Multiclass Domain Generalization

Aniket Anand Deshmukh
Department of EECS
University of Michigan
Ann Arbor, MI 48109, USA
aniketde@umich.edu

Srinagesh Sharma
Department of EECS
University of Michigan
Ann Arbor, MI 48109, USA
srinag@umich.edu

James W. Cutler
Department of Aerospace Engineering
University of Michigan
Ann Arbor, MI 48109, USA
jwcutler@umich.edu

Clayton Scott
Department of EECS
University of Michigan
Ann Arbor, MI 48109, USA
clayscot@umich.edu

Abstract

Domain generalization is the problem of assigning labels to an unlabeled data set, given several similar data sets for which labels have been provided. In this work, we consider a kernel-based algorithm for domain generalization that was developed in the binary setting. In particular, we show that the generalization error bound for this algorithm extends to the multi-class setting. Our main contribution includes generalization error bound and scalable implementation of the approach. We demonstrate improved performance with respect to a pooling strategy on four data sets.

1 Introduction

Transfer learning, domain adaptation, and weakly supervised learning all have the goal of generalizing without access to conventional labeled training data. One particular form of transfer learning that has garnered increasing attention in recent years is *domain generalization* (DG) [2, 3]. In this setting, the learner is given unlabeled data to classify, and must do so by leveraging labeled data sets from similar yet distinct classification problems. In other words, label training data drawn from the same distribution as the test data are not available, but are available from several related tasks. We use the terms “task" and “domain" interchangeably throughout this paper.

Applications of DG are numerous. For example, each task may be a prediction problem associated to a particular individual (e.g., handwritten digit recognition), and the variation between individuals accounts for the variation among the data sets. Domain generalization is needed when a new individual appears, and the only training data come from different subjects.

As another application, below we consider DG for determining the orbits of microsatellites, which are increasingly deployed in space missions for a variety of scientific and technological purposes. Because of randomness in the launch process, the orbit of a microsatellite is random, and must be determined after the launch. Furthermore, ground antennae are not able to decode unique identifier signals transmitted by the microsatellites because of communication resource constraints and uncertainty in satellite position and dynamics. More concretely, suppose c microsatellites are launched together. Each launch is a random phenomenon and may be viewed as a task in our framework. One can simulate the launch of microsatellites using domain knowledge to generate highly realistic training data (feature vectors of ground antennae RF measurements, and labels of satellite ID). One can then

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

transfer knowledge from the simulated training data to label (identify the satellite) the measurements from a real-world launch with high accuracy.

Several approaches to domain generalization have been proposed, including complexity regularization that adapts to the variability of the sampling distribution on tasks [2, 3, 20, 26], learning this sampling distribution directly [5], task matching by optimal transport [7], learning a feature extractor that puts all tasks in a common feature space [25, 24, 22, 27, 14, 12], and using the marginal distribution from which a feature vector is drawn as a feature itself for label prediction [4].

Our work builds on the approach of [4], which develops a kernel-based framework for DG. We review this framework below, and extend their analysis, which addresses the setting of binary labels, to the multiclass setting. While several aspects of the original analysis in [4] carry over to the multiclass case, others do not. In particular, we use an extension of the contraction lemma for Rademacher complexity of Lipschitz loss classes to prove the generalization error bound [21].

A few existing approaches to DG address the multiclass case [7, 13, 23, 18, 15, 9]. Most of these works rely on neural networks and none have statistical performance guarantees. We also note that the approach of [4] can be combined with feature extraction approaches, as was done in [24], and the same is true of our multiclass extension.

Our contributions include: (1) Extending the kernel-based approach to DG from [4] to multiclass DG, (2) Extending the analysis of [4] to multiclass, (3) a scalable implementation based on random Fourier features, and (4) experimental demonstration of the method compared to a pooling approach.

In section 2 we formally state the DG problem and in section 3 we describe the kernel-based learning algorithm. Section 4 contains our theoretical analysis, and experimental results appear in section 5.

2 Formal Problem Statement

Let $\mathcal{X}$ be the feature space and $\mathcal{Y}$ the label space with $|\mathcal{Y}| = c$. Denote by $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$ the set of probability distributions on $\mathcal{X}\times\mathcal{Y}$, $\mathcal{P}_{\mathcal{X}}$ the set of probability distributions on $\mathcal{X}$, and $\mathcal{P}_{\mathcal{Y}|\mathcal{X}}$ the set of conditional distributions of Y given X. Furthermore, let μ be a probability measure on $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$, i.e., whose realizations are distributions on $\mathcal{X}\times\mathcal{Y}$.

With the above notations, DG is defined as follows. We are given training data sets $S_i = ((X_{ij}, Y_{ij}))_{1\le j\le n_i}$ such that $(X_{ij}, Y_{ij}) \sim P_{XY}^i$ and $P_{XY}^i \sim \mu$. The test data set is $S^T = ((X_j^T, Y_j^T))_{1\le j\le n_T}$ such that $(X_j^T, Y_j^T) \sim P^T$ and $P^T \sim \mu$. We assume all (X, Y) pairs are drawn iid from their respective distributions, and that $P_1, \ldots, P_N, P^T$ are iid from μ. The Y_j^T are not visible to the learner, and the goal is to accurately predict $(Y_j^T)_{1\le j\le n_T}$. For any predicted estimate of a label $\hat{Y}$, the accuracy is evaluated using a loss function $l : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$. For greater flexibility in the multiclass case ($c > 2$), the label space for prediction is relaxed to $\mathbb{R}^c$ and a surrogate loss function $l : \mathbb{R}^c \times \mathcal{Y} \to \mathbb{R}_+$ is employed.

According to the approach in [4], DG is cast as a supervised learning problem where the input to the classifier is the extended feature space $\mathcal{P}_{\mathcal{X}}\times\mathcal{X}$. A decision function is a function $f : \mathcal{P}_{\mathcal{X}}\times\mathcal{X} \to \mathbb{R}^c$ that predicts $\hat{Y}_j^T = f(\hat{P}_X^T, X_j^T)$, where $\hat{P}_X$ is the associated empirical distribution. The decision function can be separated into its components $f = \begin{bmatrix} g_1 & g_2 & \cdots & g_c \end{bmatrix}$ such that $g_m : \mathcal{P}_{\mathcal{X}}\times\mathcal{X} \to \mathbb{R}$, for $m = 1, 2, \ldots c$. We define the empirical error on test sample with sample size n_T as

$$\hat{\varepsilon}(f, n_T) = \frac{1}{T}\sum_{i=1}^{n_T} \ell(f(\hat{P}_X^T, X_i^T), Y_i^T), \tag{1}$$

and by denoting $\tilde{X} = (P_X, X)$, the generalization error of a decision function with respect to loss ℓ as

$$\varepsilon(f) = E_{P_{XY}^T\sim\mu} E_{(X^T,Y^T)\sim P_{XY}^T} \ell(f(P_X^T, X^T), Y^T) = E_{P_{XY}^T\sim\mu} E_{(X^T,Y^T)\sim P_{XY}^T} \ell(f(\tilde{X}^T), Y^T). \tag{2}$$

The goal of DG is to learn an f that minimizes this generalization error.

Remarks: (1) Although the generalization error assumes the predictor has access to P_X, at training time as well as at test time P_X is only known through the empirical marginal $\hat{P}_X$. (2) Despite the similarity to standard classification in the infinite sample case, the learning task here is different, because the realizations $(\tilde{X}_{ij}, Y_{ij})$ are neither independent nor identically distributed. (3) Examples of loss functions l can be found in Lee et. al [17], Crammer and Singer [8], Weston and Watkins [32]. For detailed discussion on different multiclass loss functions and their general forms see [10, 31, 16, 29].

3 Kernel Based Learning Algorithm

The goal of predicting an optimal classifier on the extended feature space can be solved using kernel based algorithms. For a (symmetric positive definite) kernel k, let H_k denote its associate reproducing kernel Hilbert space. Let $\bar{k} : (\mathcal{P}_X \times \mathcal{X}) \times (\mathcal{P}_X \times \mathcal{X}) \to \mathbb{R}$ be a symmetric and positive definite kernel on $\mathcal{P}_X \times \mathcal{X}$, whose construction will be described below. Let ℓ be a loss function. Further let $\hat{P}^i_X$ be the finite sample empirical distribution for sample S_i corresponding to X_{ij}, and let $\tilde{X}_{ij} = (\hat{P}^i_X, X_{ij})$ be the extended data point. We will find a decision function $f \in H^c_{\bar{k}} := H_{\bar{k}} \times \cdots H_{\bar{k}}$ (c times) and has components $g_l \in H_{\bar{k}}, l = 1, 2, ...c$, i.e., $f = \begin{bmatrix} g_1 & g_2 & \cdots & g_c \end{bmatrix}$. Define

$$\hat{f}_\lambda = \arg\min_{f \in H^c_{\bar{k}}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(f(\tilde{X}_{ij}), Y_{ij}) + \lambda r(f), \tag{3}$$

as the empirical estimate of the optimal decision function. One could define the regularizer $r(f)$ as $r(f) = \sum_{m=1}^{c} \|g_m\|^2_{H_{\bar{k}}}$. The kernel $\bar{k}$ can be constructed from 3 other kernels k_x, k'_x and κ. Let k_x and k'_x be kernels on $\mathcal{X}$. The so-called kernel mean embedding is the mapping $\Phi : \mathcal{P}_X \to H_{k'_x}$,

$$\Phi(P) = \int_X k'_x(x, \cdot) dP. \tag{4}$$

Let κ be a kernel-like function on $\Phi(\mathcal{P}_X)$, such as the Gaussian-like function $\kappa(\Phi(P^1_X), \Phi(P^2_X)) = \exp(-\|\Phi(P^1_X) - \Phi(P^2_X)\|^2 / 2\sigma^2_\kappa)$. Then $\kappa(\Phi(\cdot), \Phi(\cdot))$ is a kernel on $\mathcal{P}_X$ [6], and we can now define the kernel on the extended feature space via as a product kernel

$$\bar{k}((P^1_x, X_1), (P^2_x, X_2)) = \kappa(\Phi(P^1_X), \Phi(P^2_x)) k_x(X_1, X_2). \tag{5}$$

The empirical estimate of Φ can be computed for $\{X_{ij}\}_{1 \le j \le n}, X_{ij} \sim P^i_X$ as $\Phi(\hat{P}^i_X) = \frac{1}{n} \sum_j k'_x(X_{ij}, \cdot)$. The algorithm associated with the minimizer is similar to multiclass extensions of SVMs such as those presented in [17] applied over the extended feature space. The representer theorem applies in modified form for the optimization (3) over kernels defined over the extended feature space.

4 Generalization Error Analysis

We make following assumptions to analyze generalization error. For any kernel k, $\phi_k(x) := k(\cdot, x) \in H_k$ denotes the canonical feature map, and $\mathbb{B}_k(R)$ refers to the ball of radius R in H_k.

A I The loss function $\ell : \mathbb{R}^c \times \mathcal{Y} \to R$ bounded by B_ℓ, and is L_l-Lipschitz in the first variable: For all y, $|l(T_1, y) - l(T_2, y)| \le L_l \|T_1 - T_2\|$ for $T_1, T_2 \in \mathbb{R}^c$.

A II Kernels k_x, k'_x, κ are bounded by $B^2_k, B^2_{k'}, B^2_\kappa$ respectively.

A III The canonical feature map $\phi_\kappa : H_{k'_x} \to H_\kappa$ is α-Hölder continuous, i.e., $\forall a, b \in \mathbb{B}_{k'_x}(B_{k'})$:

$$\|\phi_\kappa(a) - \phi_\kappa(b)\| \le L_\kappa \|a - b\|^\alpha.$$

The above assumptions are similar to those presented in [4] translated to multiclass data. Condition **A III** holds with $\alpha = 1$ when κ is the Gaussian-like kernel on $H_{k'_x}$. Using the stated assumptions we shall now develop generalization error bounds for multiclass DG. To generalize the analysis, an extension of Talagrand's lemma for bounding the Rademacher complexity is needed. Such an extension was provided by [19] and [21].

Lemma 1. ***(Vector Valued Talagrand's Contraction Lemma)*** *[21] Let $\mathcal{F}$ be a class of functions from $\mathcal{X} \to \mathbb{R}^c$. Let $\{\mu_i\}_{i=1}^N$ and $\{\sigma_{ij}\}_{i=1,j=1}^{N,c}$ be two sets of independent Rademacher random variables. If $\psi : \mathbb{R}^c \to \mathbb{R}$ is L-Lipschitz under $\|\cdot\|_p$ where $p \ge 2$, then*

$$\mathbb{E}_\mu \big[\sup_{f \in \mathcal{F}} \sum_{i=1}^{N} \mu_i \psi(f(x_i)) \big] \le \sqrt{2} L \mathbb{E}_\sigma \big[\sup_{f \in \mathcal{F}} \sum_{i=1}^{N} \sum_{j=1}^{c} \sigma_{ij} g_j(x_i) \big]$$

For simplicity's sake, we assume that $n_i = n$ to state the generalization error bound.

Theorem 1. ***(Estimation error control)*** *Assuming that conditions* ***A I - A III*** *hold then for any $R > 0$, with probability at least $1 - \delta$:*

$$\sup_{f\in\mathbb{B}_{\bar{k}}(R)} |\widehat{\varepsilon}(f) - \varepsilon(f)| \le L_l L_\kappa R B_k c(B_{k'})^\alpha \left(\sqrt{\frac{2\log\frac{2N}{\delta}}{n}} + \sqrt{\frac{1}{n}} + \frac{4\log\frac{2N}{\delta}}{3n}\right)^\alpha + \frac{4\sqrt{2}RL_lB_kB_\kappa c}{\sqrt{N}} + B_\ell\sqrt{\frac{\log 2\delta^{-1}}{2N}}$$

Proof Sketch Let $\mathcal{E}(f) = |\widehat{\varepsilon}(f) - \varepsilon(f)|$.

$$\begin{aligned}\sup_{f\in\mathbb{B}_{\bar{k}}(R)} \mathcal{E}(f) &\le \sup_{f\in\mathbb{B}_{\bar{k}}(R)} \left|\widehat{\varepsilon}(f) - \frac{1}{Nn}\sum_{i=1}^{N}\sum_{j=1}^{n}\ell(f(\tilde{X}_{ij}), Y_{ij})\right| + \sup_{f\in\mathbb{B}_{\bar{k}}(R)} \left|\frac{1}{Nn}\sum_{i=1}^{N}\sum_{j=1}^{n}\ell(f(\tilde{X}_{ij}), Y_{ij}) - \varepsilon(f)\right| \\ &= (I) + (II)\end{aligned}$$

Term (I) is bounded by application of Lipschitz continuity of l, union bounds for tasks and classes over f and through Hölder continuity in assumption **A III**. Bounding the term (II) is similar to bounding term (II) in Theorem 5 in [4] with modifications for multi-class loss. In addition, the modified Talagrand's lemma 1 is applied to bound the Rademacher complexity [21].

5 Results

We test the proposed algorithm on 4 multiclass datasets and compare it with pooling, where data from all the tasks are pooled together to learn one single classifier. Datasets description are given below and summary is in Table 1.

Dataset	Training Tasks	Test Tasks	Examples Per Task	Classes
Synthetic	80	20	100	10
Satellite	400	100	77-165	3
HAR	20	10	300	6
MNIST-MOD	80	20	100	10

Table 1: Summary of Datasets

Synthetic Dataset: Features for synthetic data are drawn from the unit square. Based on one of the dimensions, the data are labeled from 0 to 10, e.g., if the feature value is between 0 and 0.1, then it's labeled as 1, if it's in between 0.1 and 0.2, then it's labeled as 2, and so on. After that, the feature vectors are rotated clockwise by an angle randomly drawn from 0 to 180 degrees to get data for one task. The process is repeated 100 times to get data for 100 tasks out of which 80 are train tasks and 20 are test tasks. Fig. 1 shows 3 such tasks for $\theta = 0, 90$ and 180 where the supports don't overlap at all, and Fig. 2 shows 13 tasks where the supports overlap.

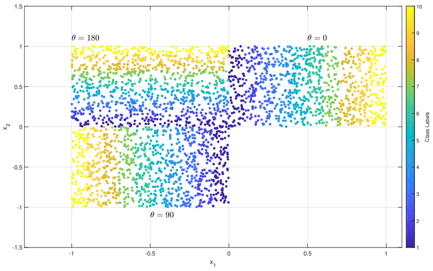

Figure 1: Synthetic Dataset: Three tasks $\theta = \{0, 90, 180\}$

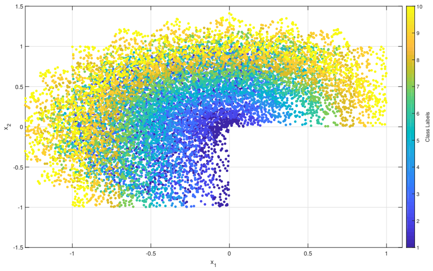

Figure 2: Synthetic Dataset: Thirteen tasks $\theta = \{0, 15, 30, ..., 180\}$

Satellite Dataset: The problem is described in the introduction, and we used the dataset presented by [30] modified for $c = 3$ spacecraft.

HAR Dataset: This is a human activity recognition using smart-phone dataset from UCI repository [1]. Each of 30 volunteers performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) wearing the smart-phone.

MNIST-MOD Dataset: We randomly draw 1000 images from MNIST's train dataset. Then we rotate each of this image by randomly drawn angle from 0 to 180 degrees and repeat this 100 times to get data for 100 tasks. Example for rotated MNIST dataset is shown in Fig. 3.

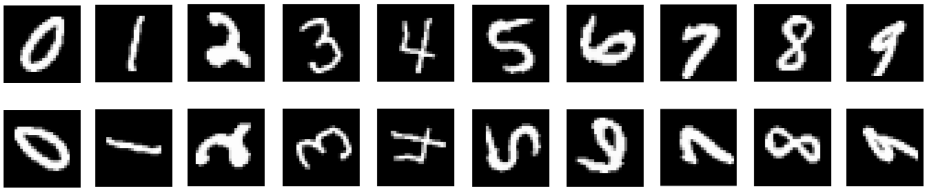

Figure 3: MNIST Data with no rotation (first row) and 90 degree rotation (second row)

We use all Gaussian kernels and a novel random Fourier Feature (RFF) approximation, which extends the usual RFF approximation on Euclidean space $\mathcal{X}$ [28] to extended the feature space $\mathcal{P}_{\mathcal{X}} \times \mathcal{X}$, to speed up the algorithm. We used Liblinear package for the implementation [11]. All hyperparameters were selected using five fold cross-validation and experiments were repeated 10 times. We show results in Table 2. The proposed method performs the best in three datasets and equally well in the one remaining dataset. The more our method outperforms pooling, the more knowledge can be shared between tasks.

Dataset	Pooling	Proposed Method
Synthetic	70.73 (±2.30)	**25.40** (±1.72)
Satellite	11.95 (±0.46)	**8.28** (±0.79)
HAR	1.69 (±0.56)	**1.68** (±0.58)
MNIST-MOD	22.79 (±1.38)	**21.39** (±1.24)

Table 2: Percentage Error

6 Conclusion and Future Work

In this work, we extended the kernel-based algorithm for domain generalization of [4] to the multiclass setting, along with its generalization error bound. We implemented the approach, demonstrating its improved performance with respect to a pooling strategy on four data sets. Future work will focus on improved generalization bounds and extensions to zero shot learning. Generalization error bound depends polynomially on c and for large number of classes, this may not be desirable. We intend to investigate assumptions under which there is a chance to improve this dependency. In extensions, we are interested in zero shot learning where training tasks have c classes and test tasks have $c + 1$ classes.

References

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones.

[2] J. Baxter. A Bayesian/information theoretic model of learning to learn via multiple task sampling. 28(1):7–39, 1997.

[3] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[4] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2178–2186. 2011.

[5] J. Carbonell, S. Hanneke, and L. Yang. A theory of transfer learning with applications to active learning. 90(2):161–189, 2013.

[6] A. Christmann and I. Steinwart. Universal kernels on non-standard input spaces. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 406–414, 2010.

[7] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. 39(9):1853–1865, 2016.

[8] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[9] Z. Ding and Y. Fu. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing*, 2017.

[10] Ü. Doğan, T. Glasmachers, and C. Igel. A unified view on multi-class support vector classification. *Journal of Machine Learning Research*, 17(45):1–32, 2016.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

[12] M. Ghifary, D. Balduzzi, B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. 39(7):1411–1430, 2017.

[13] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[14] T. Grubinger, A. Birlutiu, H. Schöner, T. Natschläger, and T. Heskes. Domain generalization based on transfer component analysis. In I. Rojas, G. Joya, and A. Catala, editors, *Advances in Computational Intelligence. IWANN 2015*, volume 9094 of *Lecture Notes in Computer Science*, pages 325–334. Springer, 2015.

[15] T. Grubinger, A. Birlutiu, H. Schöner, T. Natschläger, and T. Heskes. Multi-domain transfer component analysis for domain generalization. *Neural Processing Letters*, pages 1–11, 2017.

[16] C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[17] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

[18] D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017.

[19] B. London, B. Huang, B. Taskar, and L. Getoor. Collective stability in structured prediction: Generalization from one example.

[20] A. Maurer. Transfer bounds for linear feature learning. 75(3):327–350, 2009.

[21] A. Maurer. A vector-contraction inequality for rademacher complexities. *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, pages 3–17, 2016.

[22] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 343–351, 2013.

[23] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. *arXiv preprint arXiv:1709.10190*, 2017.

[24] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML'13)*, volume 28 of *Proceedings of Machine Learning Research*, pages I–10–I–18, 2013.

[25] S. J. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[26] A. Pentina and S. Ben-David. Multi-task and lifelong learning of kernels. In K. Chaudhuri, C. Gentile, and S. Zilles, editors, *Algorithmic Learning Theory: 26th International Conference (ALT'15)*, volume 9355 of *Lecture Notes in Computer Science*, pages 194–208. Springer, 2015.

[27] A. Pentina and C. Lampert. A pac-bayesian bound for lifelong learning. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 991–999, 2014.

[28] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[29] H. Ramaswamy and S. Agarwal. Convex calibration dimension for multiclass loss matrices. *Journal of Machine Learning Research*, 17, 2016.

[30] S. Sharma and J. W. Cutler. Robust orbit determination and classification: A learning theoretic approach. *Interplanetary Network Progress Report*, 203:1, 2015.

[31] A. Tewari and P. L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(May):1007–1025, 2007.

[32] J. Weston and C. Watkins. Multi-class support vector machines. Technical report, Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May, 1998.