Smart Task Orderings for Active Online Multitask Learning

Shaoning Pang* Jianbei An † Jing Zhao ‡ Xiaosong Li ‡ Tao Ban §
Daisuke Inoue §Abdolhossein Sarrafzadeh ‡

April 26, 2014

Abstract

This paper promotes active oMTL (i.e., oMTL with task selection) by proposing two smart task ordering approaches: QR-decomposition Ordering and Minimal-loss Ordering, in which the optimal sequence of tasks for oMTL is computed as the training data/tasks are being presented. Our experimental results on four real-world datasets show that the proposed task orderings outperform all existing task ordering approaches to active oMTL.

1 Introduction

Multitask learning (MTL) investigates offline mode machine learning systems that can learn a set of related tasks in one batch [1], yet in real world applications a bunch of tasks often arrive over an extended period of time. In this context, online multitask learning (oMTL) models systems that can learn multiple related tasks in parallel, by sharing common information among these tasks. As compared to traditional online single task learning (oSTL), oMTL achieves often better generalization performance across all tasks than independently learning each task; and effectively utilizing task relateness rather than simply ignoring it makes oMTL tremendously outperform oSTL in many real world applications [2].

In the basic online learning setting, typical oMTL methods do not control the order in which they learn the tasks. However ordering effects exist for oMTL when, given a set of tasks, different ordered sequences of these tasks lead to different learning results. For the same training data in two different sequences, such ordering of tasks allows one to favor the learning more than the learning with random task selection. In the literature, oMTL with task ordering or selection is called active oMTL [3]. The key of active oMTL is task ordering, which is to find from all permutations of tasks the one that produces the best performance.

In this paper, we aim to promote active oMTL by proposing two novel approaches to task ordering for active oMTL. One approach called QR-decomposition ordering is based on comparison of within-task distance of the training data, and the other called Minimal-loss ordering is based on minimizing the loss of prediction over all tasks. To test the effectiveness, we use the two orderings for the training of an existing oMTL algorithm, the empirical results show that the algorithm can learn tasks more efficiently than the random task selection; and systems utilizing these active learning methods all obtain a clear reduction of necessary training data for achieving a particular level of performance.

The rest of paper is organized as follows: Section 2 introduces related work and overview of our methods. Section 3 presents the proposed QR-decomposition and Minimal-loss task ordering methods, and also explains their differences to existing task ordering methods for active oMTL. In Section 4, we describe our experimental setup, compare proposed methods with existing task ordering methods for active oMTL, and present experimental results. Finally, we conclude the paper with a discussion and an outlook on further extensions in Section 5.

2 Related Work

.

In literature, oMTL has been researched for new algorithm developments mostly in extending online STL (oSTL) to oMTL or deriving from batch MTL to oMTL.

Consider extension of oSTL to oMTL, precise characterization of task relatedness differs how relevant information across the multiple related tasks is being utilized for individual task learning, thus is the key of the research. Dekel et al. [4] employed a global cumulative loss function to model the relatedness among multiple tasks, and extended the oSTL Perceptron [5], to three oMTL algorithms, which includes finite-horizon multitask Perceptron, infinite-horizon multitask Percep-

*Department of Computing, Unitec Institute of Technology, New Zealand (ppang@unitec.ac.nz)

†Department of Mathematics, The University of Auckland, New Zealand. (j.an@auckland.ac.nz)

‡Department of Computing, Unitec Institute of Technology, New Zealand.

§National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

tron, and implicit update multitask Perceptron. Alternatively, Cavallanti et al. [2] used various types of regularization to characterize the relationship among different tasks, and proposed kernel based and matrix based multitask Perceptron. Saha et al. [6] invented an adptive task-relationship matrix to specify the relatedness among a set of tasks, and proposed several oMTL algorithms, including LogDet divergence-based online algorithm, von-Neumann divergence-based online algorithm, and covariance-based online algorithm. All these oMTL algorithms have been demonstrated significantly outperforming traditional oSTL, however most of them are only applicable to classification tasks (not regression) and rely exclusively on perceptron learning.

Recently, Ruvolo et al. considered developing oMTL from batch MTL. They developed firstly an oMTL algorithm, called Efficient Lifelong Learning Algorithm (ELLA)[7], based on the batch GO-MTL [8]. ELLA conducts efficient oMTL by maintaining a sparsely shared basis for all task models, transferring knowledge from the basis to learn each new task, and refining the basis continuously to maximize performance across all tasks. Later, they developed ELLA-SVD [9] based on MTL-SVD, a batch MTL built on the dictionary learning algorithm K-SVD [10]. This reduces the computational complexity of ELLA owing to the more effiecient K-SVD updating strategy. Their further oMTL developments include ELLA Incremental and ELLA Dual Update, where ELLA Incremental adds an incremental update onto the original ELLA, and ELLA Dual Update combines ELLA-SVD and ELLA Incremental to obtain lower computational cost. It is worth noting that all the above oMTLs derived from batch MTL handle both classification and regression tasks and they do not rely on perceptron learning at all.

These aforementioned oMTL methods, derived on above oSTL or batch MTL, are counted as a type of passive learning in that they do not control the order in which they learn the tasks. In contrast, active learning [11] aims to actively order training data so as to maximize learning performance across all tasks. In the literature of oMTL, active oMTL has been addressed in recent years. Saha et al. [12] proposed an adaptive framework of active oMTL based on instance ordering, in which an adaptive relationship matrix was used to quantify the informativeness of an arriving instance across all tasks, and the most informative instance was always chosen to be learned next. Ruvolo et al. [3] developed another two active oMTL algorithms based on task ordering: InfoMax and Diversity. InfoMax is based on information maximization and the next task is selected according to the expected information gain about the shared basis; and Diversity is based on model performance and the next task is chosen according to the worse case fit of shared basis to each candidate task. The difference of these two previous works is, Saha optimized oMTL by ordering training instances, whereas Ruvolo conducted tasks ordering. All these methods nevertheless obtain the reduction of necessary training data for achieving a certain level of performance.

This paper derives mathematically two novel task ordering approaches: QR-decomposition Ordering and Minimal-loss Ordering. The QR-decomposition Ordering measures the within-task distance of the training data, and selects the next task with the shortest within-task distance. The Minimal-loss Ordering calculates the preditive loss of the learned model, and chooses the next task with the minimal loss. The proposed task orders are more effective than all exisiting task ordering approaches in enabling oMTL (e.g., ELLA) to achieve better performance.

3 Proposed Task Ordering Methods

In the setting of oMTL, usually the training instance or the training task arrives one-at-a-time, in such cases, the learner has no control over the order in which learning tasks are presented. But occasionally, a chunk of training data which consists of multiple related tasks are received in a batch, in such case, obviously the learner has the opportunity to actively order these tasks for learning, so as to improve the learning efficiency. We now investigate task ordering methods for active oMTL.

We assume the problem of task ordering in the following setting, in which, at the mth iteration, the learner receives training data for n tasks, which are indexed as $\{T_1, T_2, \cdots, T_n\}$. Using some strategies of task ordering, we aim to make these n tasks be learned in a particular order, which means a new permutation of $\{T_1, T_2, \cdots, T_n\}$, so as to achieve more learning efficiency than random task ordering. For the rest of the paper, parenthetical superscripts denote valuables related to a particular task, for example, $X^{(T_1)}$ and $y^{(T_1)})$ are related to task T_1.

Usually the criteria for active learning focus on choosing the most uncertain [13] or the most informative instances [14]. Whereas, in curriculum learning [15], the easiest instances are suggested to be learned firstly, then harder instances are to be incrementally processed. Alternatively, to achieve the best performance across multiple tasks, we derives below two formulated strategies to choose the most suitable next task to learn from a set of candidate tasks.

3.1 QR-decomposition Ordering

The QR decomposition (or QR factorization) [16] of a real matrix has already been utilized in many applications, especially in linear least squares problems. The QR decomposition of matrix A is defined as follows: $A = QR$, where Q refers to an orthogonal matrix, and R denotes an upper triangular matrix. In this paper, we apply QR decomposition to active oMTL, and propose a task ordering method based on comparing the within-task distance of the training data, in which QR decomposition of centroid matrix of the training data is employed.

Suppose that at the mth iteration we receive training data $(X_{new}^{(t)}, y_{new}^{(t)})$, t denotes task $t \in \{T_1, T_2, \cdots, T_n\}$. Given a data matrix $X_{old}^{(t)} = [A_1^{(t)}, \cdots, A_k^{(t)}] \in \mathbb{R}^{d\times n}$ with $A_i^{(t)} \in R^{d\times n_i}$ (write $X_{old}^{(t)} = 0$ when t is new), where $A_i^{(t)}$ represents the previously received training data for task t, n_i denotes the number of instances contained in $A_i^{(t)}$, and d is feature dimension. Suppose $C^{(t)} = Q^{(t)}R^{(t)}$ is the QR decomposition of the centroid matrix $C^{(t)} = [m_1^{(t)}, \cdots, m_k^{(t)}]$ and $H_w^{(t)} = [H_1^{(t)}, \cdots, H_k^{(t)}]$, where $H_i^{(t)} = [A_i^{(t)} - m_i^{(t)} e_i^T]$ with $e_i \in (1, \cdots, 1)^T \in \mathbb{R}^{n_i}$.

Let $X_{new}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \cdots, x_{u_t}^{(t)}]$ as column vectors. We define

$$w^{(t)}(x_i) = \frac{\|(Q^{(t)})^T(x_i - m_j)\|^2}{n_j + 1} \tag{3.1}$$

or

$$w^{(t)}(x_i) = \frac{\|(H_w^{(t)})^T(I - Q^{(t)}(Q^{(t)})^T)x_i)\|^2}{\|(I - Q^{(t)}(Q^{(t)})^T)x_i\|^2} \tag{3.2}$$

accordingly, as x_i lies in the jth class of $X_{old}^{(t)}$ or a new class. Set

$$w^{(t)}(X_{new}^{(t)}) = \frac{1}{u_t}\sum_{i=1}^{u_t} w^{(t)}(x_i). \tag{3.3}$$

Suppose at the mth iteration we receive training data $(X_{new}^{(t)}, y_{new}^{(t)})$ and $(X_{new}^{(t')}, y_{new}^{(t')})$.

Suppose, moreover that $w^{(t)}(X_{new}^{(t)}) \le w^{(t')}(X_{new}^{(t')})$. Then the effective ordering for optimizing the models to minimize the loss over all tasks is first task t and then task t' when update the model.

The criterion on QR-decomposition Ordering to choose the next task is summerized as,

$$t_{next} = \underset{t \in \{T_1, T_2, \cdots, T_n\}}{argmin} w^{(t)}(X_{new}^{(t)}), \tag{3.4}$$

where $w^{(t)}(X_{new}^{(t)}) = \frac{1}{u_t}\sum_{i=1}^{u_t} w^{(t)}(x_i)$.

3.2 Minimal-loss Ordering

Recently, a shared basis $\boldsymbol{L}$ were used in MTL and oMTL for modelling task relatedness and sharing useful information among a set of learning tasks [7, 8]. In these cases, the model of task t is represented as a parameter vector $\theta^{(t)}$ that is a linear combination of the columns of shared basis $\boldsymbol{L}$ according to the weight vector $s^{(t)} : \theta^{(t)} = \boldsymbol{L}s^{(t)}$. The mechanism of intelligently choosing the next task for this proposed Minimal-loss Ordering method is based on calculating the predictive loss of the learned model, and chooses the next task with the minimal loss.

Suppose that at the mth iteration we receive training data $(X_{new}^{(t)}, y_{new}^{(t)})$. Define $X^{(t)} = [X_{old}^{(t)}\, X_{new}^{(t)}]$ or $X^{(t)} = X_{new}^{(t)}$, $y^{(t)} = (y_{old}^{(t)}; y_{new}^{(t)})$ or $y^{(t)} = y_{new}^{(t)}$ accordingly as t is an old or new task. Write $X^{(t)} = [x_1^{(t)}, x_2^{(t)}, \cdots, x_{n_t}^{(t)}]$ as column vectors. Let

$$F^{(t)}(\theta) = \frac{1}{n_t}\sum_{i=1}^{n_t} \mathcal{L}(f(x_i^{(t)}; \theta), y_i^{(t)}), \tag{3.5}$$

$$D^{(t)} = \frac{1}{2}\nabla^2_{\theta,\theta}(F^{(t)}(\theta))|_{\theta=\theta^{(t)}}, \tag{3.6}$$

where $\mathcal{L}$ is a known loss function and f is the prediction function. Let

$$\ell(L, s, \theta, D) = \mu\|s\|_1 + \|\theta - Ls\|_D^2 \tag{3.7}$$

and

$$\begin{array}{lcl} G(L) & = & \hat{g}_m(L) \\ & = & \lambda\|L\|_F^2 + \frac{1}{T}\sum_{i=1}^{T} \ell(L, s^{(t)}, \theta^{(t)}, D^{(t)}) \end{array}. \tag{3.8}$$

There are two steps to update the model: compute $s^{(t)}$ and update L:

Step (1). Compute $s^{(t)}$.
Firstly compute $\theta^{(t)}$ such that

$$\theta^{(t)} = \text{ arg min }_\theta F^{(t)}(\theta); \tag{3.9}$$

then compute

$$D^{(t)} = \frac{1}{2}\nabla^2_{\theta,\theta}(F^{(t)}(\theta))|_{\theta=\theta^{(t)}} \tag{3.10}$$

and re-initialize L; and then compute $s^{(t)}$ such that

$$s^{(t)} = \text{ arg min }_s \ell(L_m, s, \theta^{(t)}, D^{(t)}). \tag{3.11}$$

Step (2). Update L.
Set

$$\nabla_L G(L) = \mathbf{0}$$

and solve for L, called $L_{m+1}^{(t)}$.

Proposition 3.1. *Suppose at the mth iteration we receive training data* $(X_{new}^{(t)}, y_{new}^{(t)})$ *and* $(X_{new}^{(t')}, y_{new}^{(t')})$. *Suppose, moreover that*

$$G(L_{m+1}^{(t)}) \leq G(L_{m+1}^{(t')}).$$

Then the effective ordering for optimizing the models to minimize the lost over all tasks is first task t and then task t′ when update the model.

Proof. By [7, (1)], the objective function $e_T(L) \sim G(L)$ and so

$$e_T(L_{m+1}^{(t)}) \leq e_T(L_{m+1}^{(t')}) \iff G(L_{m+1}^{(t)}) \leq G(L_{m+1}^{(t')}).$$

□

The strategy for Minimal-loss Ordering method to choose the next task is as follows:

$$t_{next} = \underset{t \in \{T_1, T_2, \cdots, T_n\}}{argmin} G(L_{m+1}^{(t)}), \tag{3.12}$$

where

$$G(L) = \lambda \|L\|_F^2 + \frac{1}{T}\sum_{i=1}^{T} \ell(L, s^{(t)}, \theta^{(t)}, D^{(t)}).$$

4 Experiment and Result

We evaluate the proposed QR-decomposition and Minimal-loss Ordering method by comparing them against two existing task ordering approaches: InfoMax and Diversity [3]. We apply these four methods to an existing lifelong learning algorithm (ELLA) [7] to assess their performance. The same as in [3], we conduct oMTL experiments on dataset Facial Expression Recognition, Land Mine Detection, London Schools and Computer Survey, respectively.

For each dataset, we randomly split data for 20 times in its predefined ratio of training to testing. We repeat active oMTL experiment on generated data split for 1,000 times to smooth out variability. The average results are reported for each task ordering method. In parameters setting, we follow [3] to maximize performance on the evaluation tasks averaged over all the task ordering methods, and use a grid-search approach to select the value of the parameter k in $\{1, 2, \cdots, 10\}$, and the ridge term Γ from the set $\{e^{-5}, e^{-3}, e^{-1}, e^{1}\}$. The value of λ and μ are determined as e^{-5} and 1 respectively through cross validation experiments.

For performance evaluation, we set observed oMTLs to achieve and maintain a certain level of performance, then estimate, as in [3], how many less tasks (in terms of the percentage to the total number of tasks) are demanded as compared to the number of tasks required for oMTL on random task order. The oMTL performance for classification tasks is measured by the area under the ROC curve (AUC), and regression tasks by the negative root mean squared error (-rMSE). A positive score on % *Less Tasks Required* indicates the task ordering approach has higher learning efficiency than random task ordering, a negative score displays that it is less efficient, and a score of 0 reveals that it has no improvement in learning efficiency.

We compare the proposed task ordering methods: (a) QR-decomposition ordering (QR) and (b) Minimal-loss ordering (Minimal-loss) with two existing methods: (c) InfoMax and (d) Diversity. We measure the less tasks required during the learning process, the average less tasks required, and the final less tasks required, as compared to random task ordering.

Figure 1 shows the oMTL experimental results of less tasks required during the learning process for four different task ordering methods on four real-world datasets. As we can see, all four task ordering methods achieve more or less learning efficiency gain over the random task ordering, which indicates oMTL with task ordering is always more efficiently than that without ordering. Particularly, the plots from QR and Minimal-loss are shown both on the top of the plots from InfoMax and Diversity for all datasets, which follows that the proposed task ordering methods have completely dominated the existing InfoMax and Diversity methods in all datasets. However, those top two methods are very competitive to each other. The QR method wins on the regression oMTL of London School dataset and the classification oMTL of Facial Expression dataset, whereas the Minimal-loss approach is more efficient for the regression oMTL of Computer Survey dataset and the classification oMTL of the Land Mine dataset, respectively. Further investigation is needed to find out whether a certain dataset characteristic will impact the optimal task ordering in oMTL.

Table 1 depicts the results, which are measured in the percent less tasks required for oMTL with task ordering, and averaged across all performance levels. Table 2 presents the less tasks required by task ordering at the highest performance level. In these tables, the mean and standard deviations are reported, numbers in bold represent the best performance on the column dataset. As we can see, the results of these two tables reveal the same behavior as that of Figure 1 that, InfoMax and Diversity were dominated in all four datasets by the proposed QR and Minimal-loss ordering.

Additionally, we calculate both in Table 1 and Table 2 the performance difference between the best proposed method and the best existing methods. As seen, the proposed task ordering methods are in general over

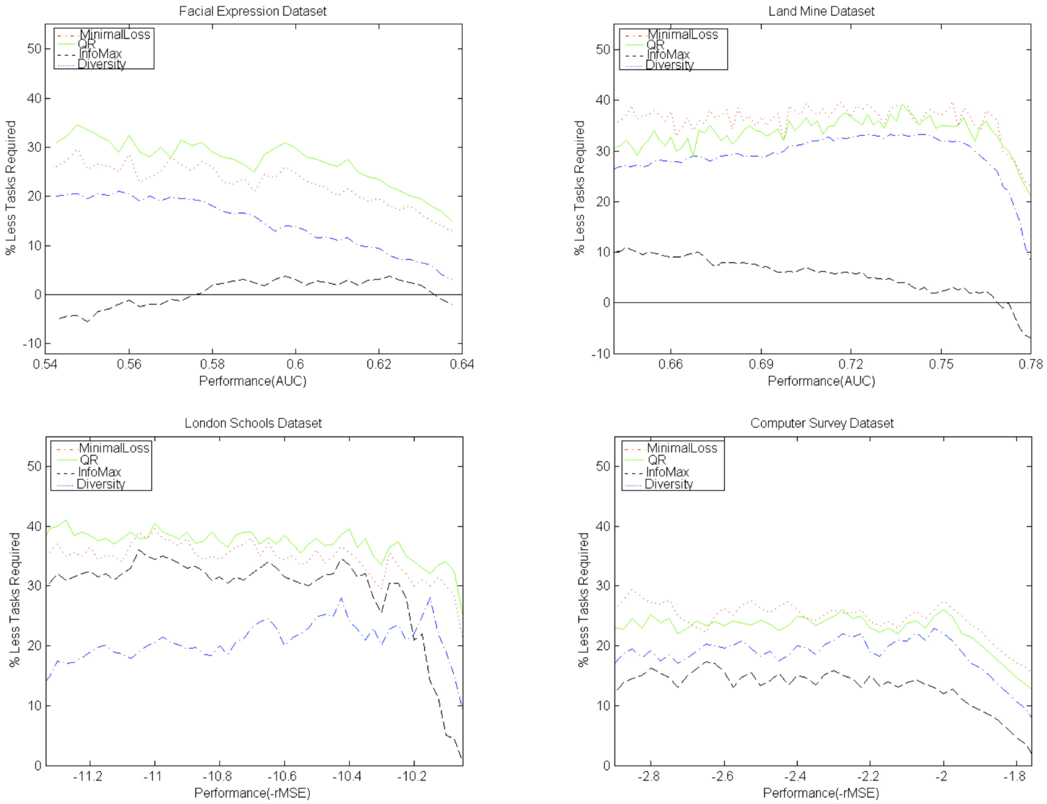

Figure 1: The results of task ordering on oMTLs. Each plot shows the accuracy achieved by each method versus the oMTL efficiency (in terms of the number of tasks, and in comparison to random task ordering).

40% more efficient that the existing methods for all performance level experiments. But, this superiority increases to above 208% when the highest performance level experiment is counted. The maximum superiority of QR to Diversity is close to four times (i.e., 387.1%).

5 Conclusions

This paper presents two novel approaches to task ordering for active oMTL: QR-decomposition Ordering and Minimal-loss Ordering. Recall in literature the criteria for active learning include, (1) high diversity, which was interpolated as the most uncertain [13, 17] or the most informative [14] instances ; and (2) low risk, which was implemented in [18, 19, 15] as selecting the easiest instances to be learned firstly. The diversity measure is on training data, whereas the risk evaluation is from the learning function. In principle, QR-decomposition ordering performs a diversity criterion that measures the tasks diversity/discrimination on within-task data distance and selects always tasks with less within-task distance. Minimal-loss ordering specially combines the diversity and risk criteria, which evaluates both prediction loss and task relatedness, and chooses always task with the minimum of prediction loss and task relatedness.

The orders derived in this paper present a generic task ordering approach independent to any particular oMTL algorithms, classification or regression, in spite that we used only ELLA for active oMTL for performance evaluation. In practice, they can be incorporated into any individual oMTL for active online multitask learning. Our experimental results on four real-world datasets show that the proposed task orderings significantly promote the learning efficiency of oMTL and outperform all existing task ordering approaches. It is worth noting that neither QR-decomposition nor Minimal-loss ordering dominates the active oMTL experiments on all four datasets. This indicates that how to select the best task ordering method for each individual dataset is a difficult issue and in practice it is determined by many factors such as characteristics of datasets, task ordering criteria, task relatedness model, and updating rules of oMTL, etc. For future work, we will investigate task ordering key factors to datasets characteristics and oMTL functions, respectively.

Table 1: Average Less Tasks Required over all performance levels.

2*Method	Average %Less Tasks Required (Standard Deviation)			
	Facial Expression	Land Mine	London School	Computer Survey
Minimal-loss	22.7(±3.4)	**36.1(±3.2)**	34.5(±4.3)	**24.5(±2.9)**
QR	**27.1(±3.8)**	33.5(±4.4)	**37.0(±3.6)**	22.6(±3.1)
Diversity	14.6(±5.1)	29.4(±4.1)	21.0(±3.1)	18.5(±3.4)
InfoMax	0.5(±2.6)	5.1(±3.7)	29.8(±6.8)	12.9(±3.8)
Diff.	+85.6%	+22.8%	+24.2%	+32.4%
Average	+41.3%			

Table 2: Less Tasks Required by task ordering at the highest performance level.

2*Method	Final %Less Tasks Required (Standard Deviation)			
	Facial Expression	Land Mine	London School	Computer Survey
Minimal-loss	13.0(±3.2)	**23.1(±3.5)**	21.2(±3.3)	**15.5(±3.9)**
QR	**15.1(±3.5)**	21.2(±3.9)	**25.0(±3.8)**	12.8(±3.3)
Diversity	3.1(±4.1)	8.2(±3.1)	9.0(±3.3)	8.3(±3.2)
InfoMax	-2.2(±2.9)	-6.9(±3.5)	0.5(±3.8)	2.0(±3.6)
Diff.	+387.1%	+181.7%	+177.8%	+86.7%
Average	+208.3%			

References

[1] S. Thrun and J. O'Sullivan, "Discovering structure in multiple learning tasks: The tc algorithm," in *the 13th international conference on Machine learning (ICML 1996)*, 1996.

[2] G. Cavallanti and N. Cesa-Bianchi, "Linear algorithms for online multitask classification," in *The 21st Annual Conference on Learning Theory (COLT 2008)*, 2008.

[3] P. Ruvolo and E. Eaton, "Active task selection for lifelong machine learning," in *the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[4] O. Dekel, P. M. Long, and Y. Singer, "Online multitask learning," in *The 19th Annual Conference on Learning Theory (COLT 2006)*, 2006.

[5] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–407, 1958.

[6] A. Saha, P. Rai, H. DauméIII, and S. Venkatasubramanian, "Online learning of multiple tasks and their relationships," in *the 14th International Conference on Artificial Intelligence and Statistics*, 2011.

[7] P. Ruvolo and E. Eaton, "Ella: An efficient lifelong learning algorithm," in *the 30th International Conference on Machine Learning(ICML 2013)*, 2013.

[8] A. Kumar and H. Daumé III, "Learning task grouping and overlap in multi-task learning," in *the 29th International Conference on Machine Learning (ICML 2012)*, 2012.

[9] P. Ruvolo and E. Eaton, "Online multi-task learning based on k-svd," in *the ICML 2013 Workshop on Theoretically Grounded Transfer Learning*, 2013.

[10] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[11] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, Computer Sciences Technical Report 1648, 2009.

[12] A. Saha, P. Rai, H. Daumé III, and S. Venkatasubramanian, "Active online multitask learning," in *the 27th International Conference on Machine Learning(ICML 2010)*, 2010.

[13] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[14] D. J. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, pp. 590–604, 1992.

[15] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Press, 2011.

[16] J. W. Daniel, W. B. Bragg, L. Kaufman, and G. W. Steward, "Reorthogonalization and stable algorithms for updating the gram-stchmidt qr factorization," *MATHEMATICS OF COMPUTATION*, vol. 30, no. 136, pp. 772–795, 1976.

[17] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[18] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *the 26th International Conference on Machine Learning (ICML 2009)*, 2009.

[19] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," *In Advances in Neural Information Processing Systems*, vol. 23, pp. 1189–1197, 2010.