# On Ambiguity in Robot Learning from Demonstration

Suna BENSCH [a], Thomas HELLSTRÖM [a,1]
[a] *Department of Computing Science, Umeå University, Sweden*

**Abstract.** An overlooked problem in Learning From Demonstration is the ambiguity that arises, for instance, when the robot is equipped with more sensors than necessary for a certain task. Simply trying to repeat all aspects of a demonstration is seldom what the human teacher wants, and without additional information, it is hard for the robot to know which features are relevant and which should be ignored. This means that a single demonstration maps to several different behaviours the teacher might have intended. This one-to-many (or many-to-many) mapping from a demonstration (or several demonstrations) into possible intended behaviours is the ambiguity that is the topic of this paper. Ambiguity is defined as the size of the current hypothesis space. We investigate the nature of the ambiguity for different kinds of hypothesis spaces and how it is reduced by a new concept learning algorithm.

**Keywords.** Ambiguity, Learning from Demonstration, Robotics, Predicting Human Intention

## Introduction

*Learning from Demonstration* (LfD) is a well established robot learning technique (see for instance [1] for an excellent overview). Within LfD a human teacher provides demonstrations which reflect the behaviour necessary for the robot to accomplish a specific task. The robot observes and identifies the demonstrations and is then supposed to learn such that it can repeat the behaviour in new, similar but not necessarily identical, situations. In order to simplify LfD, the robot often has a number of pre-programmed parameterised behaviour primitives [2,3,4,5]. An example of a high-level behaviour primitive is a robot's ability to grip an object, where the parameters to the behaviour primitive *grip* specify the object to be gripped. A primitive like *grip* would contain programme code that enables the robot to locate an object, navigate towards it and grip it. A major part of LfD is the identification of a primitive and its associated parameter values, given one or several demonstrations. This process is often denoted as *behaviour recognition* and a number of different techniques may be used (see for instance [6,4,7,8]). In what follows we assume that behaviour recognition is satisfactorily accomplished. Our interest lies in the connection between the demonstrations and the teacher's intention. The problem is illustrated by the following simplified scenario in which a teacher demonstrates a wanted robot behaviour by remote-controlling the robot. The robot is placed in a room with two

[1]Corresponding Author: Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden; E-mail: thomash@cs.umu.se

types of objects: *balls* and *cubes*. With its camera the robot can determine the relative location and type of objects and it can distinguish between the three object colours *red*, *blue*, and *green*. The teacher's aim is to teach the robot to collect the balls in the room and shows this by a demonstration. The teacher remote-controls the robot towards a green ball and grips it. How should the robot behave in order to repeat the demonstrated behaviour? By gripping a green ball (and ignoring the blue and red balls)? By gripping a ball of any colour? By gripping a green object of any type? By gripping a ball or a cube of any colour? Without any type of bias or additional information there is no reason for the robot to prefer any of these alternatives for the others. Thus, a single demonstration maps to several different behaviours the teacher might have intended. This one-to-many (or many-to-many) mapping from a demonstration (or several demonstrations) into possibly intended behaviours is the ambiguity that is the topic of this paper. It is quite evident that ambiguity can prevent a robot from performing its task in a satisfactory way. We investigate the nature of the ambiguity arising from the one-to-many mapping illustrated above and how it is reduced in the learning process.

The paper is organised as follows. In Section 1 we go through related research and different notions of ambiguity. In Section 2 we define ambiguity as investigated in this paper, formulate restricted hypothesis spaces and present a new concept learning algorithm. In Section 3 we analyse how ambiguity for different kinds of hypothesis spaces is reduced during the learning process. Section 4 gives a conclusion of the investigations in this paper and future research tasks.

## 1. Related research

In the literature the problem of ambiguity in LfD is acknowledged but to the best of our knowledge there has been no explicit investigation thereof. Most often, sensors and perception are tailored to specific tasks, and ambiguity is therefore most often not a real issue. In the example given in the introduction, the colour sensor is irrelevant for the intended task of collecting balls of any colour, and removing the colour sensor would indeed make the discussion about ambiguity unnecessary. The robot may successfully repeat the intended behaviour by considering *all* perceived aspects (i.e. the *type* percept) of the demonstration. However, a robot capable of learning a large number of different tasks has to be equipped with a large number of sensors and perception abilities. Simply copying as many aspects of a demonstration as possible, is normally *not* what the human teacher wants the robot to do.

It is important to distinguish the used meaning of the word ambiguity from other uses in robot learning. Ambiguity is sometimes used to denote the problems that appear due to insufficient sensing or perception. Bad colour perception may for instance result in a one-to-many mapping from demonstrations to intended behaviours. A similar ambiguity may appear due to differences in teacher and robot perspectives during demonstrations (see [12]). For example, a visual occlusion could block the teacher's view of a shared workspace such that several demonstrations, different from the teacher's point of view, look identical from the robot's point of view. The term ambiguity is also used to describe the phenomenon that one natural language sentence can have several meanings, which can be problematic also in verbal human robot interaction. Another source of uncertainty which is sometimes denoted ambiguity (e.g. in [13]) is caused by inconsistencies between several demonstrations. One common approach to deal with such ambiguity is to provide several demonstrations and let the robot deduce the common denominators such that the ambiguity is reduced or eliminated. However, the ambiguity we deal with in this paper would not be solved by improved perception or perfectly consistent demonstrations. Not even a human being with superior perception can for certain determine the intention of a teacher who grips a green ball. The intention is simply not uniquely described by the demonstration.

## 2. Concept Learning for LfD

Concept learning is a machine learning technique in which the definition of a concept is acquired through positive and negative training examples of that concept (see [9]). Although concept learning is not commonly used in practical machine learning, it provides insight into the characteristics of hypothesis selection and is useful for our analysis of ambiguity. The learning process is formulated as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best matches the training examples. We adopt this problem formulation to LfD with the teacher's demonstrations taking the role as positive training examples and all the possible intended behaviours taking the role as all potential hypotheses. In the following we first give some preliminaries and define the necessary notions and later give an illustrative example.

Let $A$ and $B$ be two sets. The inclusion of $A$ in $B$ is denoted by $A \subseteq B$, while the strict inclusion is denoted by $A \subset B$. The empty set is denoted by $\emptyset$. By $A \setminus B$ we denote the set difference of $A$ and $B$. By $2^A$ we denote the power set of $A$, that is, the set of all subsets of $A$. The Cartesian product of a finite family of sets $A_1, A_2, \ldots, A_n$, denoted by $A_1 \times A_2 \times \ldots \times A_n$ is defined as $\{ (a_1, a_2, \ldots, a_n) \mid a_i \in A_i, 1 \leq i \leq n \}$. A singleton is a set $A$ with a single element; if $A = \{a\}$ we simply write $a$. The cardinality of a set $A$ is denoted by $|A|$.

Let $V_1, \ldots, V_k$ be a family of finite value sets. A value set $V$ may contain colour values, such as *red, blue, green*, for example.

**Definition 1.** For a given family of finite value sets $V_1, \ldots, V_k$, an *instance* is a $k$-tuple $(a_1, \ldots, a_k)$, where each $a_i$ is a singleton from $V_i$, $1 \leq i \leq k$.

That is, an instance is a tuple of values from the corresponding value sets. In LfD, the parameter values to a behaviour primitive can be represented as an instance. Consequently, a demonstration is represented by an instance. The name of the behaviour primitive may be viewed as an extra parameter, but is in this paper for clarity reasons assumed to be already identified.

**Definition 2.** For a family of finite value sets $V_1, \ldots, V_k$, the *instance space* $I$ is the Cartesian product of all value sets, that is, $I = V_1 \times V_2 \times \ldots \times V_k$.

That is, the instance space $I$ is the set of all instances over the value sets $V_1, \ldots, V_k$. The size of $I$ is

$$|I| = \prod_{i=1}^{k} |V_i|.$$

**Definition 3.** For a given instance space $I$, a *hypothesis* $h$ is a subset of the instance space, that is, $h \subseteq I$.

A hypothesis corresponds to a possible intention of the teacher. The *hypothesis space* is the set of all possible hypotheses.

**Definition 4.** For a given instance space $I$, the *unrestricted hypothesis space* is given by $\hat{H} = 2^I \setminus \emptyset$.

That is, the unrestricted hypothesis space comprises all possible non-empty hypotheses. The size of $\hat{H}$ is

$$|\hat{H}| = 2^{|I|} - 1.$$

Different kinds of hypothesis spaces can be defined for certain applications. In what follows we simply write *hypothesis space* $H$ if the considered hypothesis space can be of any type.

In terms of concept learning, the purpose of LfD is to learn a concept $C$ which is a subset of the instance space $I$, that is, $C \subseteq I$, and which matches the teacher's intention as shown by the demonstrations. Given a set of training examples of $C$, the problem faced by the robot is to hypothesise $C$. The concept to be learned can also be seen as a Boolean valued function defined over the instance space: $f_c : I \rightarrow \{0, 1\}$, where $f_c(\iota) = 1$ for all instances $\iota \in I$ that belong to the concept and $f_c(\iota) = 0$ for all other instances. A concept learning algorithm finds a hypothesis function $f_h : I \rightarrow \{0, 1\}$ by searching through the hypothesis space $H$. For a successfully learned hypothesis, we have $f_h(\iota) = f_c(\iota)$ for all $\iota$ in $I$. That is, a successfully learned hypothesis comprises all instances $\iota$ for which $f_c(\iota) = 1$.

Let us consider the example given in the introduction again. The two parameters to the behaviour primitive *grip*, $a_1$ (representing the type of the object) and $a_2$ (representing the colour of the object) take values from the value sets $V_1 = \{cube, ball\}$ and $V_2 = \{red, blue, green\}$, respectively. The teacher's demonstration of gripping a green ball is represented by the instance $d = (ball, green)$. The instance space is $I = \{(cube, red), (cube, blue), (cube, green), (ball, red), (ball, blue), (ball, green)\}$. The intention of the teacher "grip a ball of any colour" corresponds to the subset $C = \{(ball, red), (ball, blue), (ball, green)\}$, which is the concept to be learned. When any of the instances in $C$ are parameters to the *grip* behaviour primitive, the robot will "repeat" the demonstrated behaviour.

Figure 1 depicts the instance space $I$ of our example, consisting of six instances. Furthermore, four hypotheses $h_1, h_2, h_3$ and $h_4$ are illustrated (there are in total 63 hypotheses in the unrestricted hypothesis space $\hat{H}$), where each hypothesis corresponds to a possible intention of the teacher. Hypothesis $h_1$ corresponds to the intention "grip a green ball", $h_2$ corresponds to "grip a ball of any colour", $h_3$ corresponds to "grip a green object", and $h_4$ corresponds to "grip an object of any type and colour". Hypothesis $h_2$ represents the behaviour to be learned in this example. The teacher's demonstration $d = (ball, green)$ is an element of all four hypotheses $h_1, h_2, h_3$ and $h_4$. That is, the single demonstration $d$ is ambiguous in the sense that it maps one-to-many to $\hat{H}$. The robot cannot determine one unique hypothesis but is left with several possible hypotheses.

We define ambiguity as the size of a given hypothesis space.

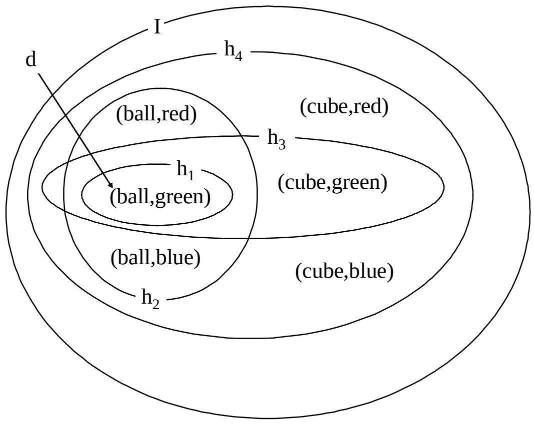


**Figure 1.** Relationships between the instance space $I$ and the hypotheses $h_1, h_2, h_3, h_4$ given the demonstration $d$ of gripping a green ball.

**Definition 5.** For a given hypothesis space $H$, *ambiguity* $A$ of $H$ is given by the size of $H$, that is, $A = |H|$.

Without any demonstrations the hypotheses in a hypothesis space $H$ represent all possibly intended behaviours of a teacher, that is, the initial hypothesis space $H$ is maximal ambiguous. In Section 3 we analyse the ambiguity during the learning process as demonstrations are given and the hypothesis space successively shrinks. In the example above, the ambiguity of the initial unrestricted hypothesis space $\hat{H}$ computes to a modest $2^6 = 64$. However, $|\hat{H}|$ suffers from combinatorial explosion when the number of parameters to a behaviour primitive increases and/or the number of the elements in a value set increases. By simply adding a third parameter $a_3$ with 4 possible values from a value set $V_4$, the size of the initial unrestricted hypothesis space $|\hat{H}|$ increases to $2^{24} - 1 = 16777215$.

### *2.1. Restricted Hypothesis Spaces*

In this subsection we define restricted hypothesis spaces that are not the powerset of a given instance set. This restricts the size of the hypothesis space and thus ambiguity.

**Definition 6.** For given value sets $V_1, \ldots, V_k$, the *cart sets* $C_1, \ldots, C_k$ are given by $C_i \subseteq 2^{V_i} \setminus \emptyset$, for $1 \leq i \leq k$.

**Definition 7.** For given cart sets $C_1, \ldots, C_k$, the *restricted hypothesis space* $H$ is the Cartesian product of all cart sets, that is, $H = C_1 \times C_2 \times \ldots \times C_k$.

Any hypothesis space $H$ that is the Cartesian product of some cart sets is referred to as *Cartesian hypothesis space* $H$. The size of a Cartesian hypothesis space $H$ is

$$|H| = \prod_{i=1}^{k} |C_i|.$$

Let us consider again our example in the introduction with value sets $V_1 = \{cube, ball\}$ and $V_2 = \{red, blue, green\}$ (in the following $red, blue, green$ are abbreviated as $r, b, g$, respectively). Let the cart sets be $C_1 = \{cube, ball, \{cube, ball\}\}$ and $C_2 = \{r, b, g, \{r, b\}, \{r, g\}, \{b, g\}, \{r, b, g\}\}$. The restricted hypothesis space is $H^{'} = \{(cube, r), (cube, b), (cube, g), (cube, \{r, b\}), (cube, \{r, g\}), (cube, \{b, g\}), (cube, \{r, b, g\}), (ball, r), (ball, b), (ball, g), (ball, \{r, b\}), (ball, \{r, g\}), (ball, \{b, g\}), (ball, \{r, b, g\}), (\{cube, ball\}, r), (\{cube, ball\}, b), (\{cube, ball\}, g), (\{cube, ball\}, \{r, b\}), (\{cube, ball\}, \{r, g\}), (\{cube, ball\}, \{b, g\}), (\{cube, ball\}, \{r, b, g\})\}$.

As can be seen, the hypotheses in $H^{'}$ may contain sets of values. This should be seen as a shorthand notation for all possible instances that can be formed by combinations of values in a hypothesis. For example, hypothesis $(cube, \{red, blue\})$ is a shorthand for $\{(cube, red), (cube, blue)\}$. Just as in the case with the unrestricted hypothesis space $\hat{H}$ (see Definition 4), applying this hypothesis to the *grip* primitive makes the robot look for and grip a red or blue cube, whichever it finds first. The hypotheses in the restricted hypothesis space $H^{'}$ may represent intentions like "grip a ball of any colour" (i.e. $(ball, \{red, blue, green\})$), "grip a green object" (i.e. $(\{cube, ball\}, green)$), and "grip a blue or green ball" (i.e. $(ball, \{blue, green\})$). Note that some intentions that can be expressed in the unrestricted hypothesis space $\hat{H}$ cannot be expressed in the restricted hypothesis space $H'$, e.g. "grip a blue cube or a red ball".

The author in [9] deals with less detailed hypotheses, as he uses the short notation $?$ for a set containing all elements in a corresponding value set. For example, the hypothesis $(?, green)$ is a shorthand for $\{(cube, green), (ball, green)\}$. Using the $?$ notation gives us an even more restricted hypothesis space. Applying the cart sets in Definition 6 to the approach in [9] leads to the following cart sets $C_1 = \{cube, ball, ?\}$ and $C_2 = \{red, blue, green, ?\}$ and the associated Cartesian hypothesis space is $H^{''} = \{(cube, red), (cube, blue), (cube, green), (cube, ?), (ball, red), (ball, blue), (ball, green), (ball, ?), (?, red), (?, blue), (?, green), (?, ?)\}$.

With $H^{''}$ we may express intentions like "grip a ball of any colour" (i.e. $(ball, ?)$), and "grip a green object" (i.e. $(?, green)$). The intentions "grip a blue or green ball" and "grip a blue cube or a red ball" cannot be expressed with hypothesis space $H^{''}$. These limitations are the price we have to pay for reducing the size of the hypothesis space and thereby making the learning problem easier.

### *2.2. A Concept Learning Algorithm*

A general algorithm for concept learning is the *candidate elimination* algorithm (see [9]). While being applicable to our LfD scenario, we propose in the following a new algorithm tailored to our focus on ambiguity. In the algorithm, the demonstrations serve as positive examples of the concept to be learned (the intended behaviour) and are used to successively shrink the hypothesis space. By $H_j$ we denote a hypothesis space $H$ at time $j$, for $j \geq 0$. The initial hypothesis space $H_0$ is the set of all possible instances (e.g. $\hat{H}, H'$ or $H''$ as defined above). At each time step $j$ a demonstration is given and a new (smaller) hypothesis space $H_{j+1}$ is constructed.

**Definition 8.** Let the current hypothesis space be $H_j$ and let $(d_1, ..., d_n)$ be the demonstration at time step $j$, $j \geq 0$. The *successor hypothesis space* $H_{j+1}$ is given by $H_{j+1} = H_j \setminus \mathcal{M}$, where $\mathcal{M} = \{(e_1, ..., e_k) \in H_j \mid \text{ there exists } d_i \notin e_i, 1 \leq i \leq k\}$.

I.e., $H_{j+1}$ is $H_j$ minus all hypotheses for which at least one parameter does not contain the corresponding parameter in the demonstration. Definition 8 is easily turned into a straightforward algorithm.

As we have seen, the hypothesis spaces often are enormous, and explicit computation of the (successor) hypothesis spaces is seldom a feasible alternative. The candidate elimination algorithm circumvents explicit computation by relying on a generality-ordering of hypotheses. If such an ordering exists, it is sufficient to update two boundary sets for the hypothesis space. We give an alternative algorithm that uses *(successor) cart sets* to implicitly define the successor hypothesis spaces at each time step. The primary advantage compared to the candidate elimination algorithm is that the size of the hypothesis space at each time step is easily computed. This is of special interest since we define the ambiguity as the size of the current hypothesis space (see Definition 5).

**Definition 9.** Let $C_1, \ldots, C_k$ be cart sets (see Definition 6). The cart sets at time 0 are denoted by $C_{1,0}, \ldots, C_{k,0}$, where $C_{i,0} = C_i$, $1 \leq i \leq k$.

For a given demonstration $(d_1, \ldots, d_k)$ at time $j$, $j \geq 0$, the *successor cart sets* $C_{i,j+1}$, $1 \leq i \leq k$ are given by $C_{i,j+1} = \{e \in C_{i,j} \mid d_i \in e, 1 \leq i \leq k, j \geq 0\}$.

Theorem 1 states that we can compute the (successor) hypothesis spaces from the given (successor) cart sets.

**Theorem 1.** *a) For given cart sets $C_{i,0}$, $1 \leq i \leq k$, the Cartesian hypothesis space $H_0$ is given by $H_0 = C_{1,0} \times \ldots \times C_{k,0}$.*

*b) For given cart sets $C_{i,j+1}$, $1 \leq i \leq k, j \geq 0$, the Cartesian hypothesis space $H_{j+1}$ is given by $H_{j+1} = C_{1,j+1} \times C_{2,j+1} \times \ldots \times C_{n,j+1}$.*

*Proof.* Part a) of the theorem follows from Definition 7 and 9 ($C_{i,0} = C_i, 1 \leq i \leq k$). For part b) let $(d_1, \ldots, d_n)$ be a demonstration at time $j$. Definition 7 of a Cartesian hypothesis space can be rewritten as $H_j = \{(e_1, ..., e_n) | e_i \in W_{i,j}, 1 \leq i \leq k, j \geq 0\}$.

Definition 8 of the successor hypotheses space can be rewritten as $H_{j+1} = \{(e_1, ..., e_n) \in H_j \mid d_i \in e_i, 1 \leq i \leq k\}$. This can be combined to $H_{j+1} = \{(e_1, ..., e_n) \in \{(e_1, ..., e_n) | e_i \in W_{i,j}, 1 \leq i \leq k, j \geq 0\} \mid d_i \in e_i, 1 \leq i \leq k\} =$
$\{(e_1, ..., e_n) \mid e_i \in W_{i,j} \text{ and } d_i \in e_i, 1 \leq i \leq k, j \geq 0\} =$ (insert Definition 9)
$\{(e_1, ..., e_n) \mid e_i \in W_{i,j+1}, 1 \leq i \leq k, j \geq 0\} = W_{1,j+1} \times W_{2,j+1} \times \ldots \times W_{n,j+1}$. □

Theorem 1 can be used to construct a simple algorithm that, given a current hypothesis space and a demonstration, generates an updated hypothesis space. The algorithm can be applied repeatedly for several demonstrations. Given a Cartesian hypothesis space $H_j$ and a demonstration at time $j$, Algorithm 1 computes $H_{j+1}$.

*Algorithm 1*

Input: $H_j$ given by the cart sets $C_{i,j}, 1 \leq i \leq n$ and a demonstration $(d_1, \ldots, d_n)$
Output: $H_{j+1}$
Method: $C_{i,j+1} = \{e \in C_{i,j} \mid d_i \in e\}, 1 \leq i \leq n$.
$H_{j+1} = C_{1,j+1} \times C_{2,j+1} \times \ldots \times C_{n,j+1}$.

## 3. Reduction of Ambiguity for Different Hypothesis Spaces

In this section we illustrate how an LfD process, performed by Algorithm 1, gradually reduces ambiguity as new demonstrations are presented to the robot. We will compare the reduction of ambiguity for four different hypothesis spaces: the unrestricted hypothesis space $\hat{H}$ (see Definition 4), the two Cartesian hypothesis spaces $H^{'}$ and $H^{''}$ (see Definition 7 and Subsection 2.1) and another Cartesian hypothesis space $H^{'''}$ whose cart sets are identical to the value sets, i.e. $C = V$. The assumed learning scenario contains four value sets: $V_1 = \{cube, ball\}, V_2 = \{red, green, blue\}, V_3 = \{s, m, l, xl\}, V_4 = \{a, b, c, d, e, f\}$. Instance space $I$ is the Cartesian product of these four value sets and has 144 elements, for example $(ball, green, s, a)$. Hypothesis space $\hat{H}$ is given by $2^I \setminus \emptyset$. One of the $2^{144} - 1 \sim 10^{43}$ elements of $\hat{H}$ is $\{(ball, green, s, a), (cube, green, s, b), (cube, blue, s, b)\}$. For $H^{'}$, we have $C_1 = \{cube, ball, \{cube, ball\}\}, C_2 = \{red, green, \{red, green\}, blue, \{red, blue\}, \{green, blue\}, \{red, green, blue\}\}$, $C_3 = \{s, m, \{s, m\}, l, \{s, l\}, \{m, l\}, \{s, m, l\}, xl, \{s, xl\}, \{m, xl\}, \{s, m, xl\}, \{l, xl\}, \{s, l, xl\}, \{m, l, xl\}, \{s, m, l, xl\}\}$. $C_4$ comprises all 63 combinations of values from $V_4$. $H^{'}$ is defined as the Cartesian product of these four cart sets. One of the 19845 elements of $H^{'}$ is $\{cube, \{red, blue\}, xl, b\}$. For $H^{''}$, we have $C_1 = \{cube, ball, \{cube, ball\}\}$, $C_2 = \{red, green, blue, \{red, green, blue\}\}$, $C_3 = \{s, m, l, xl, \{s, m, l, xl\}\}$, $C_4 = \{a, b, c, d, e, f, \{a, b, c, d, e, f\}\}$. To simplify notation, the last element of each cart set (the "wild card") is denoted by $?$. $H^{''}$ is the Cartesian product of the four cart sets. One of the 420 elements of $H^{''}$ is $(ball, ?, xl, a)$. For $H^{'''}$, we have $C_1 = \{cube, ball\}$, $C_2 = \{red, green, blue\}$, $C_3 = \{s, m, l, xl\}$, $C_4 = \{a, b, c, d, e, f\}$. One of the 144 elements of $H^{'''}$ is $(ball, red, xl, c)$.

For each presented demonstration, Algorithm 1 is applied and the size of the hypothesis space is reduced. The nature of this reduction depends on the type of the hypothesis space. In Figure 2, the result of learning with 12 fixed demonstrations is shown. For $\hat{H}$, each demonstration reduces $|\hat{H}_j|$, $j \geq 0$, almost exactly by a factor 2 (a discrepancy obviously occurs when $|\hat{H}_j|$ is an odd number). Any, not already rejected, element of the instance space $I$ is at any moment element of half of the total number of hypotheses in the current hypothesis space. A demonstration will therefore always result in a rejection of half of the hypotheses. With a logarithmic scale, this shows up as a straight line in the graph in Figure 2 (to increase readability, 100 is subtracted from the values plotted for $\hat{H}$). The hypothesis spaces $H^{'}$, $H^{''}$ and $H^{'''}$ are altogether denoted in the following by $H^*$. The reduction of $|H^*_j|$ is more complex. The initial $|H^*_0|$ (shown to the far left in the diagram for *Demonstration* #=0) is smaller then for $\hat{H}_0$ which is obvious since $H^*_0$ are all strict subsets of $\hat{H}_0$. Furthermore, the reduction is better than linear, which is related to the generalisation ability that comes with restricted hypothesis spaces. For $\hat{H}$ there is no generalisation such that a demonstration does not influence the preference of instances that are similar to the current demonstration. This corresponds to rote learning (see [10]). For the restricted hypothesis spaces $H^*_j$, $j \geq 0$, one demonstration may match different number of hypotheses and the reduction speed will vary. For $H^{'''}$, a single demonstration is sufficient to uniquely identify one hypothesis, which is identical to the demonstration. The size of the hypothesis space consequently drops to 1 already after one demonstration.

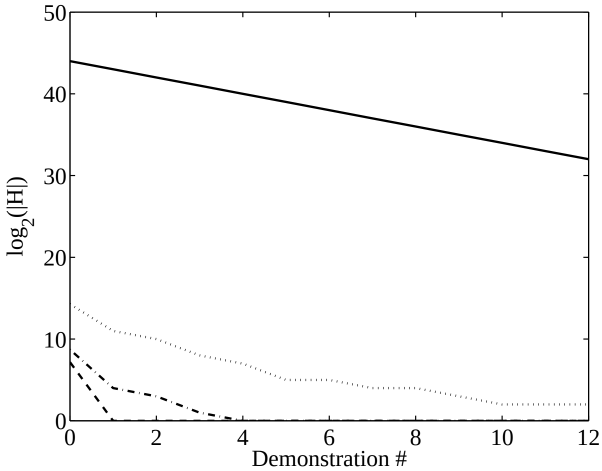


**Figure 2.** Size of the hypothesis space as 12 demonstration successively are presented to the robot. From top to bottom, $\hat{H}$, $H'$, $H^{''}$ and $H^{'''}$.

## 4. Summary and Conclusions

We have given a formal description of ambiguity in LfD. Ambiguity is defined as the size of the current hypothesis space. Concept learning can be used to reduce the size, preferably down to size 1 which means that the robot has no doubts which behaviour the human teacher wants the robot to repeat. Learning algorithms, such as Algorithm 1, construct an updated hypothesis space by removing hypotheses that are inconsistent with the demonstrations. However, there are fundamental limitations for this way of reducing ambiguity with unrestricted hypothesis spaces. A demonstration can only remove hypotheses that are inconsistent with the demonstration. This leads to a smaller hypothesis space, but the remaining hypotheses are more general than the ones removed. In practice this means that it is impossible to learn to ignore a parameter or certain parameter values. Consider for example trying to teach a robot to grip a green ball, if it is capable of distinguishing between 100 different colours. There is no way to remove the 99 incorrect colours from the hypotheses in the hypothesis space by using demonstrations of the wanted behaviour only. The standard approach in machine learning is to introduce *bias* into the learning. One major type of bias is *restricted hypothesis space bias* (see [11]), such as the usage of $H^{'}$, $H^{''}$ and $H^{'''}$ in the previous section. Restricted hypothesis spaces are valuable not only because they are smaller in size and lead to faster search but they also rule out certain hypotheses already in the definition of the hypothesis space. For example, with $H^{''}$ it is only possible to express intentions to grip balls of one specific colour, e.g. $(ball, green)$, or to ignore the colour property altogether, e.g. $(ball, ?)$. This rules out complex hypothesis such as $(ball, \{red, green, blue, purple\})$ already in the definition of the hypothesis space. Restricted hypothesis spaces also introduce a dependency between instances such that the learning process will generalise data. A related type of bias is the mechanism by which hypotheses spaces are inferred from the demonstrations. The deductive concept learning algorithms may for instance be replaced by an inductive decision tree learning algorithm (see for instance [14,15]). This would make it possible to reject the 99 incorrect colours if all (or most) of the demonstrations have green as colour value. Irrelevant parameters such as size and temperature may in the same way be left outside the generated hypothesis. Another major type of bias is *preference bias*. One example is the heuristic principle Occam's razor (see [11]). It states that one should pick the simplest hypothesis if several hypotheses match the data. This would mean keeping $(ball, green)$ while removing hypotheses such as $(ball, \{red, green, blue, purple\})$ and $(ball, \{green, white, blue, brown\})$ from the hypothesis space.

Future research will investigate how prior knowledge can be used as preference bias in the LfD process. Preferences or prior probabilities for parameter types (e.g. *type* and *colour*) and values (e.g. $red$, $green$, and $blue$) can be stored in and extracted from memory structures such as semantic networks, and utilised for both reduction of the hypothesis space and as guidance when the robot tries to repeat the demonstrated behaviour. We will also extend Algorithm 1 such that human feedback to the robot's attempts to repeat a learned behaviour can be included in the learning. This makes it possible to learn to ignore a parameter or certain parameter values. Other sources of information, such as verbal commands will also be considered.

## References


[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 1371–1394. Springer. 2008.

[2] R. Amit and M. J. Matarić. Parametric primitives for motor representation and control. In *Proceedings, IEEE International Conference on Robotics and Automation (ICRA-2002), volume 1*, pages 863–868. Washington DC. 2002.

[3] J. Peters and S. Schaal. Policy Learning for Motor Skills. In M. Ishikawa, K. Doya, H. Miyamoto, and T. Yamakawa, editors, *Neural Information Processing, LNCS 4985*, pages 233–242. Springer. 2008.

[4] P. K. Pook and D. H. Ballard. Recognizing Teleoperated Manipulations. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 578–585. Atlanta, GA, USA. 1993.

[5] H. Urbanek, A. Albu-Schäffer, and P. Smagt van der. Learning from demonstration: repetitive movements for autonomous service robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3495–3500. Sendai, Japan. 2004.

[6] M. Nocolescu. A Framework for Learning from Demonstration, Generalization and Practice in Human-Robot Domains. PhD thesis. University of Southern California. 2003.

[7] D. C. Bentivegna. Learning from Observation Using Primitives. PhD thesis. College of Computing, Georgia Institute of Technology. 2004.

[8] E. A. Billing and T. Hellström. Behavior recognition for segmentation of demonstrated tasks. In *Proceedings of the IEEE International Conference on Distributed Human-Machine Systems*, 228–234. Athens, Greece. 2008.

[9] T. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[10] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell. Machine Learning: A Historical and Methodological Analysis. *AI Magazine*, 4(3):69–79, 1983.

[11] T. Dean, J. Allen and Y. Aloimonos. *Artificial Intelligence: Theory and Practice*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[12] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. L. Thomaz. Using Perspective Taking to Learn from Ambiguous Demonstrations. *Robotics and Autonomous Systems (RAS), Special Issue on the Social Mechanisms of Robot Programming by Demonstration*, 54(5):385–393, 2006.

[13] B. D. Argall, S. Chernova, M. Veloso, B. Browning. A Survey of Robot Learning from Demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[14] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986.

[15] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, Upper Saddle River, NJ, 2009.