Measuring Cumulative Gain of Knowledgeable Lifelong Learners

Diana Benavides-Prado
Dept. of Computer Science
The University of Auckland
Auckland, New Zealand
dben652@aucklanduni.ac.nz

Yun Sing Koh
Dept. of Computer Science
The University of Auckland
Auckland, New Zealand
ykoh@cs.auckland.ac.nz

Patricia Riddle
Dept. of Computer Science
The University of Auckland
Auckland, New Zealand
pat@cs.auckland.ac.nz

Abstract

Lifelong machine learning has been a long-standing challenge for the machine learning community. Recent research defined core properties of lifelong machine learning systems, including the ability to perform continuous learning for these systems to become more knowledgeable over time. Measuring if such systems are increasingly knowledgeable is challenging since performance metrics for single task learners may be insufficient or misleading. We introduce Cumulative Gain of a Lifelong Learner, a metric to determine gain in performance for systems that learn supervised tasks sequentially. This gain can be achieved by learning new tasks better, by refining existing knowledge of previous tasks or by satisfying both of these properties. The proposed metric is agnostic to the lifelong learner. We evaluate our metric experimentally in large-scale synthetic datasets for binary classification tasks and real-world datasets of varied number of classes, using our own lifelong learner and counterparts.

1 Introduction

Learning in the long-term is a long-standing concern in machine learning. Lifelong machine learning has revived interest in systems that learn a sequence of related tasks [1]. These systems should: 1) learn new tasks better, exploiting existing knowledge; 2) store knowledge incrementally in a knowledge base; 3) perform continuous learning. Research in transfer, hypothesis transfer, multitask, meta and deep learning explored the first property [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. Research in lifelong learning and related areas have studied the second [1]. The last property, which should ideally include refinement of knowledge, has been explored only recently [24, 25]. Some work has recently focused on the challenge of learning new tasks whilst preventing *catasthropic forgetting* of old tasks [26]. The exploration of the problem of learning continuously whilst refining knowledge has been rather limited. Measuring if supervised lifelong learning systems are becoming more knowledgeable has been even less explored, and to the best of our knowledge only one test has been formulated to date [27].

In this paper we introduce CGLL, a simple metric for measuring improvement on the performance of supervised lifelong learning systems. We accompany this metric with a test to determine if a lifelong learner can be categorised as a continuous learner that encourages a system to become more knowledgeable over time as it observes more tasks. We evaluate our metric on two synthetic datasets developed specifically for binary classification tasks in lifelong learning, and three real-world datasets of different number of classes, using existing lifelong learners and our own method. Section 2 introduces the metric. Section 3 briefly introduces our lifelong learner. Section 4 presents experimental results. Finally, Section 5 concludes and identifies future directions.

Preprint. Work in progress.

2 CGLL: A General Metric for Supervised Lifelong Learning

Chen and Liu (2016) defined the ability to perform continuous learning as one of the core properties of lifelong machine learning systems. Learning continuously should ideally encourage the system to become more knowledgeable. Therefore, a lifelong learning system should demonstrate better performance over time. Measuring this performance is however a challenge since common metrics have been originally designed for learning settings where tasks are executed in isolation.

Li and Yang (2015) proposed a lifelong machine learning test to determine if an agent could be categorised as a lifelong learner. A learning agent A would pass the lifelong machine learning test depending on two conditions: 1) if its macroaveraging (mean) accuracy is better than a base learner B that learns these tasks separately, at every timestamp t during a sequence of tasks, and 2) the difference in the macroaveraging accuracy obtained by these two learners becomes larger and larger over time. The test was formally defined as [27]:

$$\forall t : \begin{cases} MA(A^{(t)}) > MA(B^{(t)}) \\ \nabla MA(A^{(t)}) > \nabla MA(B^{(t)}) \end{cases} \tag{1}$$

where $\nabla MA^{(t)} = MA^{(t)} - MA^{(t-1)}$ and $MA^{(t)} = \sum_{t=1}^{T} 1/n^{(t)} \sum_{i=1}^{n^{(t)}} P(y_i^t, f_t(x_i^t))$, with T the number of tasks learned, n the number of examples of all tasks and P a performance metric such as accuracy[1]. Note that multiple lifelong learning agents could be compared indirectly by comparing to the base learner B.

Inspired by this metric, we propose Cumulative Gain of a Lifelong Learner (CGLL) as an alternative simple metric to determine the cumulative gain in performance achieved by a lifelong learner. Intuitively, a lifelong learning system that is becoming more knowledgeable should demonstrate a larger gain over time. For a sufficiently large number of tasks that are all equally relevant for the learning system, the system should denote increasing performance if it is sufficiently good at learning new tasks as well as at refining existing knowledge of previous tasks. This can be potentially achieved if such system denotes two of the characteristics identified by Chen and Liu (2016): 1) learning new tasks better and 2) performing continuous learning whilst refining knowledge of existing tasks.

We define the cumulative gain achieved by a lifelong learning system that satisfies these characteristics as:

$$CG(LL)^t = CG(LL)^{(t-1)} + \frac{1}{T_t} \sum_{i=1}^{T_t} P(y_{si}, f_{si}^t) - P(y_{si}, f_{si}^{(t-1)}) \tag{2}$$

with $CG(LL)^0 = 0$. The cumulative gain CG of a lifelong learner LL at a timestamp t depends on the cumulative gain of that learner at the previous timestamp $(t-1)$ and the aggregation of differences in performance for a set of functions or hypotheses $f_s \in S$ at these timestamps, measured using a performance metric P such as accuracy. Similarly to Li and Yang (2015), we propose a test to determine if a lifelong learner can be categorised as a learner that is becoming more knowledgeable. Our test is defined as:

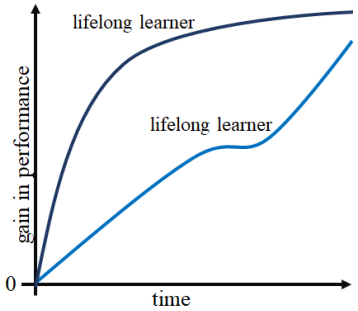

Figure 1: Two lifelong learners with increasing gain in performance.

$$\forall t : \begin{cases} \text{if} \quad t > 0 \quad CG(LL)^{(t)} \geq CG(LL)^{(t-1)} \\ \text{if} \quad t = 0 \quad CG(LL)^{(t)} = 0 \end{cases} \tag{3}$$

i.e. at any time t the cumulative gain of the performance of the system should be at least equal to the previous cumulative gain, *i.e.* $CG(LL)$ is at least monotonically increasing. Figure 1 shows two examples of lifelong learners that would be categorised as learners that encourage increasing performance over time. Note that the test assumes that all the tasks are equally important. For tasks of similar performance, it will also tend to be more stable over time (dark blue line in Figure 1).

[1]Though the original formulation by Li and Yang (2015) referred to a loss function $\mathcal{L}$, the formulation actually was meant to use a performance metric such as P.

3 HRSVM: A Lifelong Learner based on SVM

In our lifelong learning setting, a sequence of target T tasks is to be learned. At each timestamp t, the system executes a task T_t to learn a target function f_t using a set of labeled training examples $D_t = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. A lifelong learning system accumulates knowledge extracted on these tasks as a set $S = \{f_{s1}, \ldots, f_{st}\}$ of source hypotheses. We encourage transfer forward to learn a target f_t and transfer backward to refine existing $f_s \in S$ as follows.

Transferring Forward. A subset $F \subseteq S$ of source hypotheses is used to aid learning of T_t by transferring selected knowledge forward [17]. This subset is selected based on the relatedness of each $f_s \in S$ and D_t, measured using Kullback-Leibler divergence. Related source support vectors $x_{si} \in f_s \in F, 1 \leq i \leq l$, are later identified for each target example x_i, $1 \leq i \leq n$. Coefficients $\alpha_{si} \in f_s \in F$ are transferred and used to upper-bound coefficients for the corresponding target examples. As a result, training examples which are more closely resembled by source support vectors x_{si} get more importance while learning f_t, and contribute more to the objective to optimize. The learning problem in T_t is approached through an SVM dual objective [28], with a modified constraint:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ &s.t. \sum_{i=1}^{n} y_i \alpha_i = 0, \forall i \; 0 \leq \alpha_i \leq C + c_i, c_i = \frac{|F|}{|S|} \sum_{k=1}^{s} \alpha_k \end{aligned} \tag{4}$$

A coefficient α_i for f_t is upper-bounded by the constraint $C + c_i$, composed of the original upper-bound C and c_i, an aggregation of $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ coefficients transferred from s source support vectors $x_s \in f_s \in F$. A factor that accounts for the number of f_s contributing to the target task, $|F|/|S|$, is also considered. Details can be found in our previous paper [17].

As a result of this transfer process, tuples can be identified that match source support vectors (x_s, y_s, α_s) with target support vectors (x_t, y_t, α_t) learned for f_t, which were involved in transfer. A tuple is represented as $Z = \{(x_s, y_s, \alpha_s), (x_t, y_t, \alpha_t)\}$. These tuples contain insights on the knowledge shared by f_s and f_t. Therefore, exploiting these pairs could be potentially useful for a learning system that aims to refine existing f_s using knowledge collected on the recent task T_t. The next section explains a method to exploit this knowledge.

Transferring Backward. We approach the problem of refining an existing f_s by maximising an SVM dual function that includes an additional term to represent subspaces of shared knowledge between an existing f_s and a target f_t learned recently. This modification allows to optimize for the space of support vectors in f_s and the space of support vectors in f_s and f_t simultaneously. We propose a formulation that pursues refinement whilst encouraging retention of existing knowledge. The latter is a relevant characteristic for lifelong learning systems to remain in the long-term [29]. Our formulation is based on ν-SVM [30, 31], an SVM variant that considers a parameter ν to control the influence of training examples. Originally, this parameter limits both the degree of compression of an SVM hypothesis, acting as a lower bound on the number of support vectors, and the training error, acting as an upper bound on the number of margin errors. In our method, refinement of an f_s is controlled by controlling training error, whilst retention of knowledge in f_s is controlled by controlling compression. We formulate the hypothesis refinement problem with retention as[2]:

$$\begin{aligned} &\max_{\alpha} F(\alpha) = -\frac{1}{2}\left[(1-\Gamma) \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \Gamma \sum_{i,k=1}^{l,2o} \alpha_i y_i \alpha_k K(x_k, x_i)\right] \\ &s.t. \sum_{i=1}^{l} y_i \alpha_i = 0, \sum_{i=1}^{l} \alpha_i \geq \nu, \forall i \; 0 \leq \alpha_i \leq 1/l \end{aligned} \tag{5}$$

The first term in brackets optimizes on the space of the current $x_s \in f_s$. The second term optimizes on the shared space of the source f_s and the target f_t hypotheses. Here, α_k and x_k, with $1 \leq k \leq 2o$, are extracted from o functions learned with one-class SVM [32]. Each of these functions uses elements from one tuple $Z = \{(x_s, y_s, \alpha_s), (x_t, y_t, \alpha_t)\}$, as training examples. In binary classification tasks $Z = \{(x_s, y_s, \alpha_s), (x_t, y_t, \alpha_t)\}$ are conformed such that $y_s = 1$ and $y_t = 1$, or $y_s = -1$ and $y_t = -1$, to encourage transfer between corresponding classes. A parameter Γ, set generally small, controls the contribution of the last term.

[2]Details of this method are currently under review in another venue.

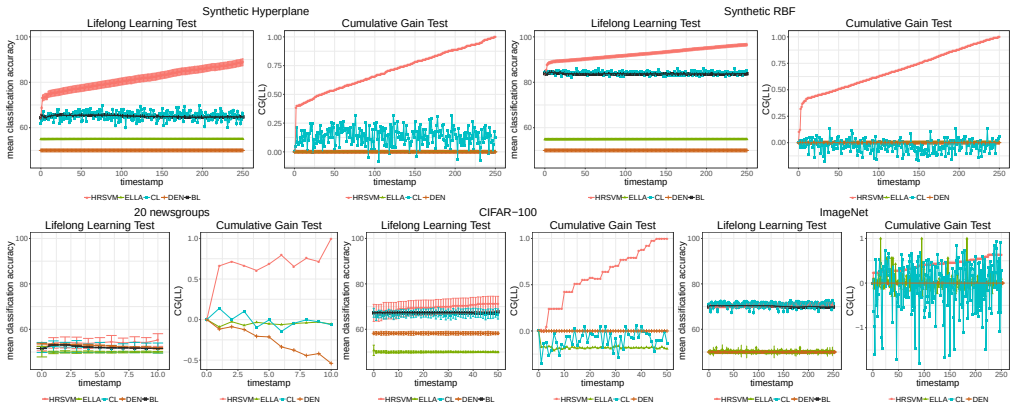

Figure 2: Lifelong learning test [27] and a new test to measure cumulative gain achieved by lifelong learning systems, evaluated on two synthetic and three real-world datasets. Error bars denote 95% c.i. t_0 denotes performance after half of the tasks have been learned. Note that, while for HRSVM t_0 is the initial knowledge base without transfer, for ELLA, CL and DEN t_0 some transfer/refinement may have already occurred since half of the tasks have been learned sequentially using these methods.

4 Experimental Evaluation

We experiment with synthetic and real-world datasets. We explore two kinds of synthetic problems: hyperplanes and RBF concepts. We generate these datasets as described in the supplementary material. For real-world datasets, we explore 20 newsgroups, CIFAR-100 and ImageNet. The experimental setting for HRSVM, ELLA, CL and DEN is described in the supplementary material.

Figure 2 (left for each dataset) shows the existing lifelong learning test [27] applied using HRSVM, ELLA, CL and DEN[3]. BL is an SVM base learner that learns tasks separately. For synthetic hyperplane, this learner is trained with a linear kernel. For synthetic RBF this is trained with an RBF kernel and $\gamma = 0.1$. For 20 newsgroups, CIFAR and ImageNet this learner is trained with an RBF kernel and $\gamma = 1/d$, with d the number of features. In all cases, $C = 1$. We can observe that HRSVM achieves increasing performance over time with an increasing gap with respect to the base learner BL, for synthetic hyperplane and synthetic RBF. Methods such as ELLA, and DEN find it difficult to achieve increasing performance, whilst their gap with respect to BL remains almost constant over time. CL is volatile over time. For real-world datasets some gap can be achieved also using HRSVM, compared to other methods that seem more unstable.

Figure 2 (right for each dataset) shows the proposed CGLL metric on HRSVM, ELLA, CL and DEN. For proper visualization, we normalize the cumulative gain over all methods using min-max normalization, with the minimum set to 0. Therefore, in practice the gain of each method is also relative to its counterparts. Similarly to the previous test, for synthetic hyperplane and synthetic RBF we observe that HRSVM, which aims to refine existing knowledge while learning new tasks, can effectively encourage increasing gain. This is an indication of a learning system that is becoming more knowledgeable. ELLA, CL and DEN find it difficult to pass this test on these datasets. For real-world datasets some gain can be achieved also using HRSVM, compared to other methods that seem more unstable according to this metric.

5 Conclusion and Future Work

We have proposed a general metric to determine if a lifelong learning system is becoming more knowledgeable. As future work we propose to analyse theoretical properties of this metric and extended scenarios such as systems composed of tasks inherently different in terms of performance, or systems for which some tasks are more relevant than others.

[3] After extensive experimentation with a variety of parameters we confirmed that for a large number of tasks the performance of DEN remains around 50%. Therefore, for synthetic hyperplane, synthetic RBF and ImageNet we only run up to 100 tasks using this method. We show extended results for proper visualization.

References

[1] Zhiyuan Chen and Bing Liu. Lifelong Machine Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145, 2016.

[2] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359, 2010.

[3] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):1–40, 2016.

[4] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv:0907.1815*, 2009.

[5] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, pages 289–296, 2009.

[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520, 2011.

[7] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE CVPR*, pages 2066–2073, 2012.

[8] Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, pages 702–715. Springer, 2012.

[9] Rich Caruana. Multitask Learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[10] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *ACMMM*, pages 188–197, 2007.

[11] Yusuf Aytar and Andrew Zisserman. Tabula rasa: Model transfer for object category detection. In *IEEE ICCV*, pages 2252–2259, 2011.

[12] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Learning categories from few examples with multi model knowledge transfer. *IEEE TPAMI*, 36(5):928–941, 2014.

[13] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. Transfer learning through greedy subset selection. In *ICIAP*, pages 3–14. Springer, 2015.

[14] Luca Oneto, Alessandro Ghio, Sandro Ridella, and Davide Anguita. Shrinkage learning to improve SVM with hints. In *IJCNN*, pages 1–9, 2015.

[15] Azadeh Sadat Mozafari and Mansour Jamzad. A SVM-based model-transferring method for heterogeneous domain adaptation. *Pattern Recognition*, 56:142–158, 2016.

[16] Yu-Xiong Wang and Martial Hebert. Learning by Transferring from Unsupervised Universal Sources. In *AAAI*, pages 2187–2193, 2016.

[17] Diana Benavides-Prado, Yun Sing Koh, and Patricia Riddle. Accgensvm: selectively transferring from previous hypotheses. In *IJCAI*, pages 1440–1446. AAAI Press, 2017.

[18] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[19] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE CVPR*, pages 1717–1724, 2014.

[20] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016.

[21] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE TPAMI*, 2017.

[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv:1703.03400*, 2017.

[23] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2017.

[24] Paul Ruvolo and Eric Eaton. ELLA: An Efficient Lifelong Learning Algorithm. *ICML*, 28:507–515, 2013.

[25] Geli Fei, Shuai Wang, and Bing Liu. Learning cumulatively to become more knowledgeable. In *KDD*, pages 1565–1574, 2016.

[26] Jaehong Yoon, Eunho Yang, et al. Lifelong learning with dynamically expandable networks. *arXiv:1708.01547*, 2017.

[27] Lianghao Li and Qiang Yang. Lifelong Machine Learning Test. In *AAAI Workshop on "Beyond the Turing Test"*, 2015.

[28] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[29] Andrew J Bremner, David J Lewkowicz, and Charles Spence. *Multisensory development*. Oxford University Press, 2012.

[30] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[31] Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

[32] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[33] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *JMLR*, 11(May):1601–1604, 2010.

[34] Tom M Mitchell. *Machine Learning*, volume 45. Burr Ridge, IL: McGraw Hill, 1997.

[35] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255. IEEE, 2009.

Supplementary Material

In this section we explain the synthetic datasets used in our experiments, and the detailed experimental setup for these and the real-world datasets.

We propose a synthetic hyperplane dataset for binary classification tasks received sequentially. This dataset is especially useful to explore the proposed metric experimentally, since the corresponding tasks are expected to perform similarly. In our dataset, each of these tasks is about learning a linear boundary separating examples from two classes. To generate each of these problems, we use an existing method that generates isotropic Gaussian blobs for clustering[4]. This method first generates c random centers in d dimensions. Then, examples within 1 standard deviation of these centers are generated randomly. We generate 500 problems using the described procedure. We set c to 2 for binary tasks. For results presented in Section 4, we set d to 100. We generate $1,000$ of these examples. We also scale the data to the range $[0, 1]$. To make each task subject to transfer and refinement, we repeatedly extract 10% of the corresponding examples as training data, and 30% as test data. Each extraction is performed without replacement. We repeat this procedure 30 times such that we have 30 different samples to test on each problem. We also add 40% noise to the training sets by shuffling labels.

The second synthetic dataset is composed of RBF concepts, as an implementation of an existing method [33]. We generate 100 centroids, defined by a random center of 100 features in the range $[0, 1]$, and a standard deviation in the same range. We then generate $1,000$ random examples for each RBF concept. We repeatedly extract training (10%) and test samples (30%) without replacement for each class, 30 times, and compose balanced binary classification problems of each RBF concept vs. rest. We add 40% noise by shuffling training labels.

For real-world experiments, we work with 20newsgroups [34], CIFAR-100 [35], and an ImageNet subset of 500 classes [36][5]. We use the same approach as for RBF problems to sample training and test sets, and to compose binary classification tasks, 30 times.

In our lifelong learning setting, for the synthetic hyperplane dataset we learn the 500 tasks described previously. For our lifelong learner (HRSVM), we first compose an initial knowledge base of 250 tasks. These tasks are learned using an SVM with a linear kernel and $C = 1$. Then we learn the other 250 tasks sequentially. Our transfer forward method requires a threshold on the Kullback-Leibler divergence (KL), which we set to 0.3, and a threshold on the number of nearest-neighbours for transfer (nn), which we set to 2. For transferring backward, we set the parameter Γ in Eq. 5 to 0.01 and ν to the maximal feasible value, $\nu = 2 * min(l_+, l_-)/l$ [31]. For ELLA [24], we set the percentage of latent components as 0.25 of the number of features, after grid-search on the set $\{0.05, 0.10, 0.15, 0.20, 0.25\}$ using a 5% validation set. We select the value of 1 for the sparsity level, after searching on the values $\{0.05, 0.1, 0.2, 0.5, 0.8, 1\}$. For CL [25], we set the similarity threshold to 0.15 after grid-search over the values $\{0.10, 0.15, 0.20, 0.25, 0.30\}$ on a 5% validation set. Finally, for DEN [26] we learn a simple feedforward network with parameters: two hidden layers of 200 and 300 neurons, epoch of 500, batch size of 500, learning rate of 0.001, sparsity for L1 of 0.0001, L2 lambda of 0.0001, group Lasso lambda of 0.001, regularization lambda of 0.5, threshold for dynamic expansion of 0.01, threshold for split and duplication of 0.05 and number of units of expansion of 10, given the large number of tasks. The order of tasks is randomized on each repetition.

The HRSVM setting is similar for the other four datasets: we first learn a target hypothesis by optimizing Eq. 4, and the existing hypotheses are refined by optimizing Eq. 5. A preliminary step trains half of the hypotheses as initial sources. Initial hypotheses for all datasets are trained using an SVM, with $C = 1$. Synthetic RBF tasks are trained with an RBF kernel and $\gamma = 0.1$ to make it subject to refinement. 20newsgroups, CIFAR and ImageNet tasks are trained with RBF kernels and $\gamma = 1/d$, with d the number of features. Parameters for learning target hypotheses are as follows: for synthetic RBF $KL = 0.45$, for 20newsgroups $KL = 0.5$, for CIFAR-100 and for ImageNet $KL = 0.3$. In all cases, the number of nearest neighbours, nn, is set to 2. Hypothesis refinement is performed with the modified ν-SVM in Eq. 5 with the same SVM parameters as their corresponding initial sources, ν equals to the maximal feasible value and $\Gamma = 0.01$ in all cases. The order of the tasks is randomized for each of the 30 repetitions on each dataset. For ELLA,

[4] `http://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html`

[5] `http://image-net.org/download-features`, for 500 classes selected randomly from `http://image-net.org/api/text/imagenet.sbow.obtain_synset_wordlist`

we tune the number of latent components using grid-search on a 5% validation set, with values in $\{0.05, 0.10, 0.15, 0.20, 0.25\}$, as a percentage of the total number of features. For the sparsity level, we select the optimal value from $\{0.05, 0.1, 0.2, 0.5, 0.8, 1\}$. Best values for the percentage of latent components are as follows: synthetic RBF 0.25, 20newsgroups 0.10, CIFAR-100 0.25, ImageNet 0.20. For all datasets, sparsity level is 1. For datasets of more than 200 features, 200 features are first extracted using PCA[6]. For CL we tune the similarity threshold using grid-search on a 5% validation set, with values in $\{0.10, 0.15, 0.20, 0.25, 0.30\}$. Best values for each dataset are as follows: synthetic RBF 0.15, 20newsgroups 0.20, CIFAR-100 0.20, ImageNet 0.20. The order of tasks is also randomized for both ELLA and CL. Finally, for DEN we use the following setting: for synthetic RBF we use a network with two hidden layers of 250 and 200 neurons. For ImageNet we use a network with two hidden layers of 500 and 250 neurons. For 20newsgroups we use two hidden layers of 500 and 250 neurons. For CIFAR-100 we use a network with two hidden layers of $1,500$ and 500 neurons. For the other parameters we use the default values, for all datasets: maximum number of iterations of $5,000$, batch size of 500, learning rate of 0.001, L1 sparsity of 0.0001, L2 lambda of 0.0001, group Lasso lambda of 0.001, regularization lambda of 0.5, threshold for dynamic expansion of 0.1, threshold for split and duplication of 0.5. For the number of units of expansion, we use the default value of 10.

[6]Implementation of ELLA allows up to 200 features.