Inter-Robot Transfer Learning for Perceptual Classification

Zsolt Kira
College of Computing
Georgia Institute of Technology
85 Fifth Street N.W.
Atlanta, GA, 30308
zkira@gatech.edu

ABSTRACT

We introduce the novel problem of inter-robot transfer learning for perceptual classification of objects, where multiple heterogeneous robots communicate and transfer learned object models consisting of a fusion of multiple object properties. Unlike traditional transfer learning, there can be severe differences in the data distributions, resulting from differences in sensing, sensory processing, or even representations, that each robot uses to learn. Furthermore, only some properties may overlap between the two robots. We show that in such cases, the abstraction of raw sensory data into an intermediate representation can be used not only to aid learning, but also the transfer of knowledge. Further, we utilize statistical metrics, learned during an interactive process where the robots jointly explore the environment, to determine which underlying properties are shared between the robots. We demonstrate results in a visual classification task where objects are represented via a combination of properties derived from different modalities: color, texture, shape, and size. Using our methods, two heterogeneous robots utilizing different sensors and representations are able to successfully transfer support vector machine (SVM) classifiers among each other, resulting in speedups during learning.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics; I.2.10 [**Vision and Scene Understanding**]: Perceptual Reasoning; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms

Keywords

Inter-Robot Transfer, Transfer Learning, Multi-Robot Systems

1. INTRODUCTION

As autonomous robots become increasingly common, it is likely that there will be multiple robots that each learn through experience; that is, via embodied interaction with the world. This type of grounded learning, however, ignores social aspects of learning. With multiple robots, it is crucial for the robots to be able to share knowledge either through explicit communication or implicit means such as imitation. Such knowledge sharing can speed up learning significantly and can reduce the need for costly human teaching.

Several problems can prohibit effective sharing of knowledge, however. Knowledge learned via exploration of the world is often embodiment-specific, that is unique to the particular sensing capabilities of the robot. It is quite common to have some degree of heterogeneity among robots, however, and there can be slight perceptual differences even among two robots of the same model. For example, the camera color models may differ slightly. It is an even greater problem when different types of robots are used. Currently, there is a plethora of robotic systems in use in home environments (e.g. the Roomba and lawn mowing robots), research labs, and in various domains where task allocation to differing robots is necessary.

In this paper, we look at this issue from a transfer learning perspective. Transfer learning is a method that has been recently applied in the computer vision, machine learning, and reinforcement learning fields. From a machine learning perspective, transfer learning is a subfield that attempts to speed up learning by transferring previously learned knowledge to new domains or categories (e.g. [13], see [12] for a survey), where the distributions of the features differ in the new domain or category. Unlike these fields, however, little attention has been paid in robotics to the transfer of learned representations across multiple robots.

In order to apply these techniques to robots, there are some unique challenges that must be addressed as well as unique opportunities that can be leveraged. For example, there can be severe differences in the distributions of data sensed by heterogeneous robots, resulting from differences in sensing, sensory processing, or even representations, that each robot uses to learn. Furthermore, there may be missing properties with only some of them overlapping in the two robots. Despite these challenges, the fact that robots are embodied and can explore their environment jointly presents opportunities for finding properties that robots have in common, as will be shown. We hope that this paper will raise the awareness of the advantages of inter-robot transfer learning in the robotics community, as well as spur new robotics-specific formulations of the problem that can be tackled by the machine learning and computer vision communities.

Cite as: Inter-Robot Transfer Learning for Perceptual Classification, Z. Kira, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. . 13-20

In order to deal with potential sensor and representation differences, we abstract raw sensory data into natural object properties such as color or texture categories. Each robot can then learn these categories, and we show that such abstractions aid in learning to classify objects. In addition, common properties that are shared between the robots can be found by allowing the two robots to interact jointly in an environment. The resulting property mappings can then be used to transfer learned classifiers between robots, and again we show empirically that the property abstractions aid transfer, especially when properties are grounded differently in each robot due to processing or representation differences. We demonstrate empirical evidence for our claims using two heterogeneous robots (a Mobile Robots Amigobot and Pioneer 2DX) and four types of object properties: color, texture, shape, and size. We show that classifiers for thirty four real-world objects can be successfully transferred between the robots despite their differences.

2. RELATED WORK

Transfer learning has recently flourished in multiple fields including computer vision (e.g. [3, 16]), machine learning [12], and reinforcement learning [15]. In computer vision, there has recently been a heavy focus on cross domain and cross category transfer of knowledge in the form of, for example, priors in a probabilistic framework. These transfers occur across different categories, as opposed to across sources of data in the case of robots.

Despite these advances in learning, there has been little work focusing on transfer learning for robotics. Transfer in reinforcement learning has focused on differing state and action spaces [15]. Transfer in perceptual classification and learning for robotics, on the other hand, is almost nonexistent. Some computer vision results, such as those of [11], have used robots with multiple cameras and transferred SVM classifiers as in our work. However, in that case the cameras were of an identical model, hence differing from our focus on heterogeneity. We hope that this paper results in more focus on the issue of transfer learning in the robotics community, in addition to bringing to light special requirements that arise in transfer learning for robotics so that they may be explored in machine learning and other similar fields. For example, by virtue of the fact that they are embodied robots in the real world, joint interaction in the world can be used to learn differences between robots, as will be shown in this paper.

The key issues and motivation in this paper are also related to social symbol grounding, that is finding common symbols for similar concepts across a population of agents. This is related to language formation and has been studied extensively in linguistics and evolutionary or artificial life [17, 14]. For example, work done by Luc Steels and his colleagues in the area of shared vocabulary development used shared attention to synchronize two robot's symbols [14]. This is a similar concept to ours, although they did not explicitly deal with the issue of robot heterogeneity where robots may have different feature spaces.

Another example of this in robotics includes work by Jung and Zelinsky, who studied two robots that perform the same task (vacuuming) but had different capabilities; one robot swept small pieces and reached into corners, while the other could only vacuum the larger piles and could not reach corners [7]. In that case, a shared ontology was developed by establishing a physically shared context during learning: The two robots followed each other around the room and agreed on symbols for specific locations in the environment. In a similar vein, Billard and Dautenhahn have looked at a situation involving two homogeneous robots where one teacher attempts to share its symbols with another robot via imitation, namely following [1]. In this paper, the robots can learn by themselves separately, and then utilize joint interaction only to facilitate transfer.

3. A FRAMEWORK TO FACILITATE TRANSFER OF CONCEPTS

We will now describe the proposed framework designed to facilitate both learning and transfer of learned object models between two robots. The key idea of this framework is that an intermediate representation is built from raw sensory data. The importance of abstraction into intermediate representations for learning have been recognized in computer vision as well as machine learning. In this paper, we claim and empirically confirm that this is also useful for facilitating knowledge transfer between heterogeneous robots.

The intuition is that a small number of properties can be used to represent an order-of-magnitude larger number of concepts. Since there will be only a few number of properties to be mapped between robots, compared to the number of concepts, less effort is needed for transfer to occur. Note that this abstraction also allows the robots to represent the same properties using their own sensing, and can even use different representations (for example, an HSV color space versus an RGB color space) or different modalities (for example, the width of an object can be sensed via laser or stereo camera). We will provide evidence for the claim that this aids both learning as well as transfer, especially when the underlying representations used by the robots differ, in the next section.

In order to determine how to abstract sensory data, we take inspiration from Gärdenfors' conceptual spaces [5], a cognitively-inspired multi-level representation that uses geometric spaces to represent concepts (in our case objects). The most basic primitive of the representation is a **dimension** which takes values from a specific range of possible values. For example, the hue of an object can be specified as an angle in the range [0, 1]. The values of these dimensions come from perceptual features processed from sensor data. For example, a camera sensor measures physical properties of the world (light), converting them into a digital representation consisting of multiple pixels in the form of an RGB space. A perceptual feature detector can convert regions of the image into an HSV space, and the H (hue) value can make up a dimension. The feature detector returns a set of these, one for each region of the image that it determines is salient. (Note that in this paper, we will use the term features interchangeably with dimensions.)

Gärdenfors posits that there are integral dimensions that cannot be separated in a perceptual sense. For example, the HSV color space can be argued to consist of three integral dimensions. Another example used is pitch and volume that is perceived by the auditory system. A set of such integral dimensions is referred to as a **domain**. A domain defines a space that consists of all possible values of the integral dimensions. For example, the HSV color space is a domain consisting of three dimensions. It is useful to abstract and divide the space into specific regions, which define a **prop-**

erty. These properties carve out natural regions of the domain borne out of structure in data obtained from the world. For example, "blue" can be a property that corresponds to some region of the color space. Note that different dimensions can result in similar properties of objects; for example, similar color property can be learned using both HSV and RGB domains. The regions can be arbitrarily shaped, although Gärdenfors defines what he calls natural properties consisting of regions with certain characteristics such as convexity. This property is preserved in our work and is important in order to ensure consistent distance measurements.

Now suppose that there are two robots, robot A and robot B. These robots each have a set of features (or dimensions). There may or may not be an overlap in these features, and even if there is the data resulting from the sensors may differ in distributions anyways (e.g. RGB spaces will differ due to differences in the color models of different cameras; see [8] for experiments demonstrating this.) We claim, and will show evidence, that these raw features pose difficulty when transferred across heterogeneous robots. Instead, each robot individually learns a set of properties $P^A = \{p_1^A, p_2^A, ..., p_n^A\}$ and $P^B = \{p_1^B, p_2^B, ..., p_m^B\}$. The properties can be learned separately by each robot, and as mentioned similar properties can be learned even if they differ in their underlying representations or domains. We will discuss the particular representation and learning algorithms we use in our particular implementation of the framework in the next section. Since these are natural properties of objects, such as color or texture categories, there will be some commonalities in the properties learned by the two robots. This is especially true if they are supervised by humans, as they are in this paper. The point is to leverage as much similarity as possible when transferring object models.

A particular object can have real-valued membership values in these properties, as we will later discuss. In order to learn objects or concepts, then, we can feed property memberships into a binary classifier that determines whether the object belongs to a particular class. Again, supervised learning can be used, where instances of an object are given, property memberships are extracted, and these are used to train the classifier. In order to transfer these classifiers, it becomes necessary to determine which properties on robot A map onto properties on robot B. As detailed in the subsections below, we use joint interaction where the robots achieve a shared context (that is, view a similar scene with the same objects in them). These instances, where both robots sense the same objects, are used to build statistical metrics that can be used to determine whether two properties (one from each robot) correspond to the same property in the world. Since some properties may not map, it is important that the classifier be able to handle missing attributes. It is also necessary for the classifier to support incremental learning, so that after receiving a classifier from another robot, the receiving robot can continue learning.

In summary, each robot begins by learning properties using labeled data. These properties are then combined in a classifier to learn objects, again using labeled data. After this learning period, where each robot learns individually, the robots may transfer knowledge to each other. Before transfer can occur, however, the robots jointly interact in the world to map their intermediate property representations. After this mapping period, all of the concepts on one robot can be transferred to the other. We will now detail the specific algorithms and representations that we use to learn and map properties.

3.1 Learning Abstracted Object Properties

In this paper, we represent properties as Gaussian Mixture Models (GMMs) and learn them via the Expectation Maximization algorithm, although only one Gaussian is used in this case to preserve convexity properties suggested by conceptual spaces [5]. Properties are learned using a supervised learning framework where the properties are empirically determined a-priori and hand-labeled instances are provided by picking out segments from a graph-cut based automatic segmentation algorithm [4]. Future work will look at unsupervised learning of intermediate properties.

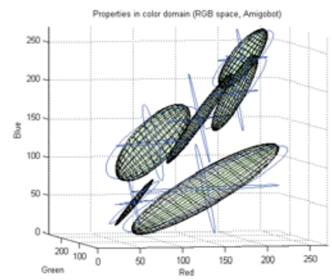


Figure 1: Six color properties, represented as GMMs, after training with multiple objects.

In order to learn the representation for object properties, we will scaffold the robot's learning by first providing it with multiple instances of data that contain a property. Each scene contains a target object and results in a set of points calculated from the output of the robot's low-level perceptual feature detectors: color, texture, size, and shape. For each property p_i, we use a Gaussian Mixture Model to characterize the regions, denoted as G_i. Specifically, each property can be modeled as:

$$P(p_i|\theta) = \sum_{j=1} w_j P(p_i|u_j, \sigma_j) \qquad (1)$$

where w_j is known as the mixing proportion and θ is a set containing all of the mixing proportions and model parameters (mean μ and standard deviation σ). An Expectation Maximization (EM) algorithm is used to determine these parameters [2]. Once models are learned, they are used to determine the membership of an instance in a property. Specifically, features (e.g. RGB values) are obtained from sensors and result in points in a given space (e.g. the RGB color space). The membership of this point in property p is the Gaussian distance function $s(i, p)$ to the nearest property cluster. This measures the likelihood of the point coming from a particular Gaussian. Fig. 1 shows the color properties in an RGB space for the Amigobot.

In order to classify objects, supervised images with object segments are provided. For each instance, property memberships are calculated and these memberships for all properties are input into a support vector machine classifier [6].

3.2 Mapping Abstract Properties

As mentioned properties are regions in domains, in our case represented as Gaussian clusters. The same property can be represented in two different robots as clusters with different characteristics (for example, different standard deviations) or even domains from different sensors (for example, the width of an object as detected by a camera or laser). Given these clusterings of a domain, the problem is to find associations between clusters from each robot. In order to do this, we use instances from each robot while viewing the same scene and compare properties that they see. In this paper, this is done manually and in a looser sense; manual selection of images is performed such that both robots see the same object, although not necessarily from the same perspective. We also gathered a smaller set of instances by teleoperating the robots to face the same object. In prior work [10], we have explored behaviors for this in simulation, such as one robot following the other and parking next to it or pointing to objects from afar. Future work will include developing a localization mechanism, which is needed to perform these behaviors on real robots. Given a scene, each robot processes its sensory data to produce a set of property memberships for all properties. For each pair of properties (one from each robot), statistics described below are maintained in order to determine whether they represent similar physical attributes.

We map individual clusters to each other by building confusion matrices between the properties (see [8] for a full description). Specifically, we utilize the confusion matrix to determine pairs of properties that may potentially represent the same physical property. The rows of the matrix correspond to the properties of one robot, while the columns correspond to the properties of the other robot. Values within the matrix represent the correlation between corresponding properties (one from each robot). Suppose that there are two clusterings G_j^A and G_k^B defining regions corresponding to properties p_j^A and p_k^B for robot A and B, respectively. Also, each clustering for robot A and B has n_j^A and n_k^B clusters, respectively. Finally, suppose that we have a set of instances I from each robot (obtained using its own sensing) with the highest property membership value corresponding to property p_j^A. Element (i, j) of the confusion matrix $PC^{A,B}$ is then set to:

$$PC_{(j,k)}^{A,B} = \sum_i^I \frac{min(s(i, p_j^A), s(i, p_k^B))}{s(i, p_j^A)} \qquad (2)$$

Here, $s(i, p)$ is the Gaussian membership function of instance i in property p. The min function is used to represent the intersection of property memberships, as is used commonly in fuzzy sets. For each property of a robot, the highest values in the corresponding property's row or column will be taken and it will be considered potentially corresponding to the respective property of the other robot. A threshold may be placed on this as well, although we do not do so in this paper.

4. ROBOT EXPERIMENTS

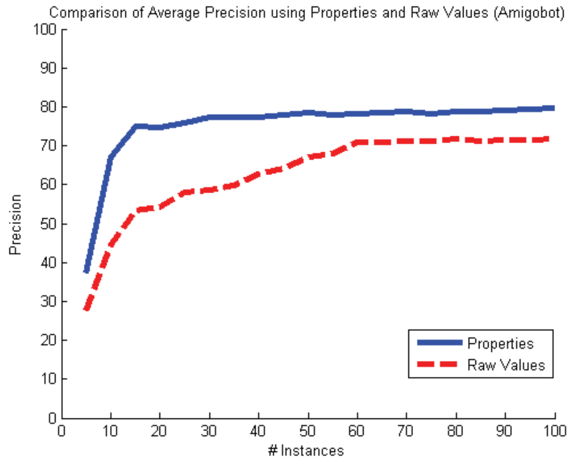
We have conducted experiments in order to verify our claims that the intermediate abstractions improve learning, that they can be mapped successfully between robots via sensing from a shared context, and that they facilitate transfer, especially in the case where the underlying representations differ in the two robots. For all experiments, two robots were used: A Mobile Robots Amigobot with a wireless camera and a Pioneer 2DX robot with a Quickcam Express web camera (shown in Figure 3). The Amigobot used 640x480 resolution images while the Pioneer used 320x240 resolution images, another source of heterogeneity. Unlike the first robot, the Pioneer robot had a SICK laser range finder as well, data from which was processed to extract object shape (curvature) and size properties.

Figure 2: Twelve of the thirty-four objects used. Objects differed in color, texture, shape, and size.

Figure 3: Heterogeneous robots used for the experiments. Left: Amigobot with a wireless camera. Right: Pioneer 2DX robot with a Quickcam web camera and a SICK laser range finder.

In order to train color, texture, shape, and size properties, the robots were driven around a laboratory environment, resulting in a large amount of stored sensor data. Thirty-four realistic objects were used, twelve of which are shown in Fig. 2. Examples of objects included whiteboards, wood crates, some robots (e.g. a different Amigobot as well as an iRobot Create), trash cans, and so on. One hundred images of each object were chosen, seventy of which were randomly chosen for training and thirty for testing (note some object classes had somewhat fewer testing instances). Anywhere from one to six objects from the environment per category were chosen for training of properties, some of which are shown in Table 1. For each property, all that is given is the domain to be trained, a set of data instances, and segments chosen from the automatic segmentation of the target object. The color space used for the properties included RGB and HSV. For texture, an empirically-chosen Gabor filter was used and the dimensions of the space consist of the mean and standard deviation of its output. Shape, obtained from range finders, consisted of a curvature metric along the object. Object sizes were obtained by simple calculating the 3D points in which they lied and measuring the three dimensional distance between the first point on the object and the last. The camera and laser sensors were calibrated so that points from the laser could be projected onto the image.

In some cases, these features (and therefore properties) were missing if, for example, the object was not in full view and the laser readings therefore did not cover the entire object. In these cases, the features and property memberships were considered missing; as mentioned, the ability to handle



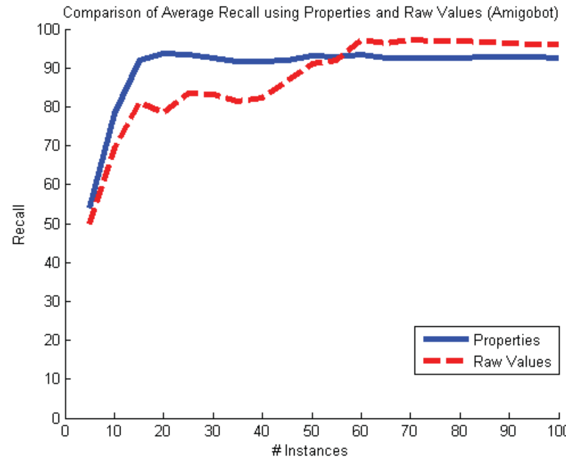


Figure 4: Results demonstrating the advantage of using abstracted properties as opposed to raw sensory data when learning. The figure on the left shows the precision, while the figure on the right shows recall.

missing features is an important capability to enable transfer learning of classifiers.

4.1 Learning Object Models

In order to learn object classes, we used a support vector machine classifier. In order to verify the claim that the abstraction of sensory data aids learning, we used two conditions. In the first condition, the property memberships for the previously learned properties were used as attributes. In the second condition, raw sensory data itself (e.g. RGB values or the curvature metric) were used as attributes for training. In later subsections, we also used a third condition where raw sensory data was used, but the robots used different representations for color, namely one robot used an RGB color space while the other used an HSV color space. In order to gauge classification rates, both recall and precision are plotted. These are standard classification metrics, where recall measures the number of true positives divided by the number of positives in the test set, while precision measures the number of true positives divided by the number of test set instances that were classified to be positive.

Figure 4 shows a comparison between the first two conditions for both recall and precision. Results are plotted for different number of training instances used. As can be seen, our hypothesis that the abstraction of properties results in higher learning curves is confirmed, showing that it is more difficult to learn with raw sensory data. When all training instances are used, recall rates are a little higher when using raw sensory data (a difference of about 3.3) but this comes at the expense of lower precision (a differences of about 7.9). Furthermore, the learning curve is significantly higher when using properties most of the time (and especially in the early stages of learning), showing that it is more difficult to learn with raw values.

4.2 Mapping Abstracted Properties

We now describe results for determining the property mappings between the robots. In order to learn the mappings, the confusion matrix was learned via all test instances of the thirty four objects. Hence, the shared context in this case was manually guaranteed (i.e. images from each robot sensing the same object were chosen). For each instance, each robot picked properties that were high for that instance, and added to the average the ratio of the other robot's property membership to its own. Figure 6 show a gray-scale representation of the learned confusion matrix, where lighter values correspond to higher values (i.e. more highly correlated properties). The diagonal represents the ground truth mappings (since we trained the properties in the same order) and are highlighted. Note that there are fewer rows than columns since four of the properties do not exist on the first robot. By taking the maximal values in each row, eight of ten properties were mapped correctly.

Note that in previous results, we have successfully mapped properties with perfect accuracy [8, 9]. In this case, the texture properties were highly correlated across all objects, meaning that the properties were not independent. This shows that such dependencies can cause errors in the property mappings, an important fact for future work when we will work on unsupervised learning of the properties since some unsupervised algorithms do not guarantee this. Figure 5 shows the number of correct mappings, averaged across ten randomized validation runs, as the number of testing instances increases. We also tested this with sixty four instances where the two robots were teleoperated to view the same object. In that case, five of ten property mappings were correct, showing similar results as the graph where images were manually chosen. The teleoperated data is shown in bold in Figure 5 to show that similar trends were achieved. This shows that, assuming the robots can localize or detect each other, these mappings could be learned autonomously once behaviors for following or pointing from our previous work performed in simulation are applied to the real robots.

4.3 Transfer Learning

We now describe results for the transfer of SVM classifiers in the three conditions. In the simple transfer case (labeled as "Transfer"), the support vectors learned by one robot are directly used to classify testing data obtained from the other robot. This highlights the strength of abstracting data into properties, as the property memberships will have similar distributions for objects unlike raw sensory data. To

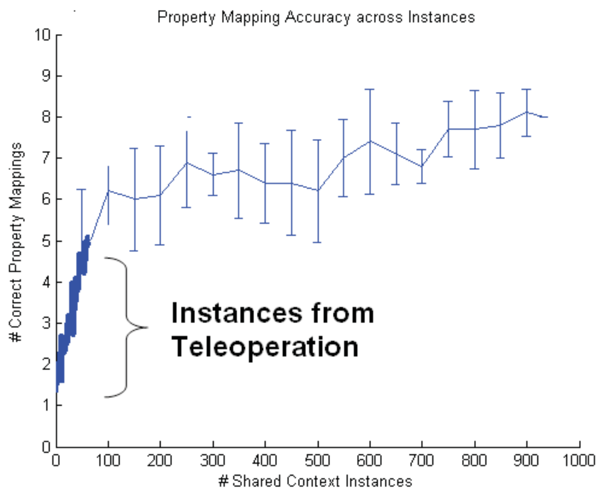


Figure 5: This graph shows the number of correctly mapped properties between the robots as the number of instances grows.

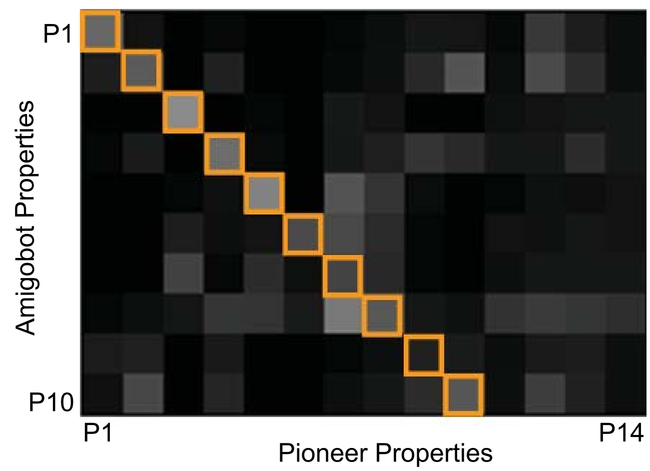


Figure 6: A gray-scale representation of the property mappings, where the rows correspond to properties of the Amigobot robot and columns correspond to properties of the Pioneer robot. Note that the latter robot has four more properties, utilizing its SICK range finder. By taking the maximal values of each row, eight of ten properties are mapped correctly (ground truth is the diagonal, highlighted).

perform continued learning after receiving classifiers from another robot, we instead use the support vectors from the transferred SVM classifier as input instances to a new classifier. This sometimes lead to an immediate slight performance change, usually a slight increase. Subsequently, additional training instances were added as input to the classifier (plotted as "Transfer + Learning"). Again, the plots show results as the number of *training* instances increases. For the "Transfer" case, the "# Instances" axis corresponds to the number of training instances used by the *sending* robot when training the classifier that is sent to the receiving robot. For the "Transfer + Learning" case, the "# Instances" axis refers to the number of training instances used by the *receiving* robot that were added to the support vectors received by the sending robot.

Figure 7 shows the recall and precision results when comparing the first two experimental conditions. The graphs compare results without transfer ("Own Learning"), transfer ("Transfer"), and continued learning by the receiving robot after transfer ("Transfer + Learning"). As can be seen, transfer learning results in the bootstrapping of both recall and especially precision. This is true for learning using properties, but is even more pronounced when learning with raw values (a more difficult task). This shows that as learning becomes more difficult, transfer learning becomes even more useful.

Table 1: Properties and example objects used to train them

Property	**Objects Incl.**
White	Whiteboard, Create Robot
Tan	Crates
Black	Chair1, Case, Trash
Gray	Fence, Cabinet
Blue	Chair2, Recycling Bin
Texture1	Chair2, Black Case
Texture2	Gray Cabinet, Fence
Texture3	Couch
Texture4	Crates
Round	Recycling Bin, Trash
Flat	Whiteboard, TV Box
Small	Bucket, Cooler
Large	Whiteboard, Crate

As can be seen from the "Transfer + Learning" curves, the Amigobot robot could achieve high recall and precision (74.0 and 73.7, respectively), even without having seen any instances by itself, compared with low rates after training by itself with only five instances (e.g. 53.8 and 37.2, respectively). As the receiving robot began to receive additional training instances, it could combine the received classifier with these instances and eventually converge towards the final asymptote. This shows that combining learned knowledge with received knowledge does not pose a problem. The same trends exist for the Pioneer robot, but are not shown here due to space limitations.

Our final results consist of the same transfer learning experiments but in the third condition, where the two robots utilized different underlying representations (HSV versus RGB) for the color properties. Figure 8 shows these results. As can be seen, transfer learning when using properties to learn continued the same trend as before, with significant gains after transfer even when the receiving robot had not seen any learning instances itself. When using raw values, however, transfer learning was completely ineffective and failed catastrophically. Even when the receiving robot continues to learn by itself (the "Transfer + Learning" curve), it fails to achieve high rates until a significant number of training instances are used. This is likely because it takes many instances to wipe out the detrimental classifier received from the other robot. In this case, when using raw sensory values, it would have been better had the Amigobot not received anything from the other robot. Transfer when using properties, however, remained as effective as before.

In this case, the results for the Pioneer robot differed

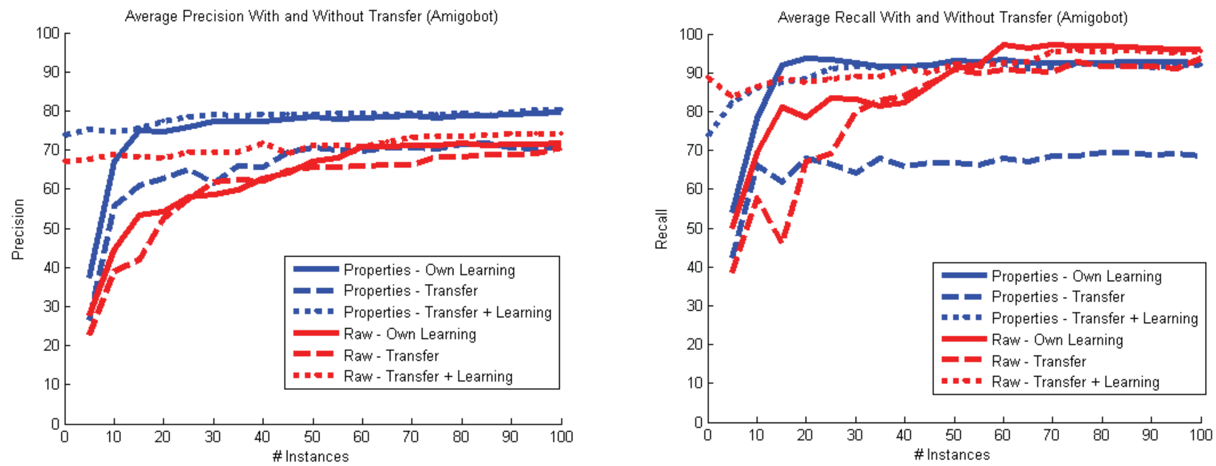


Figure 7: Results demonstrating the advantage of transfer learning. Even with no training on instances obtained by the receiving robot, a high recall (right) and precision (left) rates can be achieved. Learning can then continue, quickly achieving similar rates as when the robot only learns using its own instances.

slightly and hence are shown as well. For the Pioneer, transfer did not fail as catastrophically as for the Amigobot. However, the transfer was still considerably less effective, as you do not see the same high recall and precision rates, for example when no instances are seen by the receiving robot. In all, these results confirm our hypothesis that the abstraction of data into properties aids transfer, and is especially more effective than raw values when the robots utilize differing representations.

5. DISCUSSION AND CONCLUSIONS

In this paper, we have introduced a novel transfer learning problem that exists in robotics, where robots can be heterogeneous with respect to their sensing, perceptual processing, and representations. In order to alleviate these problems, we proposed to use an intermediate representation that abstracts raw sensory data. Such abstractions have been long proposed to improve learning, and in this paper we show that these abstractions can also facilitate transfer. Using inspiration from a psychologically-derived representation, conceptual spaces, we utilize natural object properties as the intermediate representation. We have shown empirically that these can be successfully learned and used in combination to classify a large number of everyday objects, using real data.

More importantly, we have shown that these properties can be mapped across heterogeneous robots using instances from each robot sensing the same scene. This can be done by manually choosing such images, teleoperation, or (as will be done for future work) autonomously using following or pointing behaviors. Once mapped, unshared properties can be considered missing attributes and the learned classifiers can be transferred. We have further shown the advantages of property abstractions by showing that transfer can occur even when the underlying property representations differ.

Given this foundation, several avenues of future work remain. First, there are several obvious improvements that can be made, many of which have been mentioned earlier. Namely, the process of achieving a shared context should be done autonomously, a process that requires either being able to detect other robots or shared localization. A more difficult, but important, challenge is to characterize the amount of overlap between properties instead of simply considering them either shared or unshared. Since the properties are supervised, it will be possible to change the training regime such that the resulting properties on different robots differ in their amounts of overlap. It will then be possible to see what effect this has on transfer learning.

Another important but difficult problem is to allow the robots to autonomously learn properties in an unsupervised manner. This would allow the entire framework to be automated, except for the supervised labels given during the learning of the object classes themselves. Allowing the robots to autonomously learn both the intermediate representation as well as the mappings across robots is the long-term goal of this work.

6. REFERENCES

[1] A. Billard and K. Dautenhahn. Grounding communication in autonomous robots: an experimental study. *Robotics and Autonomous Systems*, 1-2:71–81, 1998.

[2] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4, 1998.

[3] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[4] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[5] P. Gardenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.

[6] T. Joachims. Making large scale SVM learning practical. 1999.

[7] D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating

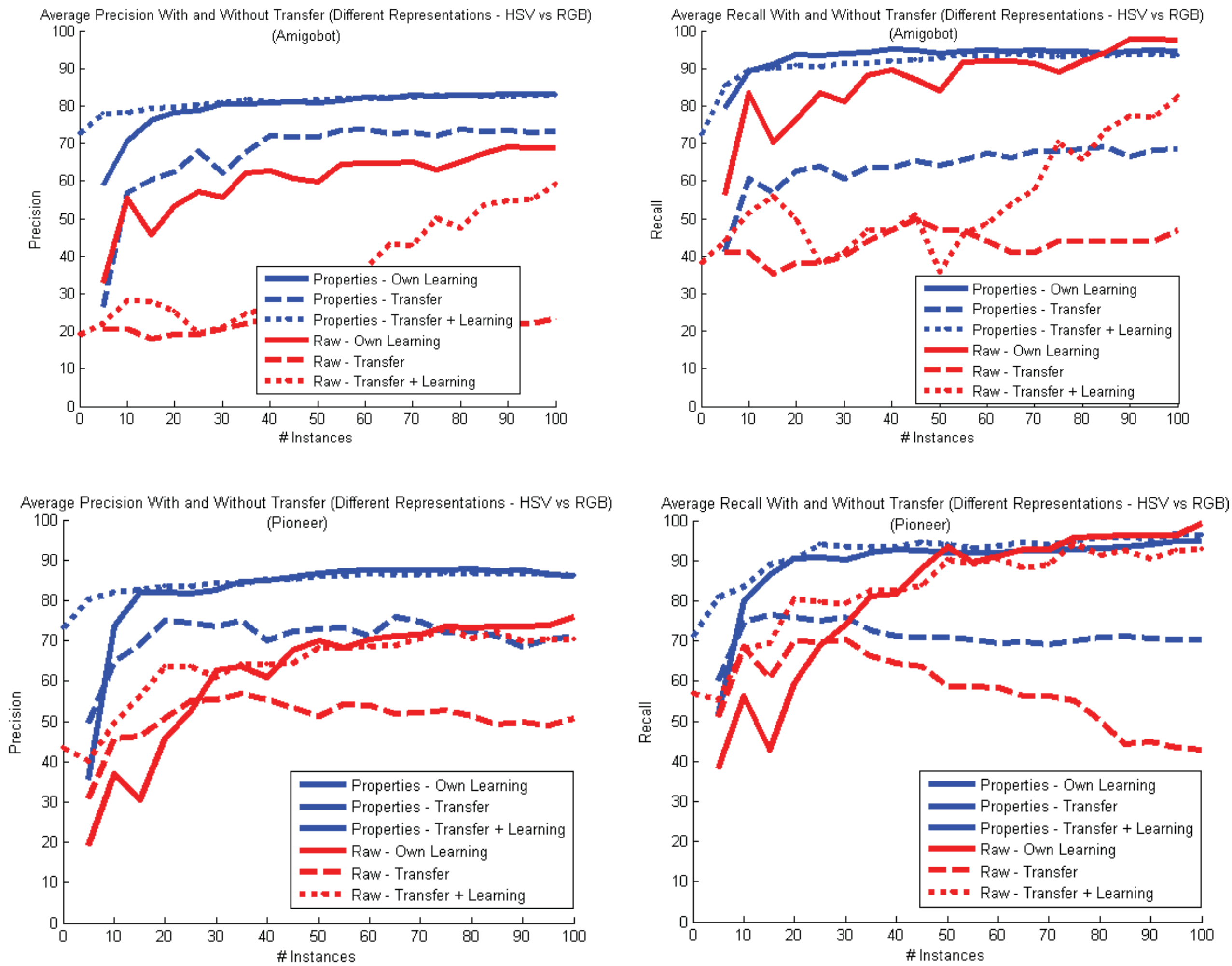


Figure 8: Results demonstrating the advantage of transfer learning when using properties. Here, the underlying representations used by the robots differ (one uses an RGB color space while the other uses HSV). Transfer learning when using raw sensory values fails, while transfer learning when using properties maintains its efficacy. Top two panels show precision (left) and recall (right) for the Amigobot while the bottom two show the results for the Pioneer robot.

robots. *Auton. Robots*, 8(3):269–292, 2000.

[8] Z. Kira. Mapping grounded object properties across perceptually heterogeneous embodiments. In *Proceedings of the Twenty-Second FLAIRS Conference*, pages 57–62, Sanibel Island, FL, USA, 2009. AAAI Press.

[9] Z. Kira. Transferring embodied concepts between perceptually heterogeneous robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4650–4656, 2009.

[10] Z. Kira and K. Long. Modeling robot differences by leveraging a physically shared context. In C. G. P. L. Berthouze, editor, *Proceedings of the International Conference on Epigenetic Robotics*, pages 53–59, 2007.

[11] J. Luo, A. Pronobis, and B. Caputo. SVM-based transfer of visual knowledge across robotic platforms. In *5th International Conference on Computer Vision Systems (ICVS)*, volume 7.

[12] S. J. Pan and Q. Yang. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.

[13] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, page 766. ACM, 2007.

[14] L. Steels and F. Kaplan. Bootstrapping grounded word semantics. Cambridge University Press, 1999.

[15] M. Taylor and P. Stone. Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.

[16] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, 2007.

[17] P. Vogt and F. Divina. Social symbol grounding and language evolution. *Interaction Studies*, 8(1), 2007.