0.5$. The delta kernel is used over the action space:

$$k^a(a_i, a_j) = \delta(a_i, a_j) = \begin{cases} 1 \text{ if } a_i = a_j \\ 0 \text{ otherwise} \end{cases} \quad (14)$$

and the kernels over the speaker space are defined in section 3.4. The size of the inducing set $\mathbf{U}_m$ is 500 and the maximum size of the TD points set $\mathbf{Z}_t$ is 2000. Whenever a new data point is observed from the target speaker, it is added to the set of inducing points $\mathbf{U}_m$, and the first point of the set $\mathbf{U}_m$ (which, due to the ordering done by data selection, corresponds to the least similar source point or to the oldest target point) is discarded from the inducing set. Whenever a new data point is observed and the size of the set of temporal difference points $|\mathbf{Z}_t| = 2000$, the first point of this set is discarded. Three variations of the DTC approach are used:

- *DTC*: Equation 8 is used to compute the Q posterior for the policy (eq. 9) and the set of temporal difference points $\mathbf{Z}_t$ is initialised with the source points.
- *Prior*: Equation 11 is used to compute the Q posterior for the policy (eq. 9) and the prior GP is trained with the source points.
- *Hybrid*: Equation 11 is used to compute the Q posterior for the policy (eq. 9), the prior GP is trained with half of the source points and the set of temporal difference points $\mathbf{Z}_t$ is initialised with the other half.

3.4 Speaker similarities

To compute the similarities between speakers a vector of speaker features $\mathbf{s}$ must be extracted. Different kinds of features may be extracted, such as meta-data based features, acoustic features, features related to the ASR performance, etc. In this paper, we explore 3 different methods to extract $\mathbf{s}$;

- *Intelligibility assessment*: The intelligibility assessment for each speaker in the UASpeech database (table 1) can be used as a single dimensional feature.
- *I-vectors*: Martínez et al. (2013) showed that *i-vectors* (Dehak et al., 2011) can be used to predict the intelligibility of a dysarthric speaker. For each speaker, $\mathbf{s}$ is defined as a 400 dimensional vector corresponding to the mean *i-vector* extracted from each utterance from that speaker. For more information on the *i-vector* extraction and characteristics, refer to (Martínez et al., 2014).
- *ASR accuracy*: The performance statistics of the ASR (e.g. accuracy) can be used as speaker features. In this paper we use the accuracy per word (command), defining $\mathbf{s}$ as a 36 dimensional vector where each element is the ASR accuracy for each of the 36 commands.

The kernel over the speaker space k^s (eq. 12), is defined as an RBF kernel (eq. 13). This kernel is used both to compute the similarity between speakers in order to select data (section 2.1.2), and to weight the data from each source speaker (section 2.1.3). k^s has variance $\sigma_k^2 = 1$ and the lengthscale l_k^2 varies depending on the features. For intelligibility features $l_k^2 = 0.5$, for *i-vector*s $l_k^2 = 8.0$ and for ASR accuracy features $l_k^2 = 4.0$

4 Results

In the following experiments the *reward* is computed as -1 for each dialogue turn, +20 if the dialogue was successful[4]. The system has been tested

[4]Because of the variable depth tree structure of the spoken dialogue system, the sum or average of cumulative rewards obtained in each sub-dialogue is not a good measure of the overall system performance. If the dialogue gets stuck in a loop going back and forth between two sub-dialogues, the extra amount of turns spent in this loop would not be reflected in the average of rewards

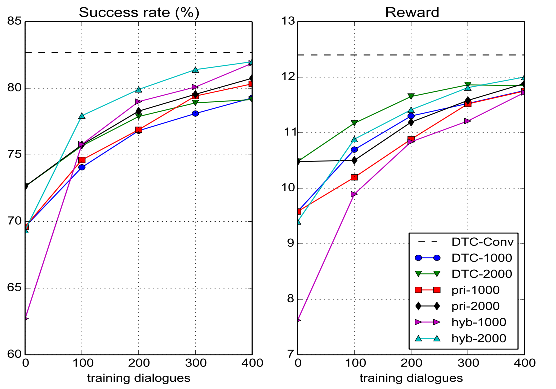

Figure 1: *different policy models compared*

with the 24 speaker-ASR pairs explained in section 3.1, and in the following figures, each plotted line is the average results for these 24 speaker-ASR pairs. As the behaviour of the simulated user and some data selection methods partially depend on random variables, each experiment has been initialised with four different seeds and all the results presented are the average of the four seeds tested over 500 dialogues. In all the experiments the data to initialise each POMDP is transferred from a pool of 4200 points corresponding to 300 points from each speaker in table 1 except the speaker being tested, where each data pool is different for each seed.

Figure 1 compares the different policy models presented in section 3.3 using the intelligibility measure based similarity to select and weight the data. The dotted line named *DTC-conv* shows the performance of the DTC policy when trained until convergence with the target speaker by simulating 1200 sub-dialogues in each node. *DTC-1000* and *DTC-2000* show the performance of the basic DTC approach when 1000 and 2000 source points are transferred respectively. It can be observed that, transferring more points boosts the performance, but at the cost of increasing the complexity. *pri-1000* and *pri-2000* show the performance of the prior policy with 1000 and 2000 transfer points respectively. The success rate is above the DTC policy but the learning rate for the reward is slower. This might be because the small amount of target data points make the predictions of the Q-function given by the GP unreliable. *Hyb-1000* and *hyb-2000* show the performance of the hybrid model, showing the best behaviour on success rate after 100 dialogues, and for *hyb-2000* even outperforming *DTC-2000* in reward after 400 dialogues.

In figure 2 the different approaches to compute the speaker similarities for data selection and weighting presented in section 3.4 are compared, using the DTC model with 1000 transfer points (named *DTC-1000* in the previous figure). *DTC-int* uses the intelligibility measure based features, *DTC-iv* the *i-vector* features and *DTC-acc* the ASR accuracy based features. *DTC-iv* outperforms the other two features, followed closely by *DTC-acc*. The performance of *DTC-int* is way below the other two metrics, suggesting that the information given by intelligibility assessments is a weak feature for source speaker selection (as it is done by humans, it might be very noisy). As *DTC-acc* uses information about the ASR statistics (which is the input for the dialogue manager), it might be expected that it will outperform the rest, but in this case a purely acoustic based measure such as the *DTC-iv* works better. The reason for this might be that these features are not correlated to the ASR performance, so hidden variables are used to better organise the data. To investigate the usefulness of similarity based data selection, two different data selection methods which do not weight the transferred data have been tried. *DTC-randspk* selects the ordering of the speakers from whom the data is transferred at random, and has a much worse performance than the similarity based method, but *DTC-allspk* selects the 1000 source points from all the speakers, selecting 1000 points at random from the pool of 4200 points and, as it can be seen, the reward obtained by this method is slightly better than with *DTC-iv*, even if the success rate is lower. This suggests that transferring points from more speakers rather than from just the closest ones is a better strategy, probably because points selected by this method are distributed more uniformly over the belief-action space. A method which does a trade-off between filling the belief-action space while selecting the most similar points could be a better option.

To further investigate the effect of selection and weighting of the data, figure 3 plots the results for the DTC policy model using the *i-vector* based similarity to weight the data but different data selection methods. *iv-clo* selects the closest speakers with respect to the *i-vector* metric, *iv-randspk* orders the speakers at random, and *iv-allspk* selects the 1000 transfer points from all the speakers but the tested one. As in the previous figure, selecting speakers by similarity works better than selecting speakers at random, but selecting the points from all the speakers and weighting them with the *i-vector* metric outperforms all the previous meth-

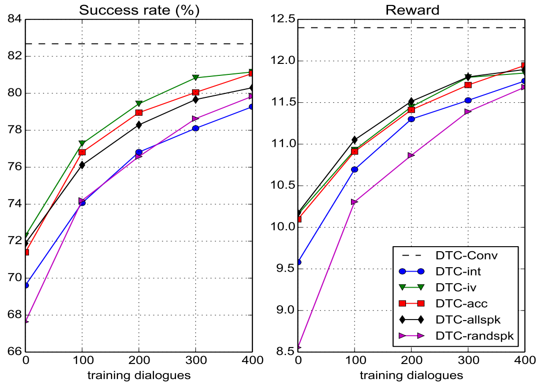

Figure 2: *different similarity metrics for data selection and weighting compared*

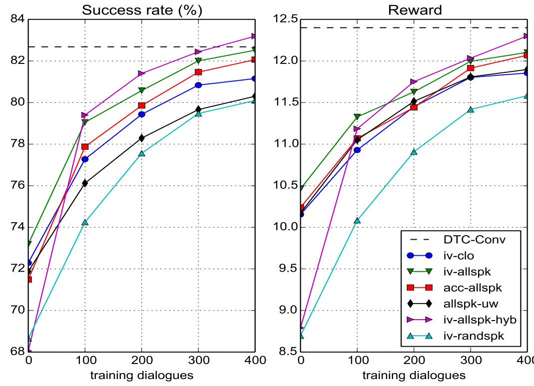

Figure 3: *different transfer data selection methods compared*

ods. This might be because weighting the data does a kind of data selection, as the data points from source speakers closer to the target will have more influence than the further ones, while transferring points from all the speakers covers a bigger part of the belief-action space. *acc-allspk* and *allspk-uw* show the results of weighting the data with the ASR accuracy metric and not weighting the data respectively, when selecting the data from all speakers. The accuracy metric performs worse than the *i-vector* metric once again, but it still outperforms not weighting the data, suggesting that data weighting works for different metrics. Finally *iv-allspk-hyb* plots the performance of the hybrid model when selecting the data from all the speakers and weighting it with the *i-vector* based similarity. Even if it is computationally cheaper, it outperforms *iv-allspk* after 100 dialogues, suggesting that with a good similarity metric and data selection method, the hybrid model in section 3.3 is the best option to take.

5 Conclusions

When transferring knowledge between speakers in a GP-RL based policy, weighting the data by using a similarity metric between speakers, and to a lesser extent, selecting the data using this similarity, improves the performance of the dialogue manager. By defining a kernel between temporal difference points and interpreting the Q-function as a GP regression problem where data points are in the TD space, sparse methods that allow the selection of the subset of inducing points such as DTC can be applied. In a transfer learning scenario, DTC permits a larger number of data points to be transferred and the selection of points collected from the target speaker as inducing points.

We showed that using part of the transferred data to train a prior GP for the mean function, and the rest to initialize the set of points of the GP, improves the performance of each of these approaches. Transferring data points from a larger number of speakers outperformed selecting the data points only from the more similar ones, probably because the belief-action space is covered better. This suggests that more complex data selection algorithms that trade-off between selecting the data points by similarity and covering more uniformly the belief-action space should be used. Also, increasing the amount of data transferred increased the performance, but the complexity increase of GP-RL limits the amount of data that can be transferred. More computationally efficient ways to transfer the data could be studied.

Of the three metrics based on speaker features tested (speaker intelligibility, *i-vectors* and ASR accuracy), *i-vectors* outperformed the rest. This suggest that *i-vectors* are a potentially good feature for speaker specific dialogue management and could be used in other tasks such as state tracking. ASR accuracy based metrics also outperformed the intelligibility based one, and as ASR accuracy and *i-vector* are uncorrelated features, a combination of them could give further improvement.

Finally, as the models were tested with simulated users in a hierarchically structured dialogue system (following the structure of the homeService application), future work directions include evaluating the policy models in a mixed initiative dialogue system and testing them with real users.

Acknowledgements

The research leading to these results was supported by the University of Sheffield studentship network PIPIN and EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology). The authors would like to thank David Martínez for providing the *i-vectors* used in this paper.

References

I. Casanueva, H. Christensen, T. Hain, and P. Green. 2014. *Adaptive speech recognition and dialogue management for users with speech disorders*. Proceedings of Interspeech.

H. Christensen, S. Cunningham, C. Fox, P. Green, and T. Hain. 2012. *A comparative study of adaptive, automatic recognition of disordered speech*. Proceedings of Interspeech.

H. Christensen, I. Casanueva , S. Cunningham, P. Green, and T. Hain. 2013. *homeService: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition*. Proceedings of SLPAT.

H. Christensen, I. Casanueva , S. Cunningham, P. Green, and T. Hain. 2014. *Automatic selection of speakers for improved acoustic modelling: recognition of disordered speech with sparse data*. Proceedings of SLT.

N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet. 2011. *Front-end factor analysis for speaker verification*. IEEE Transactions on Audio, Speech, and Language Processing.

Y. Engel, S. Mannor, R. Meir. 2003. *Bayes Meets Bellman: The Gaussian Process Approach to Temporal Difference Learning*. Proceedings of ICML.

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

M. Gašić, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis and S. Young. 2013. *On-line policy optimisation of Bayesian spoken dialogue systems via human interaction*. Proceedings of ICASSP.

M. Gašić, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis and S. Young. 2013. *POMDP-based dialogue manager adaptation to extended domains*. Proceedings of SIGDIAL.

M. Gašić and S. Young. 2014. *Gaussian Processes for POMDP-based dialogue manager opimisation*. IEEE Transactions on Audio, Speech and Language Processing.

M. Geist and O. Pietquin. 2011. *Managing uncertainty within the KTD framework*. Proceedings of JMLR.

F. Jurčíček, B. Thomson, and S. Young. 2012. *Reinforcement learning for parameter estimation in statistical spoken dialogue systems*. Computer Speech and Language.

H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. Huang, K. Watkin, and S. Frame. 2008. *Dysarthric speech database for universal access research*. Proceedings of Interspeech.

D. Martínez, P. Green and H. Christensen. 2013. *Dysarthria Intelligibility Assessment in a Factor Analysis Total Variability Space*. Proceedings of Interspeech.

D. Martínez, E. Lleida, P. Green, H. Christensen, A. Ortega and A. Miguel. 2015. *Intelligibility Assessment and Speech Recognizer Word Accuracy Rate Prediction for Dysarthric Speakers in a Factor Analysis Subspace*. ACM Transactions on Accessible Computing (TACCESS), Volume 6 Number 3. (Accepted)

S. Pan and Q. Yang. 2010. *A Survey on Transfer Learning*. IEEE Transactions on Knowledge and Data Engineering.

F. Pedregosa et al. 2011. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.

J. Quiñonero and C. Rasmussen. 2005. *A Unifying View of Sparse Approximate Gaussian Process Regression*. Journal of Machine Learning Research.

C. Rasmussen and C. Williams. 2005. *Gaussian Processes for Machine Learning*,. MIT Press.

R. Sutton and G. Barto. 1998. *Introduction to Reinforcement Learning*. MIT Press.

M. Taylor, and P. Stone. 2009. *Transfer learning for reinforcement learning domains: A survey*. The Journal of Machine Learning Research.

B. Thomson, and S. Young. 2010. *Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems*. Computer Speech and Language.

J. Williams and S. Young. 2007. *Partially observable Markov decision processes for spoken dialog systems*. Computer Speech and Language.

J. Williams. 2014. *Web-style Ranking and SLU Combination for Dialog State Tracking*. Proceedings of SIGDIAL.

S. Young, M. Gašić, B. Thomson and J. D. Williams. 2013. *POMDP-Based Statistical Spoken Dialog Systems: A Review*. Proceedings of the IEEE.

Appendix A. Temporal difference kernel

In equation 5, a linear transformation from the belief-action space to the temporal difference space is applied to the to the covariance vector $\mathbf{K}_{*,X}$ and to the covariance matrix $\mathbf{K}_{X,X}$ by multiplying them by the matrix $\mathbf{H}_t$. Deriving the term $\mathbf{H}_t\mathbf{K}_{X,X}\mathbf{H}_t^\top$ we obtain the matrix in eq. 15 (page bottom), where $k_{i,j}$ is the kernel function between two belief-action points $\mathbf{x}_i = (\mathbf{b}_i, a_i)$ and $\mathbf{x}_j = (\mathbf{b}_j, a_j)$, defined in eq. 4. The transformed matrix (eq. 15) has the form of a covariance matrix where each element is a sum of kernel functions $k_{i,j}$ between belief-action points on time i or $i+1$ weighted by the discount factors. So each element of this matrix can be defined as a function of 2 temporal differences between belief-action points (TD points), $\mathbf{z}_i = (\mathbf{b}_i, a_i, \mathbf{b}_{i+1}, a_{i+1})$ and $\mathbf{z}_j = (\mathbf{b}_j, a_j, \mathbf{b}_{j+1}, a_{j+1})$ in the form of (eq. 6):

$$k^{td}_{i,j} = (k_{i,j} + \gamma_i\gamma_j k_{i+1,j+1} - \gamma_i k_{i+1,j} - \gamma_j k_{i,j+1}) \tag{16}$$

where γ_i and γ_j will be 0 if a_i and a_j are terminal actions respectively. Deriving the term $\mathbf{K}_{*,X}\mathbf{H}_t^\top$ (and $\mathbf{H}_t\mathbf{K}_{X,*}$) we obtain:

$$\mathbf{K}_{*,X}\mathbf{H}_t^\top = \begin{bmatrix} (k_{1,*} & (k_{2,*} & \cdots & (k_{t-1,*} \\ -\gamma_1 k_{2,*}) & -\gamma_2 k_{3,*}) & & -\gamma_{t-1}k_{t,*}) \end{bmatrix} \tag{17}$$

which is a vector with $k^{td}_{i,*} = (k_{i,*} - \gamma_i k_{i+1,*})$ for each term. This is equivalent to equation 16 if the action of the new point a_* is considered a terminal action, thus $\gamma_* = 0$. Then, redefining the set of belief-action points $\mathbf{X}_t$ as the set of belief-action temporal difference points denoted as $\mathbf{Z}_t$, and defining $\mathbf{K}^{td}$ as the covariance matrix computed with the kernel function between two temporal difference points (eq. 6), eq. 7 can be derived from eq. 5 by doing the following substitutions: $\mathbf{K}_{*,X}\mathbf{H}_t^\top = \mathbf{K}^{td}_{*,Z}$, $\mathbf{H}_t\mathbf{K}_{X,*} = \mathbf{K}^{td}_{Z,*}$ and $\mathbf{H}_t\mathbf{K}_{X,X}\mathbf{H}_t^\top = \mathbf{K}^{td}_{Z,Z}$.

Appendix B. Example homeService dialogues

For a more detailed description of the hierarchical structure of the *homeService* environment, this appendix presents two example dialogues between an user and the system. The second column represents the actions taken either by the user (commands) or by the system (actions)

Dialogue 1: Goal = {*TV, Channel, One*}
Dialogue starts in node "Devices"

Sub-dialogue "Devices"

User	*TV* (Speaks the command "TV")
System	*Ask* (Requests to repeat last command)
User	*TV* (Repeats his last command)
System	*TV* (Dialogue transitions to node "TV")

Sub-dialogue "TV"

User	*Chan.* (Command "Channel")
System	*Chan.* (Transitions to node "Channel")

Sub-dialogue "Channel"

User	*One* (Command "One")
System	*One* (Performs action *TV-Channel-One*)

As an action has been taken in a terminal node, the dialogue ends.

Dialogue 2: Goal = {*Hi-fi, On*}
Dialogue starts in node "Devices"

Sub-dialogue "Devices"

User	*Hi-fi* (Command "Hi-fi")
System	*Light* (transitions to node Light)

Sub-dialogue "Light"

User	*Back* (Requests to go to previous node)
System	*Back* (transitions to node Devices)

Sub-dialogue "Devices"

User	*Hi-fi* (Command "Hi-fi")
System	*Hi-fi* (transitions to node Hi-fi)

Sub-dialogue "Hi-fi"

User	*On* (Command "On")
System	*Off* (Performs action *Hifi-Off*)

As the action taken in the terminal node does not match the goal, it is a failed dialogue.

$$\mathbf{H}_t\mathbf{K}_{X,X}\mathbf{H}_t^\top = \begin{bmatrix} (k_{1,1} + \gamma_1^2 k_{2,2} & (k_{1,2} + \gamma_1\gamma_2 k_{2,3} & \cdots & (k_{1,t_{-1}} + \gamma_1\gamma_{t-1}k_{2,t} \\ -2\gamma_1 k_{1,2}) & -\gamma_2 k_{2,2} - \gamma_1 k_{1,3}) & & -\gamma_{t-1}k_{2,t_{-1}} - \gamma_1 k_{1,t}) \\ (k_{1,2} + \gamma_1\gamma_2 k_{2,3} & (k_{2,2} + \gamma_2^2 k_{3,3} & \cdots & (k_{2,t_{-1}} + \gamma_2\gamma_{t-1}k_{3,t} \\ -\gamma_2 k_{2,2} - \gamma_1 k_{1,3}) & -2\gamma_2 k_{2,3}) & & -\gamma_{t-1}k_{3,t_{-1}} - \gamma_2 k_{2,t}) \\ \vdots & \vdots & \ddots & \vdots \\ (k_{1,t_{-1}} + \gamma_1\gamma_{t-1}k_{2,t} & (k_{2,t_{-1}} + \gamma_2\gamma_{t-1}k_{3,t} & \cdots & (k_{t_{-1},t_{-1}} + \gamma_{t-1}^2 k_{t,t} \\ -\gamma_{t-1}k_{2,t_{-1}} - \gamma_1 k_{1,t}) & -\gamma_{t-1}k_{3,t_{-1}} - \gamma_2 k_{2,t}) & & -2\gamma_{t-1}k_{t_{-1},t}) \end{bmatrix} \tag{15}$$