Inferring Categories to Accelerate the Learning of New Classes

Robert Goeddel Edwin Olson

***Abstract*— On-the-fly learning systems are necessary for the deployment of general purpose robots. New training examples for such systems are often supplied by mentor interactions. Due to the cost of acquiring such examples, it is desirable to reduce the number of necessary interactions. Transfer learning has been shown to improve classification results for classes with small numbers of training examples by pooling knowledge from related classes. Standard practice in these works is to assume that the relationship between the transfer target and related classes is already known.**

In this work, we explore how previously learned *categories*, or related groupings of classes, can be used to transfer knowledge to novel classes without explicitly known relationships to them. We demonstrate an algorithm for determining the category membership of a novel class, focusing on the difficult case when few training examples are available. We show that classifiers trained via this method outperform classifiers optimized to learn the novel class individually when evaluated on both synthetic and real-world datasets.

I. Introduction

To deploy general purpose robots into the world, it is necessary to develop algorithms that can learn on the fly. Such algorithms allow systems to deal with the ever-increasing quantity of novel experiences a robot might encounter. In this paper, we present a method for discovering and exploiting categorical relationships between new and previously discovered classes. This allows us to learn classifiers for new objects with fewer training examples by transferring knowledge from previous examples. In particular, we focus on small-data domains, where training data is either sparse or expensive to acquire (e.g. a system driven by mentor interactions).

Transfer learning aims to re-use examples of previously encountered classes to improve classification performance for another related class [1]. In this case, a *class* refers to a distribution of objects that may be described by the same label (e.g. "red"). Some classes actually subsume sets of lower level classes, establishing a hierarchy (e.g. cats and dogs are both animals or running and walking are both forms of human locomotion). In this work, we refer to higher level classes such as animal as *categories*. Exploiting these semantic relationships between classes in the same category can lead to transfer of knowledge between domains [2].

An example domain in which a robotic system interacts with colored blocks of various shapes is illustrated in Fig. 1. We expect that, given prior knowledge of the characteristics of these blocks, the robotic system should be able to exploit this knowledge when a new class of block is first observed. For example, having seen the colors red, green, and blue, we hypothesize that knowing "purple" is also a color will allow the system to learn to identify purple from fewer examples.

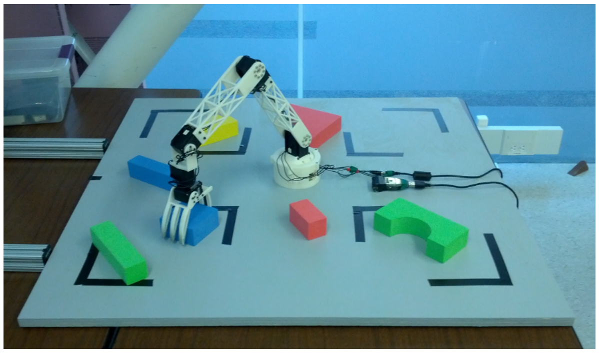


Fig. 1: A robot operating in a blocks-world domain. The system should be able to leverage prior experiences to accelerate the recognition of new blocks.

In transfer learning problems, it is commonly assumed that correspondence between the novel class and previously learned classes is known *a priori*. In the case of learning novel classes, though, the relationship to previously seen things may be unknown. Therefore, not only do we wish to use previous experiences to bootstrap the process of learning to classify novel classes, but also to deduce which experiences are most relevant to this task. In this paper, we show how to assign new classes to high-level categories for the purpose of improving classification performance. We construct a method for identifying category membership automatically and use this to perform transfer learning, demonstrating that this reduces the number of training examples necessary to train a serviceable classifier. Our contributions are:

- A method for discovering relationships between novel classes and known categories of classes.
- Evaluation of our method on synthetic and real-world data showing performance gains over classifiers trained to specialize on a specific class.
- Empirical characterization of the advantages of our method for few training examples as well as the possible failure modes of the algorithm.

II. Related Work

Transfer learning is the process of reusing knowledge from one domain in another. While the transfer domains may be tightly coupled (e.g. using previous knowledge of the category color to help learn the concept "purple"), some applications are less intuitive (e.g. learning action sequences for heterogeneous sets of robots) [3].

The authors are with the Department of Computer Science and Engineering, The University of Michigan, Ann Arbor, MI {rgoeddel,ebolson}@umich.edu

A variety of transfer learning strategies may be adopted depending on the available data [1]. For example, semi-supervised learning is an increasingly popular topic and can be applied to situations where labeled data is expensive, but unlabeled data is abundant [4]. Models can be fit to the unlabeled data that have predictive power with regards to the labeled data. For example, Pan et al. present a method for projecting unlabeled data from one domain and labeled data from another into a shared feature space such that the domains are better suited for transfer learning [5]. In general, though, semi-supervised learning techniques are most useful when it is known that the unlabeled data are instances of the labeled data, which is typically not the case for large, general purpose systems.

Another common application of transfer learning is to perform multi-class feature selection. Feature selection can greatly improve classification results when many irrelevant features are present [6], [7]. In addition, previous works have shown that pooling data from related classes for the purpose of simultaneous learning can lead to classification performance increases across classes [8]. The benefit is that feature selection may be performed such that one keeps features that are common across a set of related classes [9], [10]. Evgeniou et al. show that they are able to reduce the size of the feature space, reducing computation, and also that joint learning outperforms learning independent classifiers for each class.

This strategy can be particularly useful in the context of machine vision, as even unrelated classes of objects often share visual features [11]. Torralba et al. find that, not only do jointly learned visual features allow for similar performance to specialized classifiers while requiring fewer total features, but these features also tend to be more general purpose. As a result, the feature space only grows logarithmically with the number of jointly learned classes as many features can be reused.

A common theme for these techniques is that they rely on *a priori* knowledge of which classes are related to the class being bootstrapped. One strategy for discovering such relationships is to use WordNet or Wikipedia to learn semantic relatedness [12], [13]. It has been shown that such relatedness measures may be used to aid in visual classification [2], [14].

Unsupervised category discovery is also often employed to find relations between subsets of data for organizational or classification purposes [15], [16], [17]. These methods are powerful as they can take advantage of latent relationships between classes of data. While human-recognizable categories such as "animal" or "color" may be discovered, it is also possible that these methods will discover unique and unpredictable groupings of classes. The disadvantage to such methods is that it is not obvious which category a novel class belongs to.

In this work, we explore a variation of the transfer learning problem where categorical relationships between our initial classes are known, but where this information is not known for novel classes. The goal, then, is to 1) identify the category membership of novel classes and 2) use this to perform transfer learning. We show that extending our categories to cover new classes boosts early-stage learning for these classes in addition to providing additional object meta-information.

III. Methodology

The input for our problem is an observed feature vector, labeled with both class and category. In this work, we focus on objects with single-category membership. As a mentor provides new labeled training instances for previously unseen classes, we attempt to match the novel class to an existing category so that knowledge transfer may occur. The key idea in our work is that classes within a single category are related; specifically, the same features tend to be informative. This idea suggests both 1) a way of transferring knowledge and 2) a way of testing category membership.

Our proposed method may be outlined as follows:

1) Use known categories and the initial corpus of training data to perform joint feature selection on each category subject to the constraint that all classes within one a category use the same features.
2) When an example of a novel/uncategorized class is encountered, hypothesize that the novel class is a member of each known category in turn. For each category, construct a one-vs-all classifier based on the category's joint features that distinguishes between the novel class and other category members.
3) Evaluate each category-specific classifier using cross-validation to score its performance.
4) If one or more of the categories provides sufficiently improved performance over a standard one-vs-all classifier using all known features, propose that the novel class is a member of the best performing category.

In the following sections, we will discuss how to jointly select features for a category and how to evaluate the quality of a category membership hypothesis. Additionally, we will how to transfer knowledge from a category to a novel class.

A. Multi-Class Feature Selection

The first step is to jointly learn the appropriate features to describe each of the known categories. Category-specific features not only offer speed, but also boost classification accuracy. As feature selection can be slow, they can be initially calculated offline and then updated periodically as relevant training examples are added. In this work, we will not explore when periodic updating should happen, and instead will focus strictly on pre-computed category features.

We use a greedy feature elimination algorithm to select category features. Pseudocode for the algorithm is given in Alg. 1. The method is supplied with a set of related labels (the proposed category) as well as the relevant labeled training examples. The category feature set is initialized to contain every feature. Then features are greedily removed from the set, one at a time, based on the average cross-validation scores for all of the category classes. Features are removed until 1) only one feature is left in the category feature set or 2) the average cross-validation score decreases

Algorithm 1 Joint Feature Selection

```
$categoryFeatures \leftarrow$ all features
$thresh \leftarrow$ user defined performance threshold
$bestScore \leftarrow \text{score}(categoryFeatures)$
while $\text{size}(catFeatures) > 1$ do
    $bestFeatures \leftarrow \emptyset$
    for $feature \in categoryFeatures$ do
        if $\text{score}(bestFeatures) < \text{score}(categoryFeatures - feature)$ then
            $bestFeatures \leftarrow categoryFeatures - feature$
    if $bestScore - thresh < \text{score}(bestFeatures)$ then
        $categoryFeatures \leftarrow bestFeatures$
        $bestScore = \max(bestScore, \text{score}(bestFeatures))$
    else
        break
return $categoryFeatures$
```

below a threshold. Higher threshold values lead to the elimination of more features. For our tests, the threshold parameter is set to 0.01, which means that we are willing to tolerate no more than a percentage point drop in classification accuracy. The algorithm is run independently for each known category, resulting in categories learning specialized sets of features.

Accuracy, a common performance metric, is the ratio of true positives and true negatives to all results. An issue with this accuracy metric is that it is greatly affected by the distribution of the data. It measures the probability with which we expect to give a correct response in our classification task.

$$accuracy = \frac{T_P + T_N}{T_P + F_P + T_N + F_P} \tag{1}$$

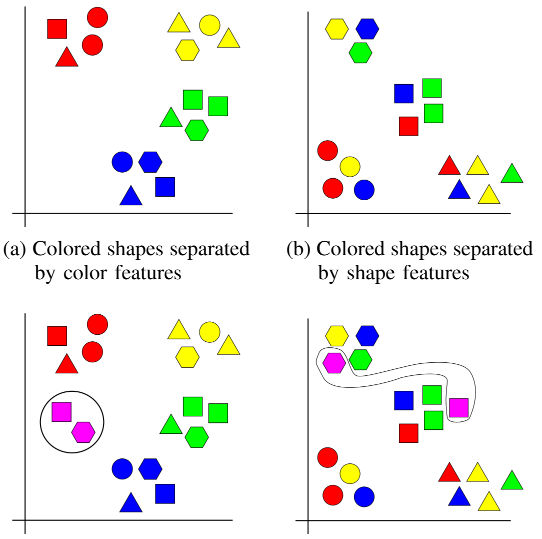
If 99% of training examples are negative, then a classifier that always reports "negative" will do very well according to the accuracy metric even though it never correctly recognizes a positive example. Our training sets, by nature, are imbalanced since we are learning a new class based on an existing large corpus of training data, making standard accuracy a poor evaluation metric.

To avoid issues arising from imbalanced data, one can use the balanced accuracy metric [18]. Balanced accuracy is the mean of sensitivity and specificity. It gives equal weight to both the true positive and true negative rates regardless of the distribution of data. This alleviates the problem of a single, more prevalent class dominating the feature selection process during joint selection. We use this metric in this paper.

$$balancedAccuracy = \frac{1}{2}\left(\frac{T_P}{F_N + T_P} + \frac{T_N}{F_P + T_N}\right) \tag{2}$$

B. Transferring Category Knowledge to Novel Classes

We hypothesize that, since category members are well distinguished by our jointly-learned category features, novel classes also belonging to that category will be distinguishable from existing category members. Conversely, novel classes *not* belonging to a category will be poorly separable, and thus difficult to distinguish, from the other category members as the features selected are irrelevant to the novel class. The categories "color" and "shape," pictured in Fig. 2, provide an illustrative example. As Figs. 2(a) and (b) show, we expect that features selected for color and shape will result in very different clusterings of identical objects. We exploit this expectation by observing the accuracy with which a novel class (in the example, purple), is classified. From this, we determine whether it is likely that purple is either a shape or a color. This allows us to construct a procedure for testing the category membership of novel classes.



(a) Colored shapes separated by color features

(b) Colored shapes separated by shape features

(c) Purple as distinguished by color features

(d) Purple as distinguished by shape features

Fig. 2: A toy example of placing "purple" into the feature spaces for the categories color and shape. When placed in the correct category, color, purple training examples are tightly clustered, making it easily distinguishable from other colors. When placed into an incorrect category such as shape, purple training examples are scattered throughout the feature space, making purple difficult to separate from other classes.

When provided with a new example of a novel class, test the hypothesis that the class is a member of one of the existing categories. For each known category, train a one-vs-all classifier to identify the novel class using the subset of jointly-selected features for that category. Hypothesized novel class membership in each category is scored by computing the balanced accuracy for the associated classifier via cross-validation. If the novel class is well separable from other category members, then the cross-validation score should be high, indicating potential category membership.

IV. Evaluation

In this section, we define an experimental procedure for evaluating learning strategies. We present experimental results for both synthetic and real-world data and examine strengths and possible failure modes of our method.

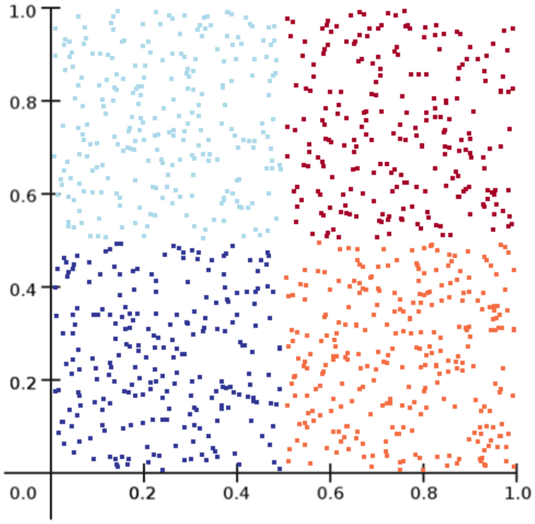


Fig. 3: An example of synthetic data from a category defined by two features and with four member classes. The axes correspond to feature values, with points representing randomly generated object instances. The colors of the points correspond to the ground truth class of the data point.

A. Classification Methods

Our first method of classification is the nearest-neighbor classifier. Nearest-neighbor classifiers are appealing as they are easy to implement and because they are non-parametric. The nearest-neighbor classifier is a special case of the k-NN, which returns a classification based on the votes of the k closest training examples to the query point. We find that values of $k > 1$ offered little to no benefit in this domain, and in fact, can be detrimental to the goal of learning from very few examples. This is because higher values of k require more positive examples to return a positive classification. In our domain, these examples may not yet exist, delaying our ability to correctly classify a new class of object. Thus, we only present results for the nearest-neighbor in this paper, i.e. $k = 1$.

Our second method of classification is the support vector machine (SVM) [19]. SVMs are popular in learning communities due to their classification efficiency and predictive power [20]. SVMs create decision boundaries for classification by maximizing the margin (separation) between the positive and negative examples of training data. One downside to SVMs is that training can be expensive. We use the LibSVM implementation of SVMs with a Gaussian kernel and parameters tuned by a grid search through the recommended ranges of values [21].

B. Experimental Procedure

In this section, we define a testing procedure in which nearest-neighbor or SVM classification may be used interchangeably. We compare three different strategies for learning novel classes.

1) *Baseline*: Train a one-vs-all classifier to discriminate between the novel class and all other classes using all available features.
2) *Single Class*: Select features to optimize the cross-validation score for a one-vs-all classifier distinguishing between the novel class and all other classes.
3) *Category Transfer*: Select the category-based classifier with the set of jointly-learned features that gives the highest cross-validation score for a one-vs-all classifier distinguishing between the novel class and all other classes.

SVM parameters are optimized individually for each method. For the generic classifier with no feature selection, a grid search through the parameter space is performed based on all of the available training data. For both of the feature selection-based methods, we find that SVM performance is improved most by optimizing the parameters for classification solely on relevant category examples.

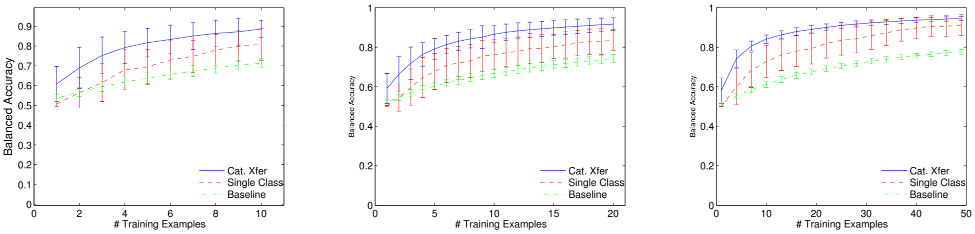
Using the above strategies, we perform 100 test trials. For each trial, we generate a training set consisting of an equal number of examples of each class. We select a random class to withdraw from the training set and consider this our “novel class.” To observe how each classification strategy performs, we introduce instances of the novel class one at a time and evaluate each against a test set.

Our expectation is that the baseline should exhibit the worst classification performance as it is most likely to suffer from problems due to high-dimensionality. Performing feature selection for the novel class alone should provide improvement over the baseline as it will eliminate features irrelevant to distinguishing examples of the class. Our method should outperform both as correctly placing a class in a category allows us to utilize previous observations when choosing our features. We expect the largest gains to be made early, before the single class feature selection has enough training examples to learn a suitable feature mask.

C. Synthetic Data

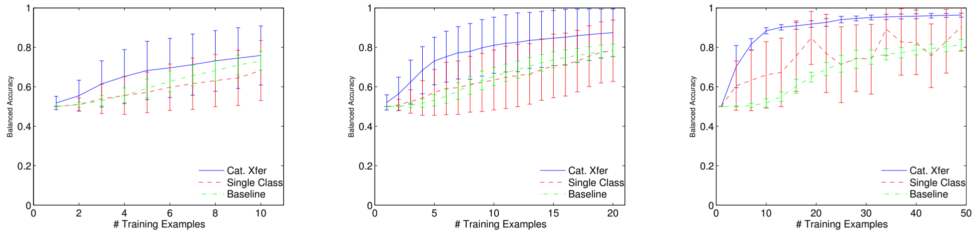
All synthetic features are normalized values between 0 and 1. While normalization is not strictly necessary, it facilitates the use of classifiers such as nearest-neighbor, where larger feature values can give disproportionate weight to those features. Classifiers such as SVMs have also been shown to be sensitive to feature normalization [22]. Alternatively, normalization factors may be learned online. However, in our domain, where data is likely to be sparse, this can be difficult as insufficient data may have been seen to accurately scale feature values.

Categories are defined by a unique, non-overlapping subsets of features. Classes belonging to a category are defined by unique ranges of values that each category-specific feature may assume. We generate these ranges such that each class occupies 1) an equal volume of the category hyperspace and 2) is perfectly separable from the other classes. Random examples are generated by selecting a set of class memberships (one for each category) for each training example and by populating the relevant features with values sampled uniformly at random from the class-defined ranges. Example synthetic data can be seen in Fig. 3.



(a) 10 ex/class (b) 20 ex/class (c) 50 ex/class

Fig. 4: Mean nearest-neighbor classification results for varying amounts of category data. Error bars indicate one standard deviation. Cat. Xfer denotes our category transfer method. Regardless of the number of training examples provided for each existing class, our method is able to train higher accuracy classifiers than methods not employing knowledge of categories.



(a) 10 ex/class (b) 20 ex/class (c) 50 ex/class

Fig. 5: Mean SVM classification results for varying amounts of category data. Error bars indicate one standard deviation. Cat. Xfer denotes our category transfer method. Though baseline learning performance improves for SVM compared to nearest-neighbor classification, our method is still able to improve classification accuracy, particularly when we have very few training examples of the novel class.

In our presented synthetic results, we use 3 categories, each with 8 classes. The feature vectors for training examples in those classes are separable using three dimensions. We find these results to be representative of other synthetic category structures tested. All results are based on a set of 10,000 synthetic data points.

D. *Real-world Data*

Additional testing is performed on two real world datasets. The first is Munroe's XKCD Color Survey, from which we take a set of 196,000 labeled color data points identifying 27 distinct colors [23]. From this, we remove two colors with insufficiently large amounts of training examples (1 and 2 examples, respectively) and select test and training sets from what remains.

We also examine results for a small data set collected in the blocks domain pictured in Fig. 1. We collected observations of colored blocks using a Microsoft Kinect. A total of 105 blocks were automatically segmented from the scene and hand-labeled for color, shape, and size. We create a test set by randomly selecting 30% of the training examples for each class and then train our classifiers on the remainder. This domain is intended to emphasize learning from very small amounts of data, as all labels must be solicited from a human mentor and are thus costly to obtain.

E. *Synthetic Data Experimental Results*

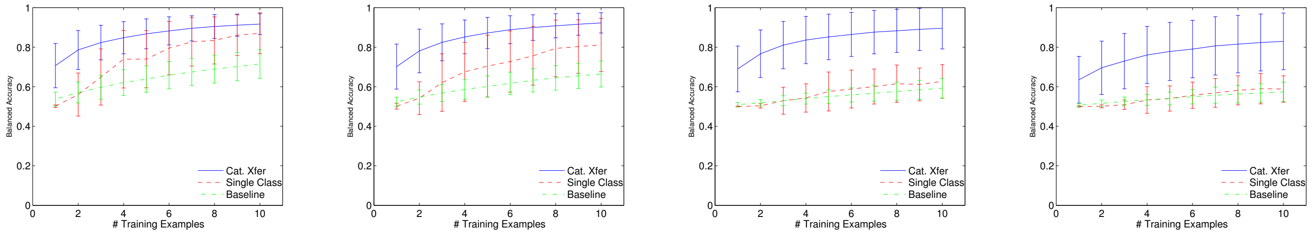
Nearest-neighbor classifier results for 10, 20, and 50 instances of each known class can be seen in Fig. 4.

The baseline algorithm improves as more data is acquired, but improvement is slow and quickly plateaus. This indicates that the method is unable to filter out the important information given by relevant features from the rest of the feature space.

Single class feature selection is better able to filter out irrelevant features, but the quality of features selected varies wildly for the first several training examples. This is likely due to overfitting the selected features to such small amounts of data. Data from individual trials backs up this hypothesis, as the features selected vary (sometimes dramatically) as new training data is added. These fluctuations in feature selection manifest themselves as large fluctuations in accuracy during testing, leading to the high variance seen in the single class results.

Classifiers trained via category transfer provide larger and more consistent improvements in accuracy. This is because, during joint feature selection, we are able to pool data from many classes to learn a general set of features that works well for all category members. Most importantly, even for very little training data, category knowledge provides a substantial boost in performance over the baseline, achieving over half its gains within seeing the first 5 training instances of a novel class. In short, these results demonstrate that our approach significantly reduces the number of training examples required in order to learn a serviceable classifier.

Results for the same tests run with SVMs can be seen in Fig. 5. The average performance of the SVM baseline and category transfer methods exhibit similar trends to those seen



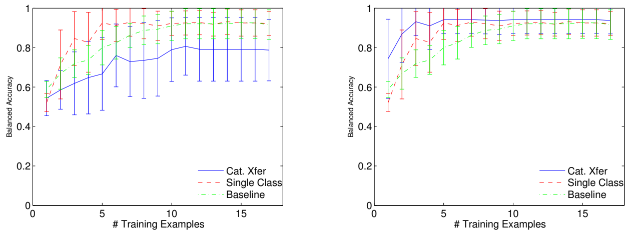
(a) 3 random features

(b) 5 random features

(c) 20 random features

(d) 30 random features

Fig. 6: Mean nearest-neighbor classification results on the XKCD Color Survey for increasing amounts of distractor features. Category transfer proves to be most robust to extraneous features.



(a) Joint FS mask

(b) Ground Truth mask

Fig. 7: Category transfer for identifying the "arch" shape with jointly learned category feature masks (a) and with human-specified feature masks (b). Categories insufficiently covered by classes are prone to overfitting, and thus perform more poorly than even the baseline.

for nearest-neighbor classification, though variance for both methods is higher. Surprisingly, the single class optimization method often performs worse than the baseline. This is likely due to the fact that SVMs are better able to deal with high-dimensionality than nearest-neighbor. Thus, while single class optimization still suffers from overfitting the feature space (and may even remove relevant features), the baseline SVM merely discounts less important features without completely discarding them.

Nearest-neighbor outperforms SVM on average for very few novel training examples. For this reason, all real-world data examples are compared with nearest-neighbor classification.

F. Real-world Experimental Results

As only one category is present in the XKCD Color Survey, we alter the experimental procedure to obscure the relevant features. For a given color example, a feature vector is initialized containing red, green, and blue color channel values in the range $[0, 1]$. Then, the feature vector is "polluted" with extra features, values for which are sampled from the Gaussian distribution $\mathcal{N}(0.5, 0.05)$ and clamped between 0 and 1. For few examples, these features are hard to distinguish from the actual distributions of the relevant color features. For each experimental trial, 10 random examples of each color are chosen as training data. We then vary the number of distractor features added until the system fails. It is expected that category transfer will not only increase performance the most, but also be best able to filter out the noisy features.

Experimental results for nearest-neighbor classification on the XKCD Color Survey can be seen in Fig. 6. Our results show that by employing category knowledge, our approach improves classification accuracy. As expected, as the number of extraneous features increases, category transfer performance is largely unaffected, while both the baseline and single class feature selection methods rapidly degrade in classification accuracy. Category transfer finally begins to fail around 30 extraneous features, at which point 90% of the features are noisy. Preliminary investigation indicates that the point of failure follows a roughly linear relationship to the amount category training data.

G. Real-world Failure Mode

The results for the Kinect blocks dataset for novel colors unsurprisingly resemble the XKCD Color Survey results (though both the baseline and single class methods improve in average performance). The shape category presents an interesting challenge, though. The feature space designed for the color, shape, and size categories is dominated by many shape features. However, variety in shape training data is limited. Thus, joint feature learning is susceptible to removing important shape features when learning the shape category.

For example, the majority of the shape objects in the data set are angular in nature (e.g. squares, triangles, rectangles). One shape, however, the "arch", is a rectangle with a semi-circular portion cut from one side. An example of a green arch can be seen in the lower left of Fig. 1. When the shape category is learned without the arch present, feature selection removes the features that would allow the classifier to distinguish an arch from a rectangle as they are not yet necessary. As a result of this premature feature removal, when the arch is introduced, category transfer actually performs worse than the other methods, as can be seen in Fig. 7. When category masks are hand-selected, category transfer once again demonstrates performance gains over the competing methods.

The implication of this result is that, while category transfer can allow us to make better use of small amount of prior training data, it is imperative that the prior category data cover the relevant feature space well. Otherwise, category transfer is susceptible to overfitting the category feature mask to the data.

V. Conclusion

In this work, we demonstrate how transfer learning can be used to boost early classification performance, reducing the number of interactions necessary for a robot to learn a serviceable classifier, even when the novel class's category membership is unknown. We present a method for determining membership of a novel class in the absence of known category relationships and demonstrate that this method works for both synthetic and real-world data. We empirically characterize the properties of the algorithm, showing that the majority of gains are made within the first 5 observations of a novel class and identifying the types of domains which will benefit most from category transfer.

Acknowledgements

The authors would also like to thank Lauren Hinkle for her useful insight during early stages of this research. This work was supported by ONR Grant N00014-13-1-0217.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba, "Semantic label sharing for learning with many categories," *Computer Vision–ECCV 2010*, pp. 762–775, 2010.

[3] B. Lakshmanan and R. Balaraman, "Transfer learning across heterogeneous robots with action sequence mapping," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3251–3256.

[4] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[5] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *Neural Networks, IEEE Transactions on*, vol. 22, no. 2, pp. 199–210, 2011.

[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[7] S. Deegalla and H. Boström, "Classification of microarrays with knn: comparison of dimensionality reduction methods," *Intelligent Data Engineering and Automated Learning-IDEAL 2007*, pp. 800–809, 2007.

[8] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," *Learning Theory and Kernel Machines*, pp. 567–580, 2003.

[9] G. Obozinski, B. Taskar, and M. I. Jordan, "Multi-task feature selection," *Statistics Department, UC Berkeley, Tech. Rep*, 2006.

[10] A. A. T. Evgeniou and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, vol. 19. MIT Press, 2007, p. 41.

[11] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 5, pp. 854–869, 2007.

[12] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[13] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of the 20th international joint conference on Artifical intelligence*, 2007, pp. 1606–1611.

[14] J. Deng, A. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" *Computer Vision–ECCV 2010*, pp. 71–84, 2010.

[15] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," *Data mining and knowledge discovery*, vol. 10, no. 2, pp. 141–168, 2005.

[16] M. Wang, X. Zhou, and T.-S. Chua, "Automatic image annotation via local multi-label classification," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, ser. CIVR '08. New York, NY, USA: ACM, 2008, pp. 17–26.

[17] J. Zhang, J. Zhang, S. Chen, Y. Hu, and H. Guan, "Constructing dynamic category hierarchies for novel visual category discovery," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2122–2127.

[18] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 3121–3124.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] N. Cristianini, "Support vector and kernel machines," *Tutorial at ICML*, 2001.

[21] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[22] A. Ben-Hur and J. Weston, "A users guide to support vector machines," in *Data mining techniques for the life sciences*. Springer, 2010, pp. 223–239.

[23] R. Munroe, "Color survey results," 2010. [Online]. Available: http://blog.xkcd.com/2010/05/03/color-survey-results/